LinearTurboFold: Linear-Time RNA Structural Alignment and Conserved Structure Prediction with Applications to Coronaviruses

Sizhen Li[a], He Zhang[b,a], Liang Zhang[b,a], Kaibo Liu[b,a], Boxiang Liu[b], David H. Mathews[c,d,e,♣], and Liang Huang[a,b,♣]

[a]School of Electrical Engineering & Computer Science, Oregon State University, Corvallis, OR 97330, USA; [b]Baidu Research USA, Sunnyvale, CA 94089, USA; [c]Dept. of Biochemistry & Biophysics; [d]Center for RNA Biology; [e]Dept. of Biostatistics & Computational Biology, University of Rochester Medical Center, Rochester, NY 14642, USA

Many functional RNA structures are conserved across evolution, and such conserved structures provide critical targets for diagnostics and treatment. TurboFold II is a state-of-the-art software that can predict conserved structures and alignments given homologous sequences, but its cubic runtime and quadratic memory usage with sequence length prevent it from being applied to most full-length viral genomes. As the COVID-19 outbreak spreads, there is a growing need to have a fast and accurate tool to identify conserved regions of SARS-CoV-2. To address this issue, we present LinearTurboFold, which successfully accelerates TurboFold II without sacrificing accuracy on secondary structure and multiple sequence alignment prediction. LinearTurboFold is orders of magnitude faster than TurboFold II, e.g., 372× faster (12 minutes vs. 3.1 days) on a group of five HIV-1 homologs with average length 9,686 *nt*. LinearTurboFold is able to scale up to the full sequence of SARS-CoV-2, and identifies conserved structures that have been supported by previous studies. Additionally, LinearTurboFold finds a list of novel conserved regions, including long-range base pairs, which may be useful for better understanding the virus.

1. Introduction

RNAs play important roles in multiple cellular processes[1–3], and many of their functions rely on folding to specific structures. To maintain their functions, secondary structures of RNA homologs are conserved across evolution[4–7]. These conserved structures provide critical targets for diagnostics and treatment. Thus, there is a need for developing fast and accurate computational methods to identify conserved structural regions.

Commonly, conserved structures involve compensating base pair changes, where two changes in primary sequences still preserve base pairs in secondary structures. For instance, a GC base pair is replaced by an AU or a CG base pair in homologous sequences. These compensating changes provide strong evidence for conserved structures[8]. Meanwhile, they also make it harder to align sequences when structures are unknown. To solve this issue, Sankoff proposed a dynamic algorithm that simultaneously predict structures and a sequence alignment for two or more sequences[9]. Several software packages provide implementations of the Sankoff algorithm[10–14]. One drawback of this approach is that the Sankoff algorithm runs in $O(n^{3k})$ against averaged sequence length n and k sequences.

TurboFold II[15], an extension of TurboFold[16], provides a more computationally efficient method to align and fold sequences. TurboFold II takes multiple unaligned RNA sequences as input, and estimates the posterior co-incidence probabilities for all pairs of sequences and the base pair probability matrix for each sequence using probabilistic models of a Hidden Markov Model (HMM)[17] and a partition function[18], respectively. It iteratively refines estimations so that the alignments and structure probabilities conform more closely to each other and converge on conserved structures. Finally, TurboFold II generates a multiple sequence alignment using probabilistic consistency transformation and progressive alignment methods, and predicts secondary structures using downstream methods, such as Maximum Expected Accuracy (MEA)[19–21] and ProbKnot[22]. TurboFold II is significantly more accurate than other methods when tested on families of RNAs with known structures and alignments. Though TurboFold II is substantially more efficient than the Sankoff approach, it can not scale to longer sequences due to its end-to-end $O(k^2n^2 + kn^3)$ runtime and $O(k^2n^2)$ memory usage, which mainly suffer from calculating RNA partition functions and base pairing probabilities for each sequence ($O(kn^3)$). For example, TurboFold II takes 3.1 days, along with 54 GB memory usage, for a group of five HIV-1 sequences with average length 9,686 *nt*.

As the COVID-19 outbreak spreads, there is a growing need for a tool, such as TurboFold II, to identify conserved regions along with their structural propensities. However, the runtime and memory usage bottlenecks of TurboFold II prevent it from being applied to full-length viral genomes, especially to SARS-CoV-2, the virus that causes the COVID-19 pandemic, which has a genome of length nearly 30,000 *nt* and is far beyond TurboFold II's scope.

Recently, we introduced LinearPartition[23], a linear-time approximation of the RNA partition function to accelerate the classical cubic-time partition function algorithm[24] and the estimation of base pairing probabilities. Unlike previous local linear-time algorithms[25,26], LinearPartition is a global algorithm without constraints on the pairing distance. Thus, a natural strategy to overcome the slowness and improve the scalability of TurboFold II is to replace the partition function and base pairing probability calculation in TurboFold II with LinearPartition. Following this idea, we propose LinearTurboFold, which can output the RNA structural alignment and identify conserved base pairs in linear time. To make LinearTurboFold an end-to-end linear-time algorithm, we further present LinearAlignment, which linearizes and approximates the pairwise alignment by applying the beam pruning heuristic algorithm[27]. LinearTurboFold uses ThreshKnot[28] to predict secondary structures. ThreshKnot uses a probability threshold θ to disallow any pair whose probability falls below θ, then builds structure of mutually maximal probability pairing partners. Based on the significant acceleration, LinearTurboFold can scale up to whole-genome viruses.

We run LinearTurboFold on a collected dataset with diverse RNA homologous sequences of length ranging from about 200 *nt* up to 30,000 *nt*. LinearTurboFold achieves linear runtime and memory usage against sequence length. We compare the secondary structure

Author contributions: L.H. and D.H.M. conceived the idea and directed the project. S.L., H.Z., L.H., and D.H.M. designed the algorithm; S.L. implemented it. D.H.M. guided the evaluation that S.L. and L.Z. carried out. S.L. and H.Z. wrote the manuscript; L.H., and D.H.M. revised it. B.L. guided the SARS-CoV-2 experiment. L.K. made the webserver.

The authors declare no conflict of interest.

♣Corresponding authors: David_Mathews@urmc.rochester.edu, liang.huang.sh@gmail.com.

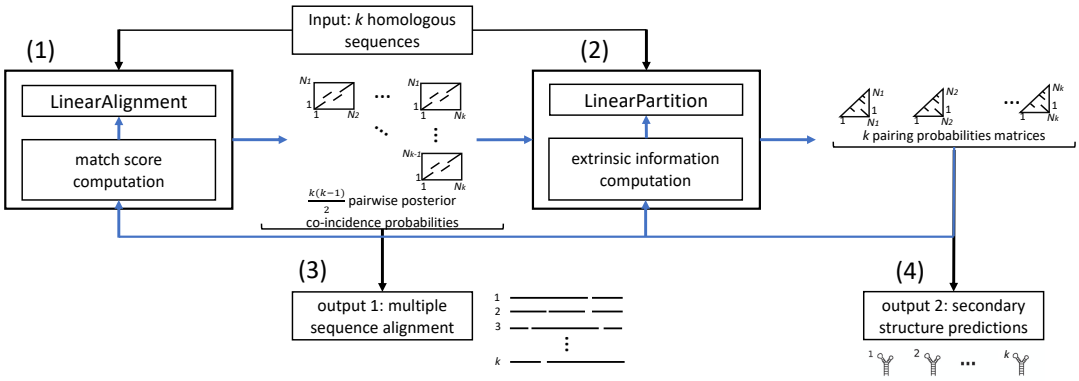


Fig. 1. The framework of LinearTurboFold. LinearTurboFold inherits the iterative process from TurboFold II with k homologous RNA sequences as input. The iterative process includes two major modules (1) pairwise alignment posterior probability estimation and (2) base pairing probability prediction. The blue lines and arrows represent the information flow between modules. In module (1), LinearAlignment, a linear-time pairwise alignment and posterior co-incidence probability computation approximation, incorporates the match score, which uses the structural information, i.e. base pairing probabilities, to guide the pairwise alignment. In module (2), LinearPartition is modified by incorporating the extrinsic information to refine the base pairing probabilities estimation. The extrinsic information maps structural information from other sequences to the target sequences under the help of the posterior co-incidence probability. After a user-specified number of iterations (with default value of 3), module (3) computes the final multiple sequence alignment over the pairwise alignment predictions, and module (4) uses ThreshKnot to predict secondary structures.

and alignment prediction accuracy between LinearTurboFold and benchmark methods, including LocARNA-P[29], MXSCARNA[12] and TurboFold II on the RNAStrAlign test set. LinearTurboFold leads to equal or better accuracies among the benchmarks. With LinearTurboFold, we explore conserved structural regions among betacoronavirus with the well-known RNA structures. Additionally, we list novel conserved regions whose functions are not well understood.

2. Results

A. LinearTurboFold Algorithm. Formally, we define a group of k homologous sequences as $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}$. For the m^{th} sequence, $\mathbf{x}^{(m)} = x_1^{(m)} x_2^{(m)} \ldots x_{n_m}^{(m)}$ of sequence length n_m, and each nucleotide $x_i^{(m)}$ takes the value from the alphabet set $\{A, U, G, C\}$. n is the average sequence length. $\mathbf{x}_{[i,j]}^{(m)}$ indicates the subsequence from position i to j of the sequence $\mathbf{x}^{(m)}$.

LinearTurboFold approximates the TurboFold II algorithm in linear time end-to-end. As presented in Figure 1, LinearTurboFold inherits the iterative framework from TurboFold II, which consists of two major modules: (1) posterior co-incidence probability estimation[17] for each pair of sequences using the HMM algorithm and (2) partition function and base pair probability prediction for each sequence. To accelerate TurboFold II, two types of iterative refinement occur with linearization. In module (1), LinearAlignment approximates the pairwise alignment and posterior co-incidence probability calculation in linear time, and is optimized iteratively by incorporating the match score, which integrates structural information, i.e. the predicted base pair probabilities from module (2). In module (2), LinearPartition refines base pairing probabilities by taking in the extrinsic information, a proclivity for base pairing inferred from the base pairing probabilities of other sequences and mapped to the target sequence via the estimated posterior co-incidence probabilities from module (1). Thus the posterior co-incidence probability and base pair probability prediction performance are jointly improved by taking advantage of the information from each other. After several iterations, LinearTurboFold computes the final multiple sequence alignment (MSA) based on pairwise alignment probabilities in module (3), and predicts secondary structures over base pairing probabilities in module (4).

A.1. Linearized Posterior Co-incidence Probability Computation. We develop LinearAlignment, an HMM alignment with the beam search algorithm, that approximates the pairwise alignment and posterior co-incidence probability computation in linear time. LinearAlignment adopts states as well as the parameters of the HMM algorithm realized in Harmanci *et al.*[17]. Specifically, there are three states: aligning a pair of nucleotides from two sequences (ALN); inserting one nucleotide in the first sequence but a gap in the second sequence (INS1); and a nucleotide insertion in the second sequence with a gap insertion in the first sequence (INS2). Transitions between any two states are allowed. State ALN emits a pair of two nucleotides ($x_i^{(m)}$, $x_j^{(n)}$); INS1 and INS2 emit a nucleotide paired with a gap ($x_i^{(m)}$, $-$) and ($-$, $x_j^{(n)}$), respectively, where a short dash ($-$) means an insertion of a gap. A pair with all gaps ($-$, $-$) is not permitted to be emitted. The emission of nucleotides $x_i^{(m)}$ or $x_j^{(n)}$ keeps the same order in the two sequences $\mathbf{x}^{(m)}$ or $\mathbf{x}^{(n)}$, respectively. As shown in Figure 2, the x-axis and y-axis of the alignment matrix represent two sequences $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$. State ALN, INS1 or INS2 is a step along the diagonal, bottom or left line of cells towards the end point, respectively. And a possible alignment is a sequence of states, i.e. a continuous path from the start to the end points. There are two complete alignment paths (in green and blue) in Figure 2 and the corresponding alignments with different colors are on the right side of the matrix. We use the forward-backward algorithm to estimate marginal probabilities for the alignment of two nucleotides.

The HMM algorithm in Harmanci *et al.*[17] computes the maximum-likelihood alignment probabilities or the forward scores for all position pairs (i, j), where i and j are positions in the two homologs $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$. Starting from the bottom-left position, it iterates columns from left to right, and cells from bottom to up in each column. While, LinearAlignment uses a different topological order based on the step counts to fill out the matrix. The step counts are the sum value of i and j, i.e. the number of nucleotides in the current alignment for the prefixes $\mathbf{x}_{[1,i]}^{(m)}$ and $\mathbf{x}_{[1,j]}^{(n)}$. States (i, j) with the same step counts make transitions to the next states together. Thus, LinearAlignment computes all the states (i, k) anti-diagonally from bottom-left towards top-right. As illustrated in Figure 2, the numbers along the paths are step counts for the prefix alignments. There are

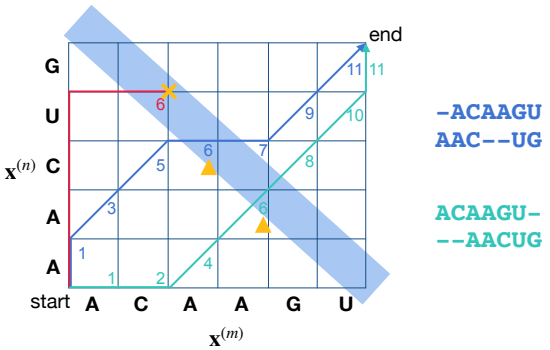


Fig. 2. Example illustration of LinearAlignment. The nucleotides along x- and y-axis are two simple sequences $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$. State ALN, INS1 or INS2 is a step along the diagonal, bottom or left line of cells towards the end point, respectively. And a possible alignment is a set of continuous and sequential states from the start to end points. Two complete pairwise alignments (in blue and green) are on the right side of the matrix with the corresponding colors, where a short dash (-) means a gap insertion. LinearAlignment is a HMM alignment based on the step counts. The numbers along with paths are step counts for the partial alignments. Position pairs with the same step counts make updates to next states together, for example, all the positions with step counts 6 in blue background. After LinearAlignment applies the beam search algorithm, position pairs with the same step counts compete with each other and only the promising states with higher probabilities survive. For the step count 6, if only three paths exist and the beam size is 2, the path in red is discarded because it has the lowest probability.

three paths of step counts 6 in blue background, and they will update next states together.

With a different topological order, the HMM alignment is still an exhaustive search algorithm and costs quadratic time and space. To reduce the runtime, for each step counts s from 0 to $n_m + n_n$, LinearAlignment applies the beam search algorithm [27] and only keeps a limited number of promising states with higher forward scores, which are further transited to next states together. By pruning low-scoring candidates, we reduce the runtime from $O(n^2)$ to $O(b_1 n)$, where b_1 is a user-specified beam size and the default value is 100. As depicted in Figure 2, three paths of step counts 6 compete with each other. The alignment path (in red) is unlikely to be a reasonable alignment because of many insertions. If only these three paths exist and the beam size is 2, the path in red is discarded due to the lowest probability.

To encourage the pairwise alignment conforming with estimated secondary structures, TurboFold II incorporates the match score [30] into the HMM alignment computation. TurboFold II separately precomputes match scores for all the $O(n^2)$ alignment pairs of all sequences in pairwise before the HMM alignment calculation. However, only a linear number of pairs ($O(b_1 n)$) survives after applying the beam pruning in LinearAlignment. To reduce redundant time and space usage, LinearTurboFold calculates the corresponding match scores for alignable pairs when they are visited in LinearAlignment. Overall, LinearTurboFold reduces the runtime of the whole module of pairwise posterior co-incidence probability computation from $O(k^2 n^2)$ to $O(k^2 b_1 n)$ by applying the beam search heuristic to the HMM alignment, and calculating only the match scores that are needed.

A.2. Base Pairing Probability Estimation. The classical partition function algorithm scales cubically with sequence length. The slowness limits its extension to longer sequences. To address this bottleneck, our recent LinearPartition algorithm approximates the partition function and base paring probability matrix in linear time. LinearPartition is significantly faster, and correlates better with the ground truth structures. Thus LinearTurboFold uses LinearPartition to predict base pair probabilities instead of $O(n^3)$-time partition function.

LinearTurboFold modifies LinearPartition to incorporate extrinsic information inferred from homologous sequences to predict base pair probabilities iteratively. The extrinsic information maps the estimated base pairing probabilities of other sequences to the target sequence via the co-incident nucleotides between two sequences. TurboFold II introduces the extrinsic information $\pi(i, j)$ in the partition function as a pseudo-free energy term for each base pair (x_i, x_j). Similarly, in LinearPartition, for each span $[i, j]$ associated with its partition function $Q_{i,j}$, the partition function is modified as $\tilde{Q}_{i,j} = Q_{i,j}\pi(i, j)^\lambda$ if (x_i, x_j) is an allowed pair, where λ denotes the contribution of the extrinsic information relative to the intrinsic information. Specifically, at each step j, among all possible spans $[i, j]$ where x_i and x_j are paired, we replace the original partition function $Q_{i,j}$ with $Q_{i,j}\pi(i, j)^\lambda$ by multiplying the extrinsic information. Then LinearTurboFold applies the beam pruning heuristic over the modified partition function $\tilde{Q}_{i,j}$ instead of the original.

TurboFold II obtains the extrinsic information for all the $O(n^2)$ base pairs before the partition function calculation of each sequence, while only a linear number of base pairs survives in LinearPartition. Thus, LinearTurboFold only requires the extrinsic information for those promising base pairs that are visited in LinearPartition. One challenge is, in TurboFold II, the extrinsic information matrix is normalized by the maximum value before being introduced into the partition function, where the normalization factor is not determined in advance and vary with iterations. We put forward an approximation as the solution: using the unnormalized values directly. We check that the maximum value has no relationship with sequence length, confirming in Figure SI 1.

A.3. Multiple Sequence Alignment and Secondary Structure Prediction. After several iterations, TurboFold II builds the multiple sequence alignment using a probabilistic consistency transformation [31], generating a guide tree and performing progressive alignment over the pairwise posterior co-incidence probabilities. The whole procedure is accelerated in virtue of the sparse matrix by discarding position pairs of values smaller than a threshold (0.01 by default). Since LinearAlignment uses beam search and only saves a linear number of co-incident pairs, the MSA computation in LinearTurboFold costs linear runtime against the sequence length automatically.

LinearTurboFold feeds estimated base pair probabilities into the downstream method to predict secondary structures. To maintain the end-to-end linear-time property, LinearTurboFold uses ThreshKnot [28], which is a thresholded version of ProbKnot and only considers base pairs with probabilities exceeding the threshold θ ($\theta = 0.3$ by default). We evaluate the performance of ThreshKnot and MEA with different hyperparameters (θ and γ). on a sampled RNAStrAlign training set. As shown in Figure SI 2, ThreshKnot is closer to the upper right-hand than MEA, which indicates that ThreshKnot always has a higher Sensitivity than MEA at a given PPV.

B. Efficiency and Scalability. To evaluate the scalability of LinearTurboFold against the sequence length, we collect a dataset including groups of five homologous sequences with sequence length ranging from 210 *nt* to 2920 *nt*. In addition to running both TurboFold II and LinearTurboFold on this dataset, we further extend LinearTurboFold to whole-genome HIV-1 and SARS-CoV sequences of length ~10,000 *nt* and ~30,000 *nt*, respectively. The detailed data collection process is introduced in the Methods section. Figure 3A indicates that LinearTurboFold scales almost linearly with the sequence length,

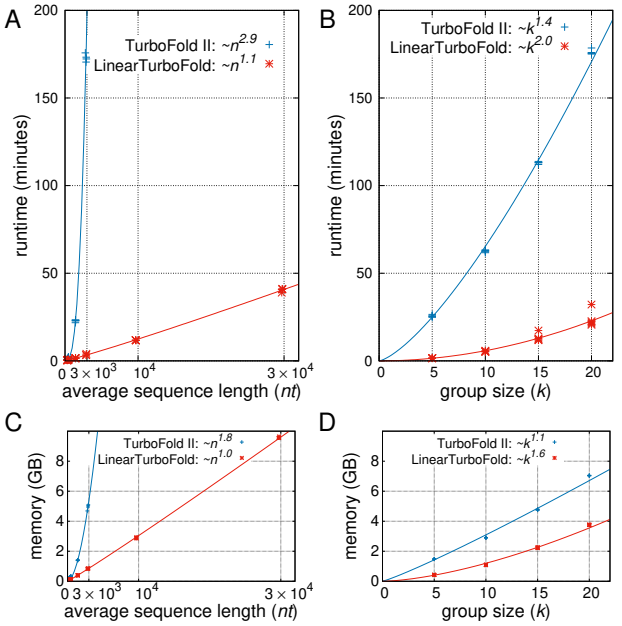


Fig. 3. End-to-end runtime and memory usage comparisons between TurboFold II and LinearTurboFold. We use a Linux machine (CentOS 7.7.1908) with 2.30 GHz Intel Xeon E5-2695 v3 CPU and 755 GB memory, and gcc 4.8.5 for benchmarks. **A**: End-to-end runtime comparison with the sequence length. We run TurboFold II and LinearTurboFold on sampled groups from the RNAStrAlign dataset and 23S rRNA sequences. LinearTurboFold was further extended to HIV-1 and SARS-CoV-2 whole-genome sequences. The curve-fittings are log-log in gnuplot and use the data for $n > 700$. The group size is 5. **B**: End-to-end runtime comparison with the group size (k) ranging from 5 to 20, and the sequence lengths are fixed around 1,500 nt. **C**: Memory usage comparisons with the sequence length on the same groups used to evaluate the runtime. **D**: Memory usage comparisons with different group sizes.

which allows it to scale to the full-length SARS-CoV sequences without any constraints on base-pairing length. For instance, for a group of five SARS-CoV sequences, LinearTurboFold only takes about 50 minutes and 10 GB space with the default hyperparameters. Figure 3A also confirms that the runtime of TurboFold II grows cubically with sequence length and is substantially slower than LinearTurboFold. TurboFold II is 372× slower (12 minutes vs. 3.1 days) than LinearTurboFold on a group of five homologous HIV sequences with the average sequence length 9,686 *nt*. Figure 3C illustrates that LinearTurboFold costs linear space with sequence length, while TurboFold II costs quadratic.

Additionally, to assess the scalability of LinearTurboFold against the group size, we build a dataset of group size ranging from 5 to 20, by sampling sequences from the RNAStrAlign 16S rRNA family. The sequence length is fixed around 1,500 *nt*. Though, in Figure 3B, the runtime complexity of TurboFold II grows less quadratically ($O(k^{1.4})$ while LinearTurboFold grows quadratically, the latter is significantly faster than the former. Figure 3D shows, in LinearTurboFold, the memory usage grows in $O(k^{1.6})$ with the group size, while it is $O(k^{1.1})$ in TurboFold II. The fact that the complexity of runtime and memory usage against k for LinearTurboFold is larger, is mainly because the cubic complexity of partition function calculation, which dominates in TurboFold II, has been linearized in LinearTurboFold. Specifically, posterior co-incidence probability estimation uses $O(k^2 b_1 n)$ space and base pairing probability prediction takes $O(k b_2 n)$ in LinearTurboFold. In practice, as shown in Figure SI 3, the alignment runtime occupies about 60% of the total used space when k is 5, and dominates the memory usage as the group size increases.

C. Secondary Structure And Alignment Prediction Accuracy.

We compare the accuracy of both predicted secondary structures and multiple sequence alignments between LocARNA-P[29], MXSCARNA[12], TurboFold II and LinearTurboFold. Both LocARNA-P and MXSCARNA predict Sankoff-style structural alignment, and like TurboFold II, they take raw RNA sequences instead of a pre-computed fixed alignments as input. All the benchmarks use the default options and hyperparameters running on the RNAStrAlign test set. TurboFold II iterates three times, then predicts secondary structures by MEA (γ=1). LinearTurboFold also runs three iterations with default beam sizes ($b_1 = b_2 = 100$) in LinearAlignment and LinearPartition, then identifies structures with ThreshKnot ($\theta = 0.3$).

We use Positive Predictive Value (PPV) and Sensitivity to measure the secondary structure prediction accuracy. Figure 4 compares the accuracies of secondary structure prediction and alignment between the benchmarks and LinearTurboFold. Regarding structure prediction, TurboFold II and LinearTurboFold are more accurate than the other two on most families, except for 16S rRNA, where MXSCARNA is the best. Compared with TurboFold II, LinearTurboFold has a slight decrease in PPV but improvement in Sensitivity. To assess the statistical significance of accuracy differences between the other programs and LinearTurboFold, two tailed significance tests are conducted and annotated in Figure 4A and B on the top of corresponding bars when $p < 0.05$. The Sensitivity of LinearTurboFold is significantly better than TurboFold II on all the test families except for telomerase, and also better on the overall. For PPV, TurboFold II is significantly better than LinearTurboFold. The PPV and Sensitivity are adjustable by changing the threshold in ThreshKnot. Overall, the F1 score of LinearTurboFold at 72.0% is better than TurboFold II at 71.0% on the test set. LinearTurboFold are significantly better than LocARNA-P and MXSCARNA in PPV and Sensitivity except for 16S rRNA.

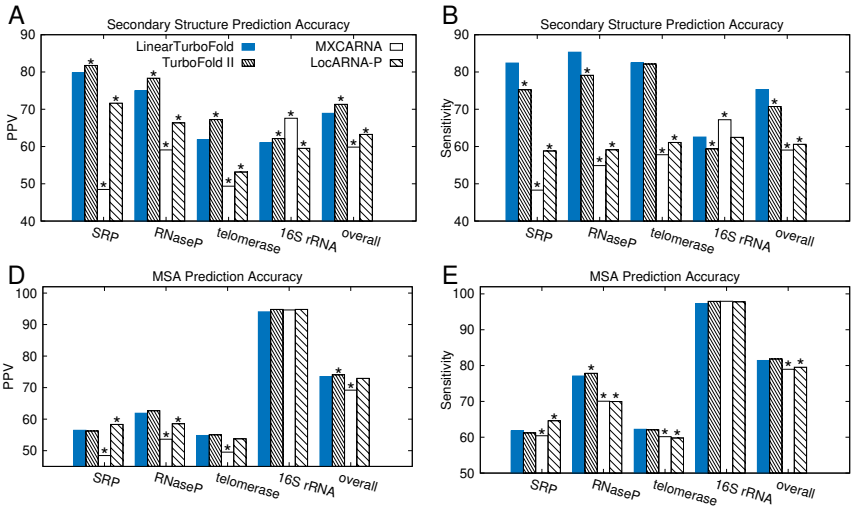
Similarly we calculate PPV and Sensitivity to evaluate the accuracy of predicted multiple sequence alignments[32]. PPV is the fraction of predicted aligned nucleotides that are also correct, and Sensitivity is the fraction of aligned nucleotides in the ground truth that are predicted. LocARNA-P acchieves the best performance among all methods on the SRP family, because it is more accurate for families with low sequence identity. Overall, LinearTurboFold obtains significantly higher PPV and Sensitivity than LocARNA-P, better PPV than MXSCARNA. LinearTurboFold leads to comparable Sensitivity and PPV with TurboFold II. The overall F1 score of TurboFold II is the best on the test set.

D. Highly conserved base pairs in SARS-related betacoronaviruses and SARS-CoV-2.

The current outbreak COVID-19 is causing a global pandemic that raises an emergent requirement for identifying potential targets for diagnostics and antiviral therapeutics. Evolutionarily conserved RNA secondary structures play vital biological roles and provide potential targets.

Conserved structures have been discovered and found to be of crucial importance in the life circle of the coronavirus, but most regions remain unexplored. We use LinearTurboFold to identify highly conserved structures across SARS-related betacoronaviruses and SARS-CoV-2. In addition to prediction for several currently known structural elements, LinearTurboFold captures novel conserved structures whose functions have not been reported yet.

We use the sequences curated by Ceraolo *et al.*[33]. Following the data-processing in the paper[34], we filter out SARS-CoV-2 sequences except the reference sequence NC_0405512.2[35], remove two MERS sequences and retain only whole-genome sequences. We further discard an identical sequence and a relatively short sequence. These remaining 9 sequences comprise five SARS sequences, three SARS-



Secondary structure prediction F1 Score	SRP	RNase P	Telomerase	16S rRNA	overall
LinearTurboFold	**81.0**	**79.8**	70.6	61.8	**72.0**
TurboFold II	78.4	78.4	**73.9**	60.7	71.0
MXSCARNA	48.4	56.9	53.2	**67.4**	67.4
LocARNA-P	64.6	62.5	56.8	61.0	61.9

MSA prediction F1 Score	SRP	RNase P	Telomerase	16S rRNA	overall
LinearTurboFold	**59.0**	68.6	**58.3**	95.6	77.3
TurboFold II	58.6	**69.4**	**58.3**	**96.3**	**77.8**
MXSCARNA	53.8	60.8	54.3	**96.3**	73.8
LocARNA-P	61.3	63.7	56.6	**96.3**	76.1

Fig. 4. The accuracy comparisons of secondary structure prediction and multiple sequence alignment prediction between the benchmarks and LinearTurboFold on the RNAStrAlign test set. **A** and **B**: PPV and sensitivity of predicted secondary structures on each family. Statistical significance (two-tailed) between the benchmarks and LinearTurboFold are marked as stars (*) on the top of the corresponding bars if $p < 0.05$. **D** and **E**: PPV and sensitivity of estimated multiple sequence alignment. **C** and **F**: The F1 score of secondary structure and multiple sequence alignment prediction on test families.

related bat coronavirus and one reference NC_0405512.2 of SARS-CoV-2. The average pairwise sequence identity of these nine whole-genome sequences is 0.88 calculated by [36].

LinearTurboFold does not provide a single consensus structure for the homologous sequences. Instead, it allows flexibility in the predicted structures to account for structure evolution as in TurboFold II. Base pairs are said being conserved if most of sequences form base pairs at the same alignment positions.

A previous paper [34] identified sequence conservations and conserved structures in SARS-related viruses and SARS-CoV-2. They used RNAz [37] to scan the alignment of SARS-CoV-2 sequences with windows of length 120 *nt* sliding by 40 *nt* for regions likely to have conserved structure, then ranked windows by probability of being thermodynamic stable. Additionally, contiguous stretches of at least 15 *nt* are required to be conserved exceeding a specific cutoff value. This method has two drawbacks: first, identified conserved structures do not include long-range base-pairing due to the limited window size; second, the constraints on the primary sequences would miss base pairs with compensating changes, i.e. two changes in sequence across evolution that conserve a base pair. The second drawback is crucial because covariations are a signal of conserved RNA secondary structure [38–42]. LinearTurboFold can solve these two issues. First, LinearTurboFold can capture long-distance interactions because of its scalibility to the whole-genome SARS-CoV-2 sequences without constraints on the pairing distance. Furthermore, LinearTurboFold encodes structural information of homologous sequences, which enables it to identify conserved co-variational base pairs. As illustrated in Figure 5, base pairs with compensating changes are highlighted in blue and annotated with the alternative pairs.

We compare LinearTurboFold results with well-characterized structures across betacoronaviruses including the 5' UTR structure, the frameshifting stimulation element (FSE) and the 3' UTR structure of SARS-CoV-2. The ~300 *nt* 5' UTR includes five conserved structural elements called SL1, SL2, SL3, SL4 and SL5, with critically functional roles in viral genome replication [43]. It harbors an essential element, the leader transcription-regulating sequences (TRS-L), involved in long-range interactions with 3' UTR essential for discontinuous transcription [44]. Additionally, the SL5 contains the start codon (AUG) for ORF1ab, which occupies about two thirds of the genome, and encodes the replicase/transcriptase polyprotein. The frameshifting simulation element locates at the boundary of ORF1a and ORF1b and it includes a pseudoknot in the canonical model, which is necessary for regulating programmed -1 ribosomal frameshifting to bypass the stop codon at the end of ORF1a and continually translate the protein in ORFab [45]. The slippery site (UUUAAAC) upstream of the pseudoknot comprises the coronavirus frameshift signal. The prevailing 3' UTR model contains several structural elements. Close to the 5' end of the 3' UTR, a mutually exclusive formation of a bulged stem-loop (BSL) and a pseudoknot is important for viral replication [46]. The hyper-variable region (HVR) is nonessential for viral RNA synthesis but affects pathogenicity in mice [47]. A triple helix junction is folded downstream of the hyper-variation region [48]. The stem–loop II-like motif (s2m) is the most highly conserved element within the coronaviruses immediately upstream of the end of 3' UTR poly-A tail. Its rigorous conservation in viral pathogen genomes suggests this element is an attractive targets for the anti-viral therapeutic design [49].

Figure 5 represents conserved base pairs identified by LinearTurboFold for the 5' UTR, FSE and 3' UTR, and most of base pairs are 100% conserved among the SARS, bat coronavirus, and SARS-CoV-2 reference sequence. LinearTurboFold predictions largely agree with the prevailing models with some variations. For the 5' UTR (Fig. 5B), LinearTurboFold identifies the SL1, SL2, SL4 and SL5 with all the base pairing probabilities higher than 0.6. For the FSE (Fig. 5A), LinearTurboFold predicts two stem loops in the canonical three-stem pseudoknot motif. For the 3' UTR (Fig. 5C), LinearTurboFold found the BSL and s2m in the 3' UTR. In Figure 5, nucleotides in grey are not fully conserved on the sequence level, which indicates those positions are with possible evolutionary variations. And we attach more importance to covariations where both of the nucleotides are changed but the base pair remains constant. Base pairs with covariations are highlighted in blue background and annotated with compensating changes in Figure 5. The SL4 and SL5 in 5' UTR and the two bulged stem-loop close to the 5' end of 3' UTR discovered by LinearTurboFold are supported by the co-variational base pairs.

Theoretical and experimental studies [50–52] demonstrate that long-range base pairs are common in natural RNAs, especially between

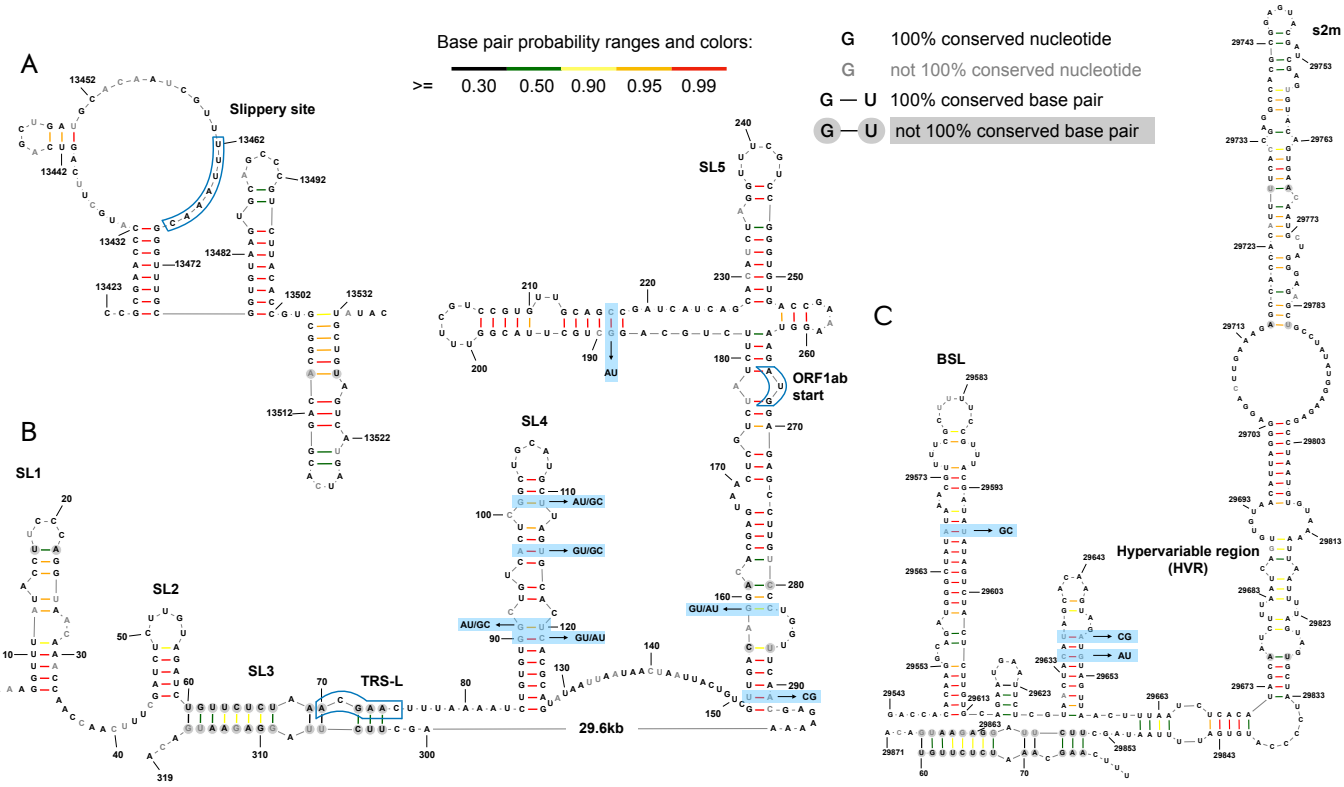


Fig. 5. Conserved secondary structures predicted by LinearTurboFold with ThreshKnot ($\theta = 0.3$) for **A**: frameshifting stimulation element, **B**: 5' UTR, and **C**: 3' UTR. Most of base pairs are fully conserved across nine sequences. Base pairs with a grey background are conserved across at least 8 sequences, except that the long-range interaction between 5' UTR and 3' UTR are conserved over 7 sequences. Base pairs are visualized by colors to indicate base pairing probabilities. Co-evolved base pairs are highlighted in blue and annotated with alternative base pairs. Nucleotides are black if 100% conserved and grey otherwise. Figures were drawn by StructureEditor (http://rna.urmc.rochester.edu/tutorials/workshop/Editor.html).

the 5' and 3' UTRs. LinearTurboFold predicts secondary structures globally without any limit on base-pairing distance, thus it can explore long-distance interactions across whole-genome sequences. LinearTurboFold detects genome cyclization of SARS-CoV-2 involving long-range base-pairing between the 5' and 3' ends. As depicted in Figure 5B and 5C, the canonical SL3 region (from the 60 nucleotide to the 75 nucleotide) in 5' UTR opens completely and some of positions form base pairs with bases at the end of 3' UTR (from the 29853 nucleotide to the 29865 nucleotide) over a distance of around 29.8 kilobases. Recently, COMRADES [53] was developed to capture RNA structural diversity and long-range base-pairing interactions *in vivo*. Interestingly, this long-distance interaction between SL3 in 5' UTR and 3' UTR was discovered *in vivo* by COMRADES [54], which highly supports LinearTurboFold's predictions.

LinearTurboFold does not identify the pseudoknots that are the prevailing model in the FSE and 3' UTR regions. Recent studies [54–56] do not detect the pseudoknot conformation in the 3' UTR either. Another study [57] performs DMS-MaPseq on infected Vero cells, then uses RNAstructure [18] to predict secondary structures, which uncovers an unexpected structure for FSE region compared with the canonical structure. The estimated FSE structure does not include the pseudoknot, but a sequence of 10 bases right after the slippery sequence is folded into a stem with a complementary sequence upstream of the slippery site, instead. As shown in Figure 5B, LinearTurboFold also detects this stem near the 5' of FSE region of eight base pairs (from the 13425 nucleotide to the 13432 nucleotide).

In addition to well-known structures, LinearTurboFold discovers new structures across the SARS-CoV-2 whole-genome, and the functional properties of those substructures may have not been explored. These structures potentially offer targeting positions for diagnosis and therapy of SARS-CoV-2. With a threshold 0.3 in ThreshKnot, LinearTurboFold detects 3801 base pairs fully conserved across nine SARS-related betacoronaviruses sequences, out of which 674 base pairs include variations, and 154 pairs have covariations as listed in Table SI 2. These conserved structures are highly conserved and more likely to be targets for antivirals. Among them, 686 conserved stems are at least 3 *nt* long, and the average length is 4.5 *nt*. The longest conserved stem consists of 11 base pairs between [8144, 8154] and [8211, 8221].

Additionally, we compare LinearTurboFold results with running LinearPartition with ThreshKnot ($\theta = 0.3$) on the reference sequence NC_0405512.2, as a negative control. For the 5' UTR structure, LinearPartition also identified long-range interactions between the 5' and the 3' UTRs, but it involves SL2 not SL3, which disagrees with a previous study [54] and LinearTurboFold's result. This is an evidence that homologous sequences assist to adjust the position of the long-distance base-pairing in LinearTurboFold.

3. Discussion

In this paper, we present LinearTurboFold, which achieves end-to-end linear runtime and memory usage for structural alignment and conserved structure prediction of RNA homologs. LinearTurboFold is orders of magnitude faster than TurboFold II on long sequences, e.g., it is 372× faster (12 minutes vs. 3.1 days) than TurboFold II on a group of five HIV-1 homologs with average length of 9,696 *nt*, and is able to scale up to the full genome of RNA viruses, such as

SARS-CoV-2 (about 3,000 *nt*).

To accelerate TurboFold II, LinearTurboFold linearizes all the computational modules in the TurboFold II framework: (1) linearizing pairwise sequence alignment by applying beam search for each step in the HMM alignment; (2) replacing $O(n^3)$-runtime partition function calculation with linear-time algorithm LinearPartition; (3) calculating extrinsic information and match scores only when needed; (4) linearizing multiple sequence alignment as a product of pairwise alignment linearization; and (5) using ThreshKnot as the default secondary structure prediction module. It is worth noticing that all these speed-up efforts do not sacrifice secondary structure and alignment prediction accuracy.

We confirm that:

1. LinearTurboFold successfully scales up to a group of nine coronavirus sequences (including SARS and SARS-CoV-2) and finishes in 1.7 hours. LinearTurboFold finds conserved structures that have been well-established in previous researches or supported by recent studies.

2. LinearTurboFold takes linear runtime and memory usage against sequence length, while TurboFold II takes cubic runtime and quadratic memory, which leads to significant improvement in effciency and scalability.

3. The approximation quality of LinearTurboFold is good, i.e., the overall F1-score of secondary structure predicted by LinearTurboFold is slightly better than TurboFold II, and much better than LocARNA-P, MXSCARNA.

We also list novel conserved regions we found by LinearTurboFold, which have not been studied or verified previously. Given the high accuracy of LinearTurboFold on benchmark datasets, as well as the high consistency of LinearTurboFold on the well-studied conserved structures, we believe that these regions are structurally conserved across evolution, and are useful for further understanding the structures and functions of the virus.

Methods

Datasets. Four kinds of dataset are used in the paper. First, to evaluate the scalability of LinearTurboFold with sequence length, we collected groups of homologous RNA sequences with sequence length ranging from 200 *nt* to 29,903 *nt* with the fixed group size 5. Sequences are sampled from RNAStrAlign dataset [15], the Comparative RNA Web (CRW) Site [58], the Los Alamos HIV database (http://www.hiv.lanl.gov/) and the SARS-related betacoronaviruses (SARSr) curated by Ceraolo *et al.* [33]. RNAStrAlign, aggregated and released with TurboFold II, is an RNA alignment and structure database. Sequences in RNAStrAlign are categorized into homologous families, and some of families are further split into subfamilies. Each subfamily or family includes a multiple sequence alignment and ground truth structures for all the sequences. 20 groups of five homologs, were randomly chosen from the small subunit ribosomal RNA (Alphaproteobacteria subfamily), SRP RNA (Protozoan subfamily), RNase P RNA (bacterial type A subfamily) and telomerase RNA families. For longer sequences, we sampled five groups of 23S rRNA (of sequence length ranging from 2,700 *nt* to 2,926 *nt*) from the CRW Site, HIV-1 genetic sequences (of sequence length ranging from 9,597 *nt* to 9,738 *nt*) from the Los Alamos HIV database, and SARSr sequences (of sequence length ranging from 29,484 *nt* to 29,903 *nt*), respectively. All the sequences in one group belong to the same subfamily or subtype. And we sampled five groups for each family and obtained 35 groups in total. Due to the long runtime, we did not run TurboFold II on HIV-1 and SARSr groups. Figure 3A and 3B results were from experiments on this collected dataset with varied sequence lengths.

To assess the scalability of LinearTurboFold with group size, we fixed the sequence length around 3,000 *nt*, and sampled 5 groups of 16S rRNA sequences from the small subunit ribosomal RNA (Alphaproteobacteria subfamily) with group size 5, 10, 15 and 20 respectively. Figure 3B and 3C were built on this dataset with different group sizes.

Following TurboFold, we built a test set from RNAStrAlign dataset to measure and compare the performance between benchmarks and LinearTurboFold. 100 groups of input sequences, consisting of 5, 10 or 20 homologous sequences, were randomly selected from the small subunit ribosomal RNA (Alphaproteobacteria subfamily), SRP RNA (Protozoan subfamily), RNase P RNA (bacterial type A subfamily) and telomerase RNA families from RNAStrAlign dataset. We removed sequences of length less than 1400, 200, 330 and 400 *nt* for these four families respectively to filter out subdomains. The detail information is summarized in Table SI 1.

The last dataset used in Figure SI 2 is a RNAStrAlign training set to compare between MEA and ThreshKnot. 40 groups of three, five and seven homologs were randomly sampled from 5S ribosomal RNA (Eubacteria subfamily), group I intron (IC1 subfamily), tmRNA, and tRNA families from RNAStrAlign dataset. We chose θ = 0.1, 0.2, 0.3, 0.4, 0.5 for ThreshKnot, and γ = 1, 1.5, 2, 2.5, 3, 3.5, 4, 8, 16 for MEA. We calculated the overall secondary structure prediction accuracy across all training families, and reported both PPV and Sensitivity.

Benchmarks. Both LocARNA-P [29] and MXSCARNA [12] predict Sankoff-style structural alignment. Sankoff's algorithm uses dynamic programming to simultaneously fold and align two or more sequences and it requires $O(n^{3k})$ time and $O(n^{2k})$ space for k input sequences with the average sequence length n. LocARNA-P extends LocARNA [10] with features based on sequence and structural match probabilities. LocARNA (local alignment of RNA) costs $O(n^2(n^2+k^2))$ time and $O(n^2+k^2)$ space by restricting the alignable regions. MXSCARNA progressively aligns multiple sequences, as an extension of the pairwise alignment algorithm SCARNA [59], with improved score functions. SCARNA first aligns stem fragment candidates, then removes the inconsistent matching in the post-processing to generate the sequence alignment. MXSCARNA reduces runtime to $O(k^3n^2)$ and space to k^2n^2 with limiting searching space of folding and alignment. Both MXSCARNA and LocARNA-P uses base pair probabilities pre-computed for each sequence as structural input.

Significance Test. We use a paired, two-tailed permutation test [60] to measure the significant different. Following the common practice, the repetition number is 10, 000, and the significance threshold α is 0.05.

References

1. Eddy. SR (2001) Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics* 2(12):919–929.
2. Doudna JA, Cech TR (2002) The chemical repertoire of natural ribozymes. *Nature* 418(6894):222–228.
3. Bachellerie JP, Cavaillé J, Hüttenhofer A (2002) The expanding snoRNA world. *Biochimie* 84(8):775–790.
4. Nawrocki EP, Eddy SR (2013) Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* 29(22):2933–2935.
5. Poch O, Sauvaget I, Delarue M, Tordo N (1989) Identification of four conserved motifs among the RNA-dependent polymerase encoding elements. *The EMBO Journal* 8(12):3867–3874.
6. Brown EA, Zhang H, Ping LH, Lemon SM (1992) Secondary structure of the 5' nontranslated regions of hepatitis C virus and pestivirus genomic RNAs. *Nucleic Acids Research* 20(19):5041–5045.
7. Ritz J, Martin JS, Laederach A (2013) Evolutionary evidence for alternative structure in RNA sequence co-variation. *PLoS Computational Biology* 9(7):e1003152–e1003152.
8. Rivas E, Clements J, Eddy SR (2020) Estimating the power of sequence covariation for detecting conserved RNA structure. *Bioinformatics* 36(10):3072–3076.
9. Sankoff D (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics* 45(5):810—-825.
10. Will S, Reiche K, Hofacker IL, Stadler PF, Backofen R (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Computational Biology* 3(4):e65.
11. Havgaard JH, Torarinsson E, Gorodkin J (2007) Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Computational Biology* 3(10):1896–1908.
12. Tabei Y, Kiryu H, Kin T, Asai K (2008) A fast structural multiple alignment method for long RNA sequences. *BMC Bioinformatics* 9(1):33.
13. Xu Z, Mathews DH (2011) Multilign: an algorithm to predict secondary structures conserved in multiple RNA sequences. *Bioinformatics* 27(5):626–632.
14. Mathews DH, Turner DH (2002) Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *Journal of molecular biology* 317(2):191–203.
15. Tan Z, Fu Y, Sharma G, Mathews DH (2017) TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research* 45(20):11570–11581.
16. Harmanci AO, Sharma G, Mathews DH (2011) Turbofold: iterative probabilistic estimation of secondary structures for multiple RNA sequences. *BMC bioinformatics* 12(1):108.
17. Harmanci AO, Sharma G, Mathews DH (2007) Efficient pairwise RNA structure prediction using probabilistic alignment constraints in dynalign. *BMC bioinformatics* 8(1):130.

18. Mathews DH (2004) Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA* 10(8):1178–1190.
19. Knudsen B, Hein. J (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research* 31(13):3423–3428.
20. Do C, Woods D, Batzoglou S (2006) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics* 22(14):e90–e98.
21. Lu ZJ, Gloor JW, Mathews DH (2009) Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA* 15(10):1805–1813.
22. Bellaousov S, Mathews DH (2010) Probknot: fast prediction of RNA secondary structure including pseudoknots. *RNA* 16(10):1870–1880.
23. Zhang H, Zhang L, Mathews DH, Huang L (2020) Linearpartition: linear-time approximation of RNA folding partition function and base-pairing probabilities. *Bioinformatics* 36(Supplement_1):i258–i267.
24. McCaskill JS (1990) The equilibrium partition function and base pair probabilities for RNA secondary structure. *Biopolymers* 29:11105–1119.
25. Bernhart SH, Hofacker IL, Stadler PF (2006) Local RNA base pairing probabilities in large sequences. *Bioinformatics* 22(5):614–615.
26. Kiryu H, Kin T, Asai K (2008) Rfold: an exact algorithm for computing local base pairing probabilities. *Bioinformatics* 24(3):367–373.
27. Huang L, Sagae K (2010) Dynamic programming for linear-time incremental parsing in *Proceedings of ACL 2010*. (ACL, Uppsala, Sweden), p. 1077–1086.
28. Zhang L, Zhang H, Mathews DH, Huang L (2019) Threshknot: Thresholded probknot for improved RNA secondary structure prediction. *bioRxiv*.
29. Will S, Joshi T, Hofacker IL, Stadler PF, Backofen R (2012) LocARNA-P: accurate boundary prediction and improved detection of structural RNAs. *RNA* 18(5):900–914.
30. Hofacker IL, Bernhart SH, Stadler PF (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics* 20(14):2222–2227.
31. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S (2005) Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome research* 15(2):330–340.
32. Eddy S (1996, accessed 2020-10-07) *Eddy Lab: Software (SQUID)*. http://eddylab.org/software.html.
33. Ceraolo C, Giorgi FM (2020) Genomic variance of the 2019-nCoV coronavirus. *Journal of medical virology* 92(5):522–528.
34. Rangan R, et al. (2020) RNA genome conservation and secondary structure in SARS-CoV-2 and SARS-related viruses: a first look. *RNA* 26(8):937–959.
35. Wu F, et al. (2020) A new coronavirus associated with human respiratory disease in china. *Nature* 579(7798):265–269.
36. Katoh K, Frith MC (2012) Adding unaligned sequences into an existing alignment using MAFFT and LAST. *Bioinformatics* 28(23):3144–3146.
37. Gruber AR, Findeiß S, Washietl S, Hofacker IL, Stadler PF (2010) RNAz 2.0: improved noncoding RNA detection in *Biocomputing 2010*. (World Scientific), pp. 69–79.
38. Holley RW, et al. (1965) Structure of a ribonucleic acid. *Science* pp. 1462–1465.
39. Michel F, Costa M, Massire C, Westhof E (2000) [29] modeling RNA tertiary structure from patterns of sequence variation.
40. Noller HF, et al. (1981) Secondary structure model for 23S ribosomal RNA. *Nucleic acids research* 9(22):6167–6189.
41. Pace NR, Smith DK, Olsen GJ, James BD (1989) Phylogenetic comparative analysis and the secondary structure of ribonuclease P RNA—a review. *Gene* 82(1):65–75.
42. Williams K, Bartel D (1996) Phylogenetic analysis of tmRNA secondary structure. *RNA* 2(12):1306–1310.
43. Madhugiri R, Fricke M, Marz M, Ziebuhr J (2016) Coronavirus cis-acting RNA elements in *Advances in virus research*. (Elsevier) Vol. 96, pp. 127–163.
44. Van Den Born E, Posthuma CC, Gultyaev AP, Snijder EJ (2005) Discontinuous subgenomic RNA synthesis in arteriviruses is guided by an RNA hairpin structure located in the genomic leader region. *Journal of virology* 79(10):6312–6324.
45. Plant EP, Dinman JD (2008) The role of programmed-1 ribosomal frameshifting in coronavirus propagation. *Frontiers in bioscience: a journal and virtual library* 13:4873.
46. Goebel SJ, Hsue B, Dombrowski TF, Masters PS (2004) Characterization of the RNA components of a putative molecular switch in the 3' untranslated region of the murine coronavirus genome. *Journal of Virology* 78(2):669–682.
47. Goebel SJ, Miller TB, Bennett CJ, Bernard KA, Masters PS (2007) A hypervariable region within the 3' cis-acting element of the murine coronavirus genome is nonessential for RNA synthesis but affects pathogenesis. *Journal of virology* 81(3):1274–1287.
48. Liu P, Yang D, Carter K, Masud F, Leibowitz JL (2013) Functional analysis of the stem loop S3 and S4 structures in the coronavirus 3' UTR. *Virology* 443(1):40–47.
49. Robertson MP, et al. (2004) The structure of a rigorously conserved RNA element within the SARS virus genome. *PLoS Biol* 3(1):e5.
50. Seetin MG, Mathews DH (2012) RNA structure prediction: an overview of methods in *Bacterial Regulatory RNA*. (Springer), pp. 99–122.
51. Li TJ, Reidys CM (2018) The rainbow spectrum of RNA secondary structures. *Bulletin of mathematical biology* 80(6):1514–1538.
52. Lai WJC, et al. (2018) mRNAs and lncRNAs intrinsically form secondary structures with short end-to-end distances. *Nature communications* 9(1):1–11.
53. Ziv O, et al. (2018) COMRADES determines in vivo RNA structures and interactions. *Nature methods* 15(10):785–788.
54. Ziv O, et al. (2020) The short and long-range RNA-RNA interactome of SARS-CoV-2. *bioRxiv*.
55. Huston NC, Wan H, Tavares RdCA, Wilen C, Pyle AM (2020) Comprehensive in-vivo secondary structure of the SARS-CoV-2 genome reveals novel regulatory motifs and mechanisms. *BioRxiv*.
56. Sun L, et al. (2020) In vivo structural characterization of the whole SARS-CoV-2 rna genome identifies host cell target proteins vulnerable to re-purposed drugs. *bioRxiv*.
57. Lan TC, et al. (2020) Structure of the full SARS-CoV-2 RNA genome in infected cells. *bioRxiv*.
58. Cannone JJ, et al. (2002) The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC bioinformatics* 3(1):2.
59. Tabei Y, Tsuda K, Kin T, Asai K (2006) SCARNA: fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments. *Bioinformatics* 22(14):1723–1729.
60. Aghaeepour N, Hoos HH (2013) Ensemble-based prediction of RNA secondary structures. *BMC bioinformatics* 14(1):139.
61. Lorenz R, et al. (2011) ViennaRNA package 2.0. *Algorithms for Molecular Biology* 6(1):1.

Supporting Information

LinearTurboFold: Linear-Time RNA Structural Alignment and Conserved Structure Prediction with Applications to Coronaviruses

Sizhen Li, He Zhang, Liang Zhang, Kaibo Liu, Boxiang Liu, David H. Mathews and Liang Huang

A. Supporting Figures and Tables

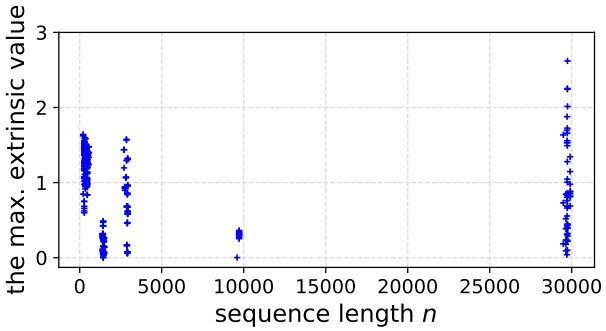


Fig. SI 1. The maximum values of the extrinsic information as a function of sequence length. The maximal value for each sequence is recorded when we ran LinearTurboFold on the collected dataset of sequence length ranging from 200 nt to 29,903 nt.

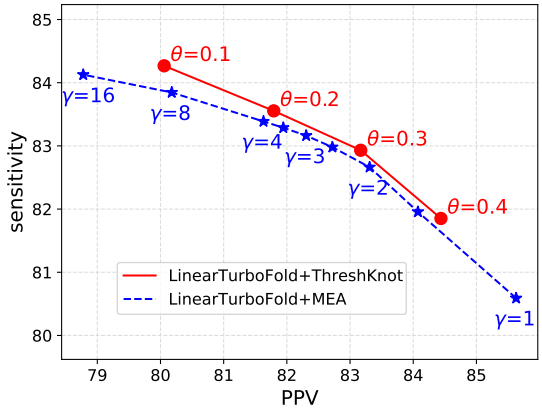


Fig. SI 2. Accuracy comparison between ThreshKnot and MEA on the training set.

	# of seqs		length				pairwise
Family	total	used	avg	max	min	cutoff	identity
SRP RNA	81	53	285.7	320	210	200	0.46
RNase P RNA	326	153	369.7	486	331	330	0.51
telomerase RNA	37	31	454.6	559	405	400	0.57
16S rRNA	2946	303	1440.2	1560	1401	1400	0.85
Overall	3,390	540	967.0	1560	210		0.65

Table SI 1. Statistics of the sequences in the RNAStrAlign test families used in this work. The last column represents the average pairwise sequence identity.

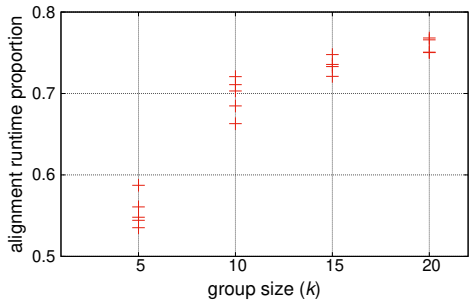


Fig. SI 3. The proportion of alignment runtime in the total runtime as the group size grows from 5 to 20.

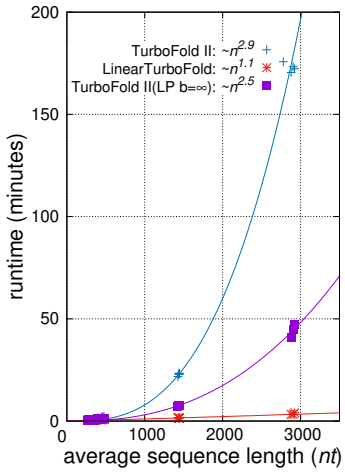


Fig. SI 4. LinearPartition uses thermodynamic parameters from Vienna RNAfold [61], which is a subset of RNAstructure partition [18]. By only replacing TurboFold II partition function with LinearPartition with an infinite beam size, the runtime decreases. This indicates a part of speedup of LinearTurboFold profits from a simplified energy model.

Table SI 2. 100% conserved base pairs across nine sequences that demonstrate compensating changes

	5'	3'	avg. prob.	Base pair (ref. seq.)	Compensating changes		5'	3'	avg. prob.	Base pair (ref. seq.)	Compensating changes
1	90	121	1.00	GC	GU/AU	78	12931	12964	0.99	UA	GC/AU
2	91	120	0.98	GU	GC/AU	79	13069	13108	1.00	UA	CG
3	97	115	1.00	AU	GC/GU	80	13078	13099	1.00	UA	UG/CG
4	101	111	0.99	GU	GC/AU	81	13216	13222	1.00	UA	UG/CG
5	153	291	1.00	UA	CG	82	13599	13628	1.00	UA	CG
6	159	282	0.94	GC	GU/AU	83	13638	13695	0.93	UA	AU
7	189	217	1.00	GC	AU	84	13641	13692	1.00	UA	CG
8	358	385	0.99	UA	CG	85	13707	13746	0.96	AU	GC
9	367	373	0.99	CG	UA	86	13831	13848	1.00	UA	CG
10	407	478	0.88	GC	AU	87	13878	13903	1.00	UG	CG/UA
11	442	448	0.99	CG	UG/UA	88	14161	14194	1.00	UA	UG/CG
12	570	616	0.93	AU	UA/CG	89	14169	14186	0.96	UG	UA/CG
13	652	724	0.98	AU	GC	90	14205	14211	0.80	AU	CG
14	670	709	0.99	UG	CG/UA	91	14224	14251	0.99	AU	GC/GU
15	880	889	0.99	AU	CG	92	14355	14361	1.00	AU	GC
16	970	981	0.99	GC	AU	93	14487	14532	1.00	AU	UG/CG/UA
17	1231	1251	0.99	GC	AU	94	14595	14604	0.99	UA	UG/CG
18	1237	1245	0.99	UG	UA/CG	95	15582	15607	1.00	AU	GC
19	2193	2200	0.91	GU	GC/AU	96	16023	16032	1.00	UA	AU
20	2278	2303	1.00	UA	CG	97	16080	16110	0.94	CG	UA
21	2855	2875	0.93	CG	UG/UA	98	16089	16101	1.00	GC	AU
22	2896	2923	1.00	UA	GU/AU	99	16125	16155	1.00	AU	UA
23	2959	2986	1.00	UA	UG/CG	100	16230	16236	1.00	CG	UA
24	3034	3061	0.91	UA	UG/CG	101	16677	16716	1.00	GC	AU
25	3712	3721	1.00	AU	GC	102	17241	17256	1.00	UA	CG
26	4096	4108	1.00	UA	UG/CG	103	17244	17253	1.00	AU	GC
27	4189	4225	1.00	CG	GC/UG	104	17304	17331	1.00	CG	UA
28	4603	4624	1.00	UA	UG/CG	105	17914	17925	0.95	AU	CG
29	4978	4987	1.00	UA	CG	106	18006	18054	1.00	UA	GU/AU
30	5164	5203	0.99	GC	GU/AU	107	18549	18561	1.00	AU	CG
31	5347	5374	1.00	UG	GC/AU/UA	108	18717	18774	1.00	UA	UG/CG
32	5356	5371	0.95	UA	GC/AU	109	19386	19419	1.00	CG	UG/UA
33	5417	5428	1.00	UA	AU	110	19395	19410	1.00	UA	CG
34	5476	5521	0.99	AU	GC/GU	111	19594	19613	0.96	CG	UA
35	5479	5518	1.00	UG	CG/UA	112	19707	19732	0.99	CG	UA
36	5482	5515	1.00	CG	UG/UA	113	19708	19731	1.00	AU	GC/GU
37	5549	5554	0.88	CG	UG/UA	114	19917	19953	1.00	UA	AU
38	5739	5770	1.00	GC	AU	115	19929	19941	1.00	UA	AU
39	6034	6055	0.99	AU	GC	116	19963	20012	1.00	AU	GC/GU
40	6037	6052	0.95	CG	UA	117	20172	20187	1.00	UA	CG
41	6103	6112	1.00	UA	CG	118	20217	20265	0.97	UA	CG
42	6154	6202	1.00	AU	GC	119	20223	20260	1.00	AU	GC
43	6328	6343	1.00	AU	UA	120	20353	20388	0.98	UA	CG
44	6364	6388	1.00	GC	AU	121	20523	20541	1.00	UA	CG
45	6367	6385	1.00	GC	AU	122	20622	20643	0.92	AU	GC
46	6458	6490	1.00	AU	GC	123	20841	20901	1.00	AU	GC/GU
47	6460	6488	1.00	UA	CG	124	21163	21201	1.00	AU	GC/GU
48	6592	6619	1.00	AU	GC	125	21300	21321	0.97	AU	GC/GU
49	6903	6922	0.69	CG	UA	126	21411	21423	1.00	CG	UA
50	6977	7006	0.96	GC	GU/AU	127	21513	21523	1.00	CG	UA
51	7480	7531	0.98	UA	UG/CG	128	22795	22810	1.00	UA	GU/AU
52	7864	7876	0.98	AU	GC/GU	129	23374	23383	1.00	UG	UA/CG
53	8146	8219	1.00	CG	UA	130	23531	23548	0.84	AU	GC
54	8147	8218	1.00	AU	GC/GU	131	23621	23647	1.00	GC	AU
55	8153	8212	1.00	UA	CG	132	23980	24088	1.00	AU	GC/GU
56	8317	8332	1.00	AU	GC/GU	133	23983	24085	1.00	AU	UA/CG

Table SI 2 continued from previous page

	5'	3'	avg. prob.	Base pair (ref. seq.)	Compensating changes		5'	3'	avg. prob.	Base pair (ref. seq.)	Compensating changes
57	8860	8881	1.00	CG	AU/UA	134	24121	24152	0.99	AU	GC
58	9046	9079	1.00	UA	UG/CG	135	24445	24487	1.00	GU	GC/AU
59	9055	9070	1.00	AU	GC	136	25016	25024	0.92	UA	CG
60	9427	9433	1.00	UA	CG	137	25336	25370	0.78	AU	GC/GU
61	9472	9511	1.00	AU	UA	138	25991	26004	1.00	GC	GU/AU
62	9689	9703	0.98	AU	GC	139	26091	26104	0.96	CG	UA
63	10213	10248	1.00	UA	CG	140	26145	26190	1.00	UA	CG
64	10225	10234	1.00	AU	GC	141	26630	26658	0.90	AU	GC
65	10651	10669	1.00	UA	CG	142	26676	26706	0.99	AU	GC
66	10711	10720	0.97	UA	CG	143	26939	26975	1.00	AU	CG
67	10864	10906	0.98	AU	GC	144	27412	27456	1.00	UA	UG/CG
68	10873	10898	0.99	AU	GC	145	27415	27453	0.99	GC	AU
69	10984	11026	1.00	AU	GC	146	27467	27484	1.00	GU	GC/AU
70	11275	11298	0.93	UA	UG/CG	147	27603	27613	0.99	CG	UA
71	11737	11763	1.00	UA	CG	148	27699	27744	0.98	UA	CG
72	11782	11803	0.97	AU	GC/GU	149	27717	27725	1.00	GC	GU/AU
73	11788	11797	1.00	CG	UA	150	28642	28664	1.00	UA	UG/CG
74	11971	12013	0.97	AU	GC/GU	151	28910	28930	0.99	AU	GC/GU
75	11989	11995	0.93	UA	UG/CG	152	29567	29597	1.00	AU	GC
76	12250	12280	1.00	AU	UG/UA	153	29635	29651	1.00	CG	AU
77	12538	12577	0.79	UA	UG/CG	154	29637	29649	1.00	UA	CG