LinearAlifold: Linear-Time Consensus Structure Prediction for RNA Alignments

Apoorv Malik[†,•] **Liang Zhang**[†,•] **Milan Gautam**[†] **Ning Dai**[†] **Sizhen Li**[†] **He Zhang**[†] **David H. Mathews**[◇,○,♣] **Liang Huang**[†,‡,•,*]

[†]School of EECS and [‡]Dept. of Biochemistry & Biophysics, Oregon State University, Corvallis, OR 97330, USA, [◇]Dept. of Biochemistry & Biophysics, [○]Center for RNA Biology, and [♣]Dept. of Biostatistics & Computational Biology, University of Rochester Medical Center, Rochester, NY 14642, USA, [•]Equal contribution. [*]Corresponding author: liang.huang.sh@gmail.com

Predicting the consensus structure of a set of aligned RNA homologs is a convenient method to find conserved structures in an RNA genome, which has many applications including viral diagnostics and therapeutics. However, the most commonly used tool for this task, RNAalifold, is prohibitively slow for long sequences, due to a cubic scaling with the sequence length, taking over a day on 400 SARS-CoV-2 and SARS-related genomes (~30,000nt). We present LinearAlifold, a much faster alternative that scales linearly with both the sequence length and the number of sequences, based on our work LinearFold that folds a single RNA in linear time. Our work is orders of magnitude faster than RNAalifold (0.7 hours on the above 400 genomes, or ~36× speedup) and achieves higher accuracies when compared to a database of known structures. More interestingly, LinearAlifold's prediction on SARS-CoV-2 correlates well with experimentally determined structures, substantially outperforming RNAalifold. Finally, LinearAlifold supports two energy models (Vienna and BL*) and four modes: minimum free energy (MFE), maximum expected accuracy (MEA), ThreshKnot, and stochastic sampling, each of which takes under an hour for hundreds of SARS-CoV variants. Our resource is at: `https://github.com/LinearFold/LinearAlifold` **(code) and** `http://linearfold.org/linear-alifold` **(server).**

INTRODUCTION

Ribonucleic acids (RNA) are involved in many cellular processes ([1], [2], [3]), and most of RNA secondary structures are highly conserved across evolution to maintain their functionalities in spite of changes to the sequence ([4], [5], [6]). Thus, predicting the *consensus structure* for a set of aligned RNA homologs is more accurate than predicting the structure for a single sequence and it is useful for identifying conserved regions, which can be used for diagnostics and therapeutics. For this task, RNAalifold ([7], [8]) is a widely used tool to predict consensus structures for aligned RNA homologs that considers both thermodynamic stability and sequence covariation. However, its cubic runtime (against sequence length n) makes it difficult to be applied to long sequences such as SARS-CoV-2 genomes ($n \simeq 30{,}000nt$), requiring over a day for 400 such genomes. As an alternative, LinearTurboFold ([9]) is an iterative fold-and-align tool (thus does not need alignment as input) that scales linearly with sequence length, but quadratically with the number of sequences (k). This limits its use case to only about 30 SARS-CoV-2 variants while it is often helpful to include hundreds of such genomes to account for as much sequence variation as possible, as new variants emerge rapidly. So there is a critical need to develop a fast consensus folding tool that scales linearly with *both* n and k. On the other hand, beyond predicting minimum free energy (MFE) consensus structures, it is also useful to calculate the *consensus partition function* and *consensus base-pairing probabilities* (BPPs), which are widely used in many downstream tasks such as maximum expected accuracy (MEA) folding ([10], [11], [12]), ThreshKnot ([13]), and stochastic sampling from the ensemble ([14], [15]). However, RNAalifold's partition function mode is even slower than its MFE mode (often by 10× or more), and both its partition function and stochastic sampling modes fail to run on SARS-CoV-2 (for any $k > 1$) due to overflow.

To alleviate this slow runtime, one can use local folding to predict structures in linear time, but inevitably giving up non-local interactions. Those base pairs, especially the end-to-end ones, are known to be prevalent in most RNAs ([16], [17]). In particular, the base pairs between the 5' and 3' untranslated regions (UTRs) of SARS-CoV-2, across ~30,000 nucleotides, are found by both purely experimental methods ([18]) and purely computational ones ([9]). How can we achieve linear runtime without without sacrificing long-distance pairs?

Here we report LinearAlifold, an efficient tool for consensus structure prediction that scales linearly with *both* the sequence length (n) and the number of aligned sequences (k) without any constraints on pair distance, building upon on our previous work LinearFold ([19]) and LinearPartition ([20]) for single sequence folding (Fig. 1A). Being orders of magnitude faster than RNAalifold, our work can fold hundreds of full-length coronavirus genomes under an hour and can recover end-to-end pairs. For example, it takes only 0.7 hours to fold the above-mentioned $k = 400$ SARS-CoV sequences, compared to 25.7 hours by RNAalifold (~36× speedup). Meanwhile, LinearAlifold significantly outperforms RNAalifold in structure prediction accuracy compared to a database of known structures of homologous sequences ([21]) (Fig. 2). More importantly, LinearAlifold's predictions

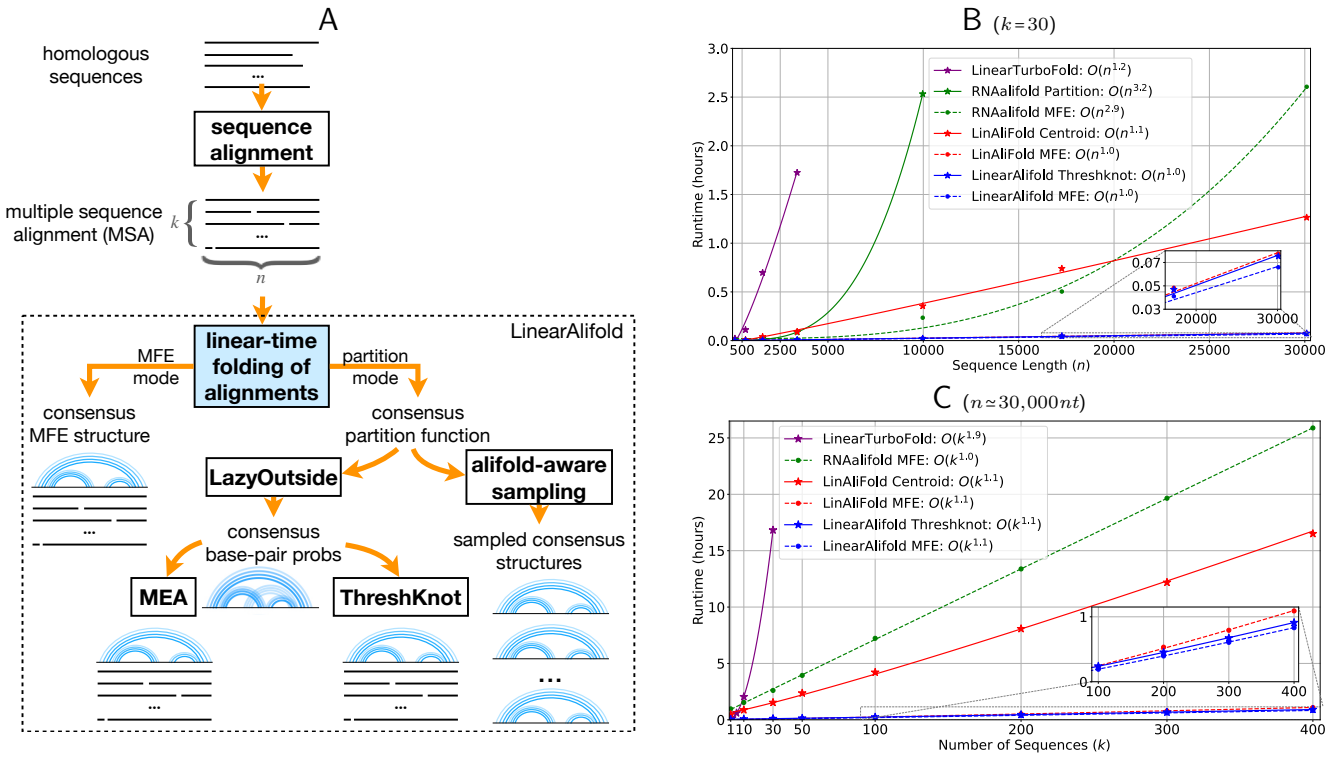


Figure 1. **A**: Overview of LinearAlifold, which takes aligned homologous sequences as input to predict consensus MFE structure, consensus partition function, and consensus base-pairing probabilities, which are used in downstream tasks such as Maximum Expected Accuracy (MEA) folding, ThreshKnot folding, and stochastic sampling from the ensemble. **B**: Runtime of various tools against sequence length (n) for $k=30$. **C**: Runtime of various tools against the number of sequences (k) for $n\simeq 30{,}000nt$.

on hundreds of SARS-CoV genomes (under an hour) correlate better with the experimentally guided structures (18, 22) than RNAalifold's (over a day) (Fig. 3). In addition to MFE folding, LinearAlifold also supports partition function, base-pairing probabilities, ensemble-based structure prediction methods MEA and Threshknot, and stochastic sampling, all of which take under an hour on hundreds of SARS-CoV variants (which RNAalifold fails due to overflow).

LinAliFold (23) is another linear-scaling tool for consensus folding, developed roughly in parallel with our initial version but published earlier.[1] Like our work, LinAliFold is also built upon our previous work LinearFold and LinearPartition, and thus also achieves linear runtime with both n and k. Unlike our work, their partition-function mode uses CentroidFold (24) and is much slower than our tool, especially with large k (for example, for $k=400$ SARS-related genomes, ours takes 0.8 hours compared to their 16.4 hours, or $\sim$20× speedup). In fact, our partition function mode is even faster than their MFE mode (Fig. 1C) thanks to the use of LazyOutside (25) (see Methods). In addition, our tool supports MEA, ThreshKnot (thus our output can contain pseudoknots), and stochastic sampling, none of which is available in their tool. Moreover, we support two different Turner-style energy models, the Vienna model in RNAalifold and the BL* model (26), while LinAliFold only supports the latter. More importantly, we also built an easy-to-use web server at `http://linearfold.org/linear-alifold`.

[1] Our initial arXiv preprint (2022) was discussed in their work.

RESULTS

Like RNAalifold, our LinearAlifold also takes a multiple-sequence alignment (MSA) as input (Fig. 1A) and outputs an MFE consensus structure or a consensus partition function. The scoring function in these systems is a combination of thermodynamic free energies and sequence covariation scores (7, 8) (see Methods). We employed the beam pruning heuristic to reduce the complexity from cubic runtime (against n) to linear time, inspired by LinearFold (19). The basic idea of the heuristic algorithm is, at each step j, we only keep the b top-scoring states and prune the other ones, which are less likely to be part of the optimal final structure. This approximate search algorithm helps reduce the time complexity from $O(kn^3)$ to $O(knb^2)$. In the MFE mode, we further reduced the time complexity to $O(knb\log b)$ following the k-best parsing idea (27). The default beam size is 100, following LinearFold and LinearPartition. Thus, we reduced the time complexity from $O(kn^3)$ (RNAalifold) to $O(kn)$ (LinearAlifold).

In the partition function mode, LinearAlifold computes in $O(knb^2)$ time the consensus partition function in an

"inside phase", which is followed by an "outside phase" to compute the consensus base-pairing probabilities (BPPs) (Fig. 1A). Normally, the outside phase takes the same amount of time as inside, but we employ our (unpublished) technique LazyOutside (25) which takes only ~1.5% of the inside time, making inside-outside calculation almost as fast as inside only (and similar to the MFE mode); see Methods for details. Our tool supports two BPP-based structure prediction methods, MEA and ThreshKnot, both of which are more accurate than MFE. From the consensus partition function, our tool also supports alifold-aware stochastic sampling, based on our LazySampling algorithm. These sampled structures are useful to "visualize" the Boltzmann ensemble, and can be used to compute the accessibility of arbitrary regions (9).

LinearAlifold supports two energy models, Vienna (as in RNAalifold) and BL* (as in LinAliFold). The latter is our default model which generally has higher accuracy on our COVID benchmark (note that COVID data is disjoint from BL*'s training set).

Scalability

To demonstrate the scalability of our work, we prepared a set of RNA sequences that contains 8 families ($n \simeq 1{,}600nt$ or less) from RNAStralign (21), 23s rRNA ($n \simeq 3{,}300nt$) from the Comparative RNA Web (CRW) site (28), and long sequences (from ~$9{,}800nt$ to ~$30{,}000nt$) from three viruses[2] from NCBI and GISAID[3]. We used MAFFT (29) (with `--auto`) to align the input sequences.

Figs. 1B–C compare the runtime of three align-then-fold tools (RNAalifold, LinAliFold, and LinearAlifold) and one iterative align-and-fold tool (LinearTurboFold). As shown in Fig. 1B, for a given k (here $k=30$), LinearAlifold scales linearly with sequence length n and is substantially faster than RNAalifold (which scales roughly cubically with n) under either MFE or partition function modes. In the MFE mode, on the SARS-CoV family, LinearAlifold is ~40.8× faster than RNAalifold (3.8 min. vs. 2.6 hours) In the partition function mode, RNAalifold cannot scale to $n > 14{,}000nt$ for $k=30$ due to overflow. On the HIV family ($\simeq 9{,}800nt$), LinearAlifold's ThreshKnot mode is ~120× faster than RNAalifold's MEA mode (1.2 min. vs. 2.5 hours)

We also tested runtime against the number of homologs (k) using SARS-CoV ($n \simeq 30{,}000nt$) (Fig. 1C). Here all three align-then-fold tools (RNAalifold, LinAliFold, and LinearAlifold) scale linearly with k, but the iterative align-and-fold tool LinearTurboFold scales quadratically with k, making it only feasible on ~30 SARS-related genomes. For $k=400$, RNAalifold requires more than a day (25.7 hours) while our tool only needs 0.7 hours (~36.3× speedup). In addition, RNAalifold partition function mode fails to run on SARS-CoV due to overflow.

[2] HIV ($n \simeq 9{,}800nt$), RSV (Respiratory syncytial virus, $n \simeq 15{,}000nt$), and SARS-CoV ($n \simeq 30{,}000nt$) genomes

[3] `www.ncbi.nlm.nih.gov` and `www.gisaid.org`

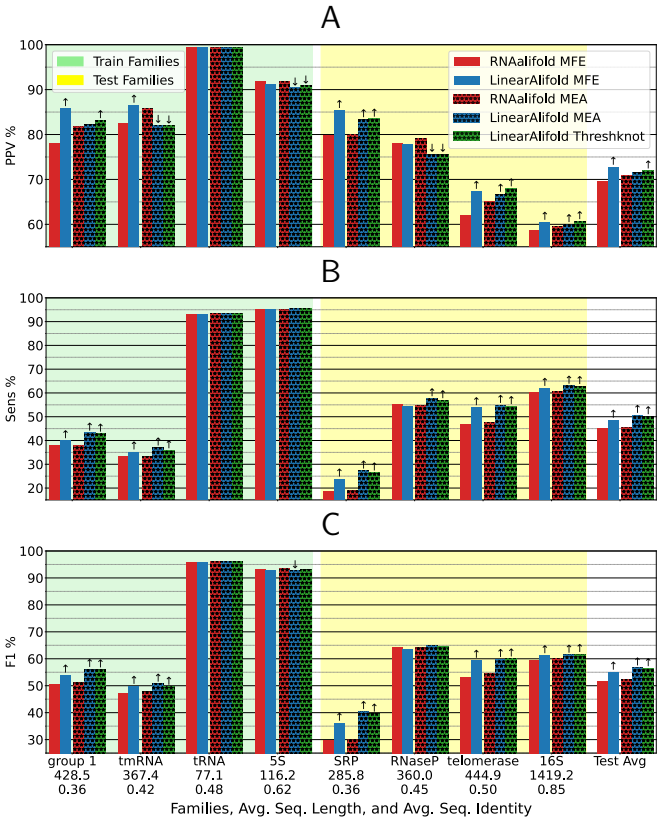


Figure 2. Accuracy comparisons between RNAalifold and LinearAlifold; each family has 10 samples and each sample is has $k=30$ homologs. Statistical significance (two-sided) is marked as '↑' if LinearAlifold is significantly better, or '↓' if RNAalifold is significantly better ($p<0.05$). See also Fig. S1.

Although LinAliFold also scales linearly with both n and k, our tool is still substantially faster, especially in the partition function mode. This is due to two reasons: (a) we employ LazyOutside (25) which reduces the outside phase to just 1–2% of the inside phase, bringing a ~2× speedup; and (b) their partition function mode uses CentroidFold, which mixes consensus BPPs with individual single-sequence BPPs (i.e., calling LinearPartition k times). As a result, on $k=400$ SARS-related genomes, our LinearAlifold ThreshKnot takes 0.8 hours compared to their 16.4 hours (~20× speedup). Actually, our partition function mode is even faster than their MFE mode (0.8 vs. 1 hour(s)).

Accuracy

We compared the accuracies of secondary structure prediction using the RNAStralign database (21), which have well-determined secondary structures of RNA homologs for eight families (Fig. 2). For each family, we take 10 samples, each of which contains $k=30$ sequences. These sequences in each sample were first aligned using MAFFT (`--auto`) before being fed into RNAalifold and LinearAlifold (both using the Vienna energy model). Following LinearTurboFold, we used the first four families (tRNA, 5S rRNA, tmRNA, and Group I Intron) to

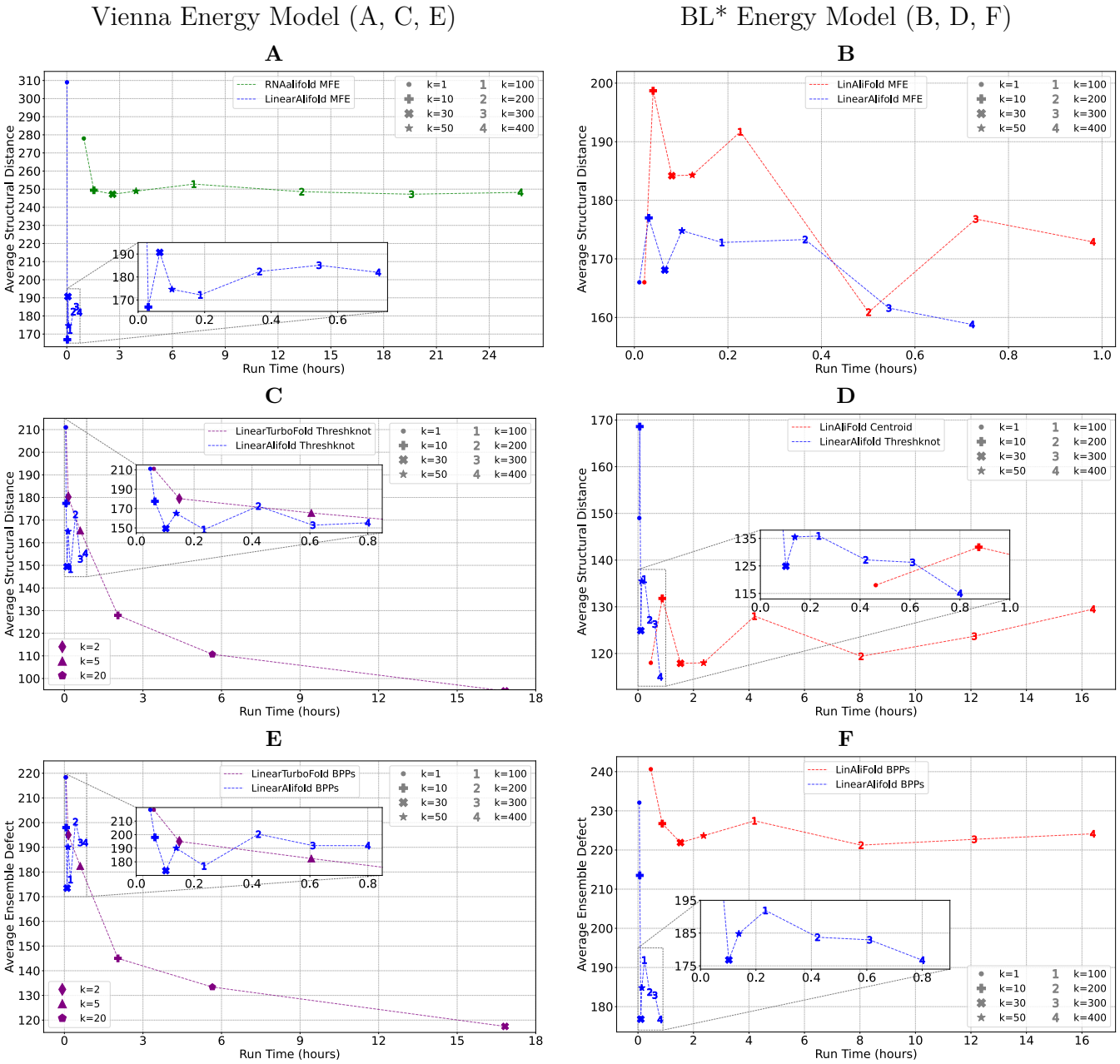


Figure 3. Structural distance and ensemble defect against run time for different energy models and different methods. The curves show the mean values over 10 samples for each k A–B: MFE prediction. C–D: partition-based structure prediction. E–F: ensemble quality. See Fig. S3 for another version which shows more statistics of each 10 samples and uses k as the x-axis.

tune the hyperparameters (see Methods), so the "Test Avg" columns include the remaining four families (SRP, RNaseP, telomerase, and 16S rRNA).

In terms of F1 score, LinearAlifold's MFE and MEA modes significantly outperform the corresponding modes of RNAalifold on all test families, and LinearAlifold's ThreshKnot mode significantly outperforms RNAalifold's MEA mode on almost all test families except for RNaseP (two-sided significance test ([30])). This high accuracy of LinearAlifold over RNAalifold is expected, and is due to the beam search in the former, which is inherited from LinearFold ([19]). As we showed in our LinearFold paper, although beam search introduces minor search errors and returns suboptimal structures in terms of the scoring function, it nevertheless makes the search more robust locally (since the scoring function is never perfect), which translates to slightly better accuracy compared to ground-truth structures. We observe this phenomenon over and over in our previous work LinearFold, LinearPartition ([20]), LinearSampling ([31]), LinearCoFold ([32]), and LinearTurboFold ([9]), as well as our earlier work in natural language parsing ([33]) which gave rise to LinearFold, so this is a universal phenomenon.

Since the BL* energy model is trained on structures which overlap with our benchmark, it overfits on it. Thus we do not include our results with BL* (nor a comparison with LinAliFold using the same energy model). Figs. S1 and S2 compare more systems including LinearTurboFold, LinAlifold, and single-sequence folding.

Note that align-then-fold systems (RNAalifold, LinearAlifold, and LinAliFold) tend to be inaccurate for low sequence indentity families (e.g., SRP and group 1) and tend to be more accurate for high sequence identity families (e.g., 16S rRNA).

Consensus Structure Prediction in SARS-CoV-2 and SARS-related Betacoronaviruses

It is known that conserved structures across mutations are critical for viruses to maintain their functions to survive. Thus, these conserved regions could be potential targets for diagnostics and therapeutics (4, 5, 6). To model consensus structures for SARS-CoV-2 and SARS-related betacoronaviruses, for each k ranging from 10 to 400, we sampled 10 sets of diverse sequences (see Methods for details), and used MAFFT `--auto` to generate 10 MSAs for each k. Following LinearTurboFold[4], the ratio of the number of SARS-CoV-2 to the number of SARS-related genomes remains 6 to 4 in all samples.

To evaluate the reliability of LinearAlifold's prediction on SARS-CoV-2, we compared the predicted structure with experimental studies (18, 22) for the well-known 5' and 3' UTR regions. Huston et al. (22) modeled secondary structures guided by the chemical probing data, but used a local folding method for prediction because the sequence length of SARS-CoV-2 is out of reach of most algorithms. As a result, long-range interactions were fully abandoned in their prediction, which are critical for regulating the viral transcription and replication pathways (17, 18). To overcome this issue, we further involved a purely experimental study of Ziv et al. (18), which can detect long-range interactions between 5' and 3' UTRs. Therefore, to take into consideration both local and global structures between 5' and 3' UTRs, we built a hybrid structure model (Fig. 4H) by combining Huston et al. and Ziv et al.'s work (see Methods).

Fig. 3 compares the quality of predictions from four tools (RNAalifold, LinAliFold, LinearTurboFold, and LinearAlifold), two energy models Vienna (A/C/E) and BL* (B/D/F), and three modalities (MFE (A–B), partition-based structure prediction (ThreshKnot/Centroid, C–D), and base-pairing probabilities (E–F)). The metrics are structural distance (the number of incorrectly predicted nucleotides) and ensemble defect (the expected structural distance over the Boltzmann ensemble), both the lower the better (closer to the above hybrid structure model). The x-axes in these plots are run time, showing the speed advantage of our tool over others.

[4]The LinearTurboFold paper (9) built a dataset of 25 SARS-CoV genomes: 16 SARS-CoV-2 plus 9 SARS-related sequences.

In Fig. 3A, our MFE is substantially faster and more accurate than RNAalifold MFE (both with Vienna energy model), and in panel B, our MFE is noticeably faster and more accurate than LinAliFold MFE (both with BL* energy model). Next, Figs. 3C–D compare our tool with LinearTurboFold and LinearAlifold in terms of partition-function-based structure prediction (note that as mentioned before, RNAalifold's partition function mode does not run on SARS-CoV genomes). In Fig. 3C, the iterative align-and-fold tool LinearTurboFold achieves substantially better structural distance than our align-then-fold tool, presumably due to folding-aware alignment, but at the cost of much slower run time and inability to scale beyond $k=30$. In Fig. 3D, LinAliFold Centroid mode achieves similar structural distance as our ThreshKnot mode, but takes ~20× more time due to mixing with single-sequence BPPs. Finally, Figs. 3E–F are similar to C–D, but instead of evaluating one predicted structure, they evaluate the quality of the whole ensemble, measured by the ensemble defect computed using the base-pairing matrix. The only difference is that in F, LinAliFold's ensemble quality (still with the same mixing in D) is substantially worse than ours, suggesting that CentroidFold was able to extract a high quality structure from a lower quality ensemble.

Across the board, the BL* column (B/D/F) is consistently better than the Vienna column (A/C/E), so we choose BL* as the default energy model, but the user can change it with a command-line switch.

Fig. S3 is similar to Fig. 3 but uses k as the x-axis, and draws the 25-75 quantile boxes (and medians) in addition to the mean curves, since we have 10 samples for each k. Fig. S4 is similar to Fig. S3 but uses Huston et al.'s model structure instead of the hybrid structure as the reference.

To further visualize our predicted structures, we choose one particular sample (#5/10) for $k=30$ SARS-CoV-2 and SARS-related genomes; this k is chosen because it is the largest for LinearTurboFold to run, and this particular sample is chosen because our LinearAlifold BL* prediction (our default setting) achieves the best structural distance (against the hybrid structure). Fig. 4A–C compare the base-pairing probabilities (BPPs) for three systems: LinearAlifold Vienna model, LinearAlifold BL* model, and LinearTurboFold (Vienna model). Here we use Ziv et al.'s ranges as references, and blue arcs indicate pairings supported by at least one Ziv et al. arc, and red ones are not supported by any Ziv et al. arc. We can see that LinearAlifold systems predict many more non-local (long-distance) pairing possibilities, although most of them are incorrect, and LinearTurboFold mostly predicts local pairings. Fig. S5 shows the corresponding ThreshKnot predictions (grouped by pairing distance) and their precision against Ziv et al. ranges. We can see that LinearAlifold's both models predicted about 2,000 non-local pairs ($\geq$100 nt), among which 36.4% of the prediction by BL* model and 32.3% of the prediction by Vienna model are supported by at least one Ziv et al. ranges, respectively. LinearAlifold BL* model also predicted 14 end-to-end

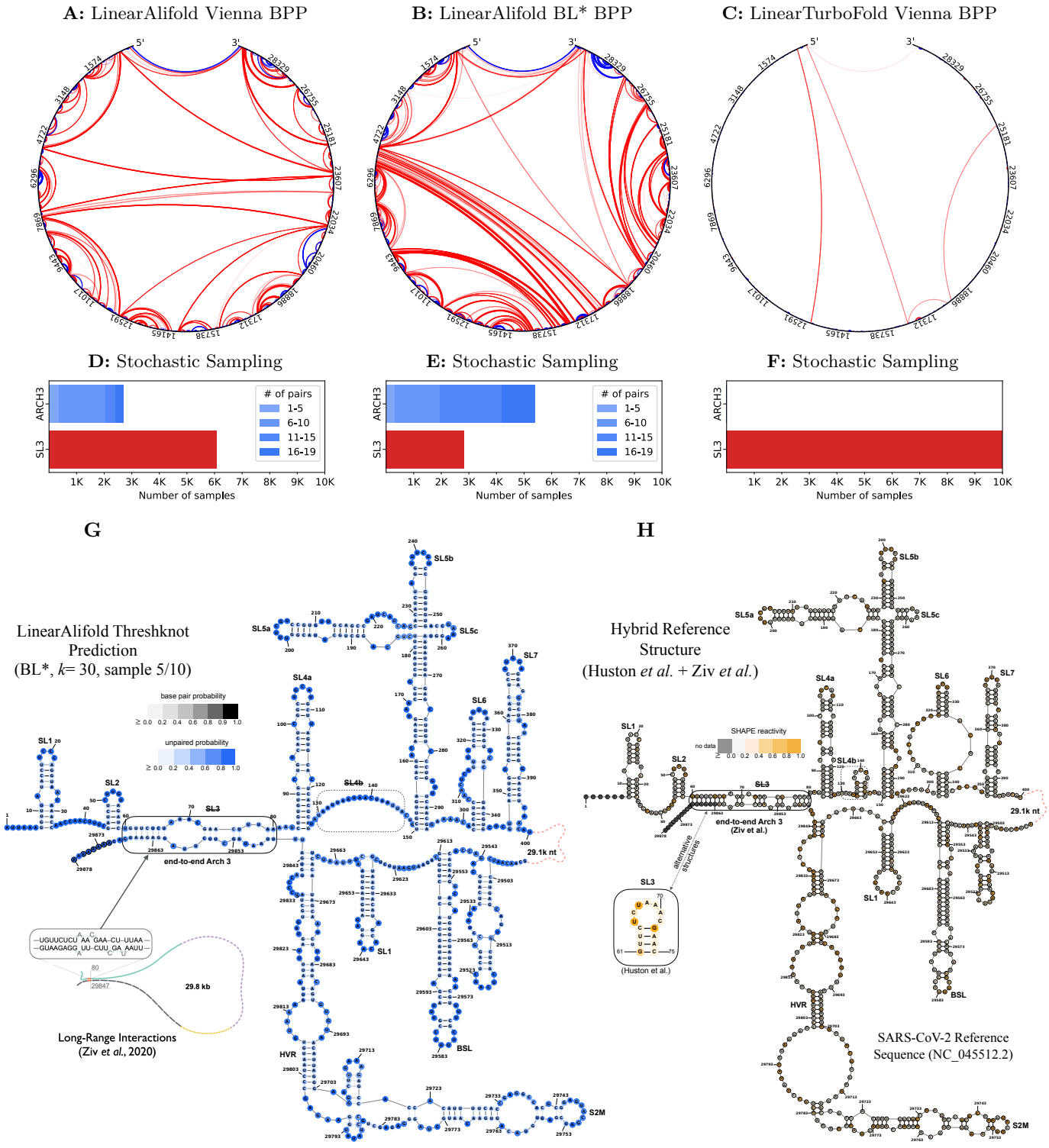


Figure 4. Visualizations of structure predictions on $k = 30$ SARS-CoV genomes (A–G) compared with the experimentally-guided hybrid structure (H). A–C: Circular plots of base-pairing probabilities (BPPs) from LinearAlifold (two energy models) and LinearTurboFold on $k = 30$ genomes (sample 5/10). Blue arcs are consistent with at least one range from Ziv et al. (18), while red arcs are not supported by any such range. The darkness of the arcs indicates pairing probability. D–F: stochastic sampling statistics (over 10,000 structures) between the competing global (arch 3 from Ziv et al.) and local (SL3 from Huston et al. (22)) structures. G: the 5' and 3' UTR structures of LinearAlifold (BL*) ThreshKnot prediction, with shades of blue for unpaired probabilities of each nucleotide and shades of black for pairing probabilities for each pair. H: the reference hybrid structure based on Huston et al.'s SHAPE-guided model but with the end-to-end arch 3 from Ziv et al. replacing SL3.

pairs which are all supported by Ziv et al., whereas the other two systems did not predict any end-to-end pairs.

More interestingly, we would like to further investigate the competition between alternative structures in the Boltzmann ensemble, in particular, the end-to-end arch 3 (from Ziv et al.) vs. the local SL3 in 5' UTR (from Huston et al.). Fig. 4D–F conduct stochastic sampling for LinearAlifold Vienna, LinearAlifold BL*, and LinearTurboFold. Interestingly, LinearAlifold BL* prefers end-to-end arch 3 (but with about 30% of sampled structures showing SL3), while LinearAlifold Vienna prefers SL3 (about 60% of sampled structures). LinearTurboFold, however, is 100% SL3.

Finally, Fig. 4G shows the 5' and 3' UTR structure of the LinearAlifold BL* ThreshKnot prediction, which is very similar to the hybrid reference structure in Fig. 4H. It is also worth noting that, unlike Huston et al.'s experimentally guided model, LinearAlifold BL* ThreshKnot predicts the SL4b region to be single-stranded (Fig. 4G), which is consistent with the experimentally guided structure by Sun et al. (34, Fig. 2C). These results, plus the fact that the prediction from the LinearTurboFold paper (9, Fig. 3) has rather weak and different pairs for SL4b, all suggest alternative structures in the ensemble for that region.

DISCUSSION

Considering the fast mutation rate of RNA viruses such as SARS-CoV-2, accurately identifying conserved regions from homologs is critical to develop mutation-insensitive diagnostics and therapeutics. Consensus folding algorithms, which can take hundreds of aligned homologs to predict consensus structure, are widely-used for this task. However, RNAalifold, the most widely used consensus folding tool, scales cubically with the sequence length in runtime, and is prohibitively slow to analyze long RNAs, especially SARS-CoV-2 (~30,000 nt). To alleviate this issue, we present LinearAlifold, an efficient tool which scales linearly with both the sequence length (n) and the number of aligned sequences (k). We confirmed that LinearAlifold is orders of magnitude faster than RNAalifold, taking less than an hour to fold 400 full-length SARS-CoV genomes (which takes more than a day for RNAalifold MFE mode). We also demonstrated that LinearAlifold achieves significantly higher accuracies on a benchmark dataset with known structures. LinearAlifold is also faster than a similar linear-time consensus folding tool LinAliFold, especially in the partition function mode and for a larger k.

LinearAlifold has four output modalities: (1) predicting consensus minimum free energy structure (MFE mode); (2) predicting the MEA structure based on the consensus BPP; (3) predicting the ThreshKnot structure based on the consensus BPP; and (4) stochastically sampling structures from the consensus partition function. All these modes can be applied to hundreds of aligned SARS-CoV-2 homologs, while RNAalifold can only handle the MFE mode for such MSAs due to overflow, and LinAliFold only supports (1) and a variant of (3) (CentroidFold). LinearAlifold's prediction on SARS-CoV-2 correlates better with experimentally-guided structures than RNAalifold's or LinAliFold's, yet takes substantially less time.

LinearAlifold is a general algorithm and can also be applied to analyze other long RNA viruses, such as HIV, WNV (West Nile Virus), and Ebola. Finally, we built a web server which will be useful for biologists.

METHODS

Scoring function of RNAalifold and LinearAlifold

Following RNAalifold, for a set of k aligned sequences S, our scoring function takes into consideration both a thermodynamic energy model and a sequence covariation score $\gamma(i,j,S)$ to evaluate the corresponding alignment column pair (i,j)'s compensatory mutations:

$$score(S,y) = \frac{1}{k}\Big[\sum_{s\in S}\Delta G(s,y) + \beta \sum_{(i,j)\in y}\gamma(i,j,S)\Big]$$

where y is a consensus secondary structure, $\Delta G(s,y)$ is the free energy of sequence s folded into structure y (when mapping consensus structure y on to an individual sequence s, we remove the pairs in y that are not pairable on s), and $\gamma(i,j,S)$ is the *(base pair) conservation score* that evaluates the corresponding alignment columns with respect to evidence for base pairing

$$\gamma(i,j,S) = \frac{1}{k}\gamma'(i,j,S) + \delta\sum_{s\in S}\begin{cases}0 & \text{if } (s_i,s_j)\in\mathcal{P}\\ 0.25 & \text{if } (s_i,s_j)=(-,-)\\ 1 & \text{otherwise}\end{cases}$$

where $\mathcal{P}=\{\text{GC},\text{CG},\text{AU},\text{UA},\text{GU},\text{UG}\}$ is the set of possible base-pairs, $-$ is a gap, $\gamma'(i,j,S)$ evaluates covariance bonuses and penalties. We follow the 2008 version of RNAalifold (8) to use the (symmetric) RIBOSUM matrix R to calculate the covariance, which replaces the Hamming distances $h(s_i,s'_i)$ and $h(s_j,s'_j)$ from the 2002 version of RNAalifold (7):

$$\gamma'(i,j,S) = \frac{1}{2}\sum_{s,s'\in S, s\neq s', (s_i,s_j)\in\mathcal{P}, (s'_i,s'_j)\in\mathcal{P}} R(s_is_j;s'_is'_j)$$

The RIBOSUM matrix R is selected from a pool of matrices based on the minimum and maximum pairwise sequence identities in the MSA. In the special case where there is no sequence variation, all values in R are set to 0 so that LinearAlifold falls back to LinearFold and LinearPartition. The basic idea of $\gamma'(i,j,S)$ is to reward *compensatory mutations* on column-pair (i,j) across all sequences. For example, on (i,j) columns, if some sequences are AU pairs while others are CG pairs, it is a stronger signal for (i,j) pairing than if all sequences are the same type of pairs. It is important note that the default version of both RNAalifold and LinAliFold still use the 2002 version of $\gamma'(i,j,S)$ but the 2008 version

is substantially more accurate (which can be invoked by a command-line switch `-r` in RNAalifold and `-r 1` in LinAliFold), so we only implemented the 2008 version. All the RNAalifold and LinAliFold results in this paper also used the 2008 version. The tunable parameters β and δ are both set to be 1 in RNAalifold and LinAliFold, but here we tune them using the BL* energy model on the four training families of RNAstralign (tRNA, 5S rRNA, tmRNA, Group I Intron) and the best setting is $\beta=1.2$ and $\delta=0.1$. For example, if we have this simple MSA as input:

```
>seq1
CCCAAAGGG
>seq2
GGGAAACCC
```

LinearAlifold's default output will be:

```
Minimum Free Energy: -3.54 kcal/mol

MFE Structure:
(((...))) (-3.54 = -0.50 + -3.04)
```

Here `-0.50` is the thermodynamic energy $\frac{1}{k}\sum_{s\in S}\Delta G(s,y)$ and `-3.04` is the conservation score $\frac{1}{k}\beta\sum_{(i,j)\in y}\gamma(i,j,S)$.

Correct Calculation of Covariance Bonus $\gamma'(i,j,S)$

The naive calculation of $\gamma'(i,j,S)$ for each (i,j) column-pair would take $O(k^2)$ time because we need to enumerate all sequence pairs, but RNAalifold employs a clever method that reduces to $O(k)$ by counting the number of sequences for each type of pair. For example, among $k=8$ sequences, for this (i,j) column-pair, assume we have 5 sequences with CG pairs and 3 with AU pairs, then we can calculate $\gamma'(i,j,S)$ by aggregating over groups of sequences with the same pair type instead of enumerating all $8\cdot 7/2$ sequence-pairs:

$$\gamma'(i,j,S) = 5\cdot 3\cdot R(\mathrm{CG};\mathrm{AU}) + \frac{5\cdot 4}{2}R(\mathrm{CG};\mathrm{CG}) + \frac{3\cdot 2}{2}R(\mathrm{AU};\mathrm{AU})$$

Or more generally, let $f_{i,j}[t]$ denote the number of sequences with pair type t at (i,j) columns ($t\in\mathcal{P}$), then

$$\gamma'(i,j,S) = \sum_{t,t'\in\mathcal{P},t\neq t'} f_{i,j}[t]\cdot f_{i,j}[t']\cdot R(t;t') + \sum_{t\in\mathcal{P}}\binom{f_{i,j}[t]}{2}\cdot R(t;t)$$

The first term calculates the contribution from compensatory mutations (different pair types t and t') and the second term calculates the contribution from the same pair type t.

However, it is worth noting that both RNAalifold and LinAliFold calculated this term incorrectly.

1. RNAalifold (and LinAliFold by inheritance) uses an oversimplified formula:

   $$\gamma'(i,j,S) = \sum_{t,t'\in\mathcal{P}} f_{i,j}[t]\cdot f_{i,j}[t']\cdot R(t;t')$$

   which incorrectly handles the score contributions for sequences that have the same base pair type t at positions (i,j). The correct method (as shown above) should multiply the RIBOSUM score $R(t;t)$ by $\binom{f_{i,j}[t]}{2}$ which reflects the correct number of pairwise comparisons without repetition among sequences while RNAalifold and LinAliFold erroneously calculates this as $(f_{i,j}[t])^2\cdot R(t;t)$. Note that this miscalculation also includes comparisons of each sequence with itself, i.e., pairs $(s_is_j;s'_is'_j)$ where $s=s'$, which should not contribute to the score since they do not provide information about covariation. This overcounting inflates the conservation score, leading to potentially incorrect results in RNA structural predictions.

2. Another computational error in RNAalifold is the normalization of the $\gamma'(i,j,S)$ term. The correct approach should normalize this term by k^2, reflecting the total number of pairwise sequence comparisons. However, RNAalifold incorrectly normalizes this term by just k. This insufficient normalization leads to amplified contributions from sequence pairs, therefore distorting the score. LinAliFold corrected this issue and computes the normalization correctly.

Partition Function Mode

The consensus partition function $Q(S)$ over a set S of aligned sequences is:

$$Q(S) = \sum_y \exp(-score(S,y)/RT)$$

where R is the molar gas constant and T is the absolute temperature. The Boltzmann probability of a consensus structure y is then:

$$p(y|S) = \frac{\exp(-score(S,y)/RT)}{Q(S)}$$

and the consensus (marginal) base-pairing probability that column i is paired with column j is:

$$p_{ij}(S) = \sum_{y:(i,j)\in y} p(y|S)$$

When projecting this consensus base-pairing matrix down to each individual sequence, we delete columns and rows that are dashes (–) in that sequence, as well as p_{ij} entries that correspond to non-pairable bases in that sequence.

LazyOutside Algorithm

By default, the inside-outside algorithm (35) is used to calculate the marginal base-pairing probabilities, where the McCaskill algorithm (36) is a special case. Conventionally, the outside phase is considered a mirror image of the inside phase, with similar or slower runtimes, which means inside-outside is (at least) twice as slow as their inside-only or MFE. We employ our unpublished technique of LazyOutside (25) which is a lazy (on-demand) algorithm that only visits high-probability states and ignores the low-probability ones. Basically, let us denote $\alpha(v)$ to be the inside partition function for node v (e.g., $P_{5,10}$) and $\beta(v)$ to be the outside partition function, then we prune nodes v if its marginal probability falls under a threshold θ:

$$\alpha(v)\cdot\beta(v)/Q(S)<\theta$$

and we use the default $\theta=5\times10^{-5}$. This pruning was also used in natural language parsing ("relative useless pruning") and machine learning ("max-marginals") (37). As a result, it only visits a tiny fraction (often as small as 1%) of the states visited in the inside phase, which implies up to 100× speedup of the outside phase, making inside-outside almost as fast as the inside phase alone.

Structural Distance and Ensemble Defect

We employ structural distance and ensemble defect (38) as two key metrics to evaluate the prediction accuracy of our tool. Structural distance is basically a structured version of Hamming distance between two structures, while ensemble defect is the expectation of structural distance in the Boltzmann ensemble.

More formally, let $\boldsymbol{x}$ be an RNA sequence and $\boldsymbol{y}$ and $\boldsymbol{y}^*$ be two secondary structures of $\boldsymbol{x}$. The structural distance between $\boldsymbol{y}$ and $\boldsymbol{y}^*$ quantifies the structural discrepancies between them, specifically in terms of mismatched base pairs and unpaired nucleotides, calculated using the following formula:

$$\begin{aligned} d(\boldsymbol{y},\boldsymbol{y}^*) &= |\boldsymbol{x}|-2|\text{pairs}(\boldsymbol{y})\cap\text{pairs}(\boldsymbol{y}^*)| \\ &\quad -|\text{unpaired}(\boldsymbol{y})\cap\text{unpaired}(\boldsymbol{y}^*)| \end{aligned}$$

The ensemble defect is used to quantify the deviation of an RNA ensemble from a target structure $\boldsymbol{y}^*$, which is the expectation of structural distance to $\boldsymbol{y}^*$ over the Boltzmann ensemble:

$$\Phi(\boldsymbol{x},\boldsymbol{y}^*)=\mathbb{E}_{\boldsymbol{y}\sim p(\cdot\mid\boldsymbol{x})}\left[d(\boldsymbol{y},\boldsymbol{y}^*)\right]$$

On the surface, this definition seems to range over all possible structures in the expectation, but we can use dynamic programming to factor this computation to the expected number of incorrectly predicted nucleotides over the whole ensemble at equilibrium:

$$\Phi(\boldsymbol{x},\boldsymbol{y}^*)=|\boldsymbol{x}|-2\sum_{(i,j)\in\text{pairs}(\boldsymbol{y}^*)}p_{i,j}(\boldsymbol{x})-\sum_{j\in\text{unpaired}(\boldsymbol{y}^*)}q_j(\boldsymbol{x})$$

where $p_{i,j}(\boldsymbol{x})$ is the probability of nucleotide i pairing with nucleotide j, and $q_j(\boldsymbol{x})$ is the probability of nucleotide j being unpaired, defined as $q_j=1-\sum p_{i,j}$.

Table S1. RNAstralign benchmark dataset. These values are specifically calculated for the dataset of 10 samples (with $k=30$ homologs per sample), with sequence identity determined from the alignments performed by MAFFT `-auto`. This data is used for the evaluations shown in Fig. 2 and Fig. S1.

family	subfamily	avg. seq. len.	avg. seq. identity
Group 1	IC1	428.5	0.36
tmRNA	-	367.4	0.42
tRNA	-	77.1	0.48
5S rRNA	Bacteria	116.2	0.62
SRP	Protozoan	285.8	0.36
RNaseP	A bacterial	360.0	0.45
telomerase	-	444.9	0.50
16S rRNA	Alphaproteobacteria	1419.2	0.85

RNAstralign Datasets

We use a procedure similar to LinearTurboFold (9) to sample homologs from the RNAstralign dataset. Four families (Group I Intron, tmRNA, tRNA, and 5S rRNA) are used for tuning and another four families (SRP, RNaseP, telomerase, and 16S rRNA) are used for testing. For Group I Intron, 5S rRNA, SRP, RNaseP, and 16S rRNA, there are multiple subfamilies within each family, so we chose one specific subfamily for these five families (see Tab. S1). For 16S rRNA, we also made sure that only full-length sequences (rather than subdomains) are included. For each (sub)family, we drew 10 samples, each with $k=30$ homologs and align them by MAFFT `--auto`. Tab. S1 presents the average sequence length and average sequence identity of the sampled MSAs in each family, with sequence identity determined from the alignments performed by MAFFT `--auto`. This sampled data is used for the evaluations shown in Fig. 2 and Fig. S1. We included all these samples in our GitHub.

SARS-CoV-2 and SARS-related Datasets

We prepared a dataset to draw representative samples of diverse SARS-CoV-2 and SARS-related genomes. Based on the genomes from GISAID (39) (downloaded on 4 April 2022) and NCBI (`www.ncbi.nlm.nih.gov`; genomes submitted from 1998 to 2019), we first filtered out low-quality genomes (i.e., those with unknown characters or are shorter than 28,000nt). After preprocessing, we obtained two datasets with ~40,000 SARS-CoV-2 (including Alpha, Beta, Delta, and Omicron variants) and 600 SARS-related genomes, respectively. Following LinearTurboFold, we used a sampling algorithm to choose 60% diverse SARS-CoV-2 genomes and 40% diverse SARS-related genomes (see Tab. S2). Unlike LinearTurboFold(9), we did not use a greedy algorithm to choose the most diverse

Table S2. SARS-CoV-2 and SARS-related datasets. Ref is the SARS-CoV-2 reference sequence, Alpha–Delta are the SARS-CoV-2 variants, and SARSr are SARS-related genomes.

k	Ref	Alpha	Beta	Delta	Omicron	SARSr
10	1	2	2	1	1	3
30	1	4	5	4	4	12
50	1	7	8	7	7	20
100	1	14	15	15	15	40
200	1	33	33	33	20	80
300	1	59	60	40	20	120
400	1	79	80	60	20	160

genomes one by one, but only randomly sample for each category (Alpha, Beta, Delta, Omicron, SARS-related). We included all the COVID samples in our GitHub.

Hybrid Reference Structure Construction in the 5' and 3' UTR regions of SARS-CoV-2

To get the hybrid reference structure in the UTR regions (Fig. 4H), we combined the experimentally guided structures from Huston et al. (22) and the experimentally determined end-to-end pairs (Arch3, ranges from (60,29868) to (80,29847)) from Ziv et al. (18, Fig. 3) by the following steps:

1. Get (local) structures in 5' and 3' UTR regions from Huston et al. (the 5' UTR ranges from 1 to 400 and the 3' UTR from 29543 to 29876 on the reference sequence).
2. Remove (local) pairs (i,j) from the structures if i or j is in the global Arch3 pairs (e.g., SL3 from Huston et al. (22, Fig. 2) is removed). These local pairs were predicted by the local folding software which can only predict pairs within a local window.
3. Combine the modified structures and the end-to-end Arch3 pairs from Ziv et al.

See Fig. 4H for details; we also released its dot-bracket format on our Github. This hybrid structure is used for evaluating prediction qualities in Figs. 3 & S3.

Software and Computing Environment

We use the following software:

- RNAalifold (Vienna RNAfold 2.4.16) (`-r` mode) `https://www.tbi.univie.ac.at/RNA/`
- MAFFT 7.490 (always with `--auto` mode) `https://mafft.cbrc.jp/alignment/software/`
- LinAliFold (`-r 1`) `https://github.com/fukunagatsu/LinAliFold-CentroidLinAliFold`
- LinearTurboFold `https://github.com/LinearFold/LinearTurboFold`

We benchmarked these tools on a Linux machine with 2 Intel Xeon E5-2660 v3 CPUs (2.60 GHz) and 377 GB memory, and used `gcc` (Ubuntu 9.3.0-17) to compile.

CODE AND DATA AVAILABILITY

Our code and data are released on GitHub: `http://github.com/LinearFold/LinearAlifold`. Server at: `http://linearfold.org/linear-alifold`.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

L.H. conceived the idea and directed the project. L.Z. designed the main algorithm and implemented the initial version; A.M. reimplemented the whole system in much higher quality, implemented LazyOutside, ThreshKnot, and alifold-aware stochastic sampling, added BL* energy model, performed parameter tuning, and conducted thorough evaluations against LinAliFold, RNAalifold, and LinearTurboFold on SARS-CoV-2 and RNAstralign. A.M. also built the web server. M.G. made the visualizations of COVID structures and stochastic sampling. N.D. contributed to the evaluations on SARS-CoV-2. S.L. contributed to the evaluations on both RNAstralign and SARS-CoV-2, as well as the comparison to LinearTurboFold. H.Z. contributed to the algorithm design. D.H.M. guided the evaluations. L.H., L.Z., A.M., S.L., H.Z., and D.H.M. wrote the manuscript.

ACKNOWLEDGMENTS

This work was supported in part by National Institutes of Health Grant R35GM145283 (to D.H.M.) and National Science Foundation Grants 2009071 (to L.H.) and 2330737 (to L.H. and D.H.M.). We thank the reviewers for insightful suggestions which greatly improved the quality of our work. We also thank Tsukasa Fukunaga and Michiaki Hamad (authors of LinAlifold) for citing our preprint version.

REFERENCES

1. S. R. Eddy. Non-coding RNA genes and the modern rna world. *Nature Reviews Genetics*, 2(12):919–929, 2001.
2. Jennifer A. Doudna and Thomas R. Cech. The chemical repertoire of natural ribozymes. *Nature*, 418(6894):222–228, 2002.
3. Jean Pierre Bachellerie, Jérôme Cavaillé, and Alexander Hüttenhofer. The expanding snoRNA world. *Biochimie*, 84(8):775–790, 2002.
4. Eric P Nawrocki and Sean R Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29(22):2933–2935, 2013.
5. Edwin A Brown, Hangchun Zhang, Li-Hua Ping, and Stanley M Lemon. Secondary structure of the 5' nontranslated regions of hepatitis C virus and pestivirus genomic RNAs. *Nucleic Acids Research*, 20(19):5041–5045, 1992.
6. Justin Ritz, Joshua S Martin, and Alain Laederach. Evolutionary evidence for alternative structure in RNA sequence co-variation. *PLoS Computational Biology*, 9(7):e1003152, 2013.
7. Ivo L Hofacker, Martin Fekete, and Peter F Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319(5):1059–1066, 2002.
8. Stephan H Bernhart, Ivo L Hofacker, Sebastian Will, Andreas R Gruber, and Peter F Stadler. RNAalifold: improved consensus

structure prediction for RNA alignments. *BMC Bioinformatics*, 9(1):1–13, 2008.
9. Sizhen Li, He Zhang, Liang Zhang, Kaibo Liu, Boxiang Liu, David H Mathews, and Liang Huang. LinearTurboFold: Linear-time global prediction of conserved structures for RNA homologs with applications to SARS-CoV-2. *Proceedings of the National Academy of Sciences*, 118(52), 2021.
10. B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428, 2003.
11. Chuong Do, Daniel Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, 2006.
12. Zhi John Lu, Jason W Gloor, and David H Mathews. Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, 15(10):1805–1813, 2009.
13. Liang Zhang, He Zhang, David H. Mathews, and Liang Huang. ThreshKnot: Thresholded ProbKnot for Improved RNA Secondary Structure Prediction. *bioRxiv*, 2019.
14. Ye Ding and Charles E Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic acids research*, 31(24):7280–7301, 2003.
15. He Zhang, Liang Zhang, Sizhen Li, David H Mathews, and Liang Huang. LinearSampling: Linear-time stochastic sampling of RNA secondary structure with applications to SARS-CoV-2. *BioRxiv*, 2020.
16. Peter Clote, Yann Ponty, and Jean-Marc Steyaert. Expected distance between terminal nucleotides of RNA secondary structures. *Journal of mathematical biology*, 65(3):581–599, 2012.
17. Wan-Jung C Lai, Mohammad Kayedkhordeh, Erica V Cornell, Elie Farah, Stanislav Bellaousov, Robert Rietmeijer, David H Mathews, and Dmitri N Ermolenko. mRNAs and lncRNAs intrinsically form secondary structures with short end-to-end distances. *Nature Communications*, 9(1):4328, 2018.
18. Omer Ziv, Jonathan Price, Lyudmila Shalamova, Tsveta Kamenova, Ian Goodfellow, Friedemann Weber, and Eric A Miska. The short-and long-range RNA-RNA interactome of SARS-CoV-2. *Molecular Cell*, 80(6):1067–1077, 2020.
19. Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics*, 35(14):i295–i304, 07 2019.
20. He Zhang, Liang Zhang, David H Mathews, and Liang Huang. LinearPartition: linear-time approximation of RNA folding partition function and base-pairing probabilities. *Bioinformatics*, 36(Supplement_1):i258–i267, 2020.
21. Zhen Tan, Yinghan Fu, Gaurav Sharma, and David H Mathews. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research*, 45(20):11570–11581, 2017.
22. Nicholas C Huston, Han Wan, Madison S Strine, Rafael de Cesaris Araujo Tavares, Craig B Wilen, and Anna Marie Pyle. Comprehensive in vivo secondary structure of the SARS-CoV-2 genome reveals novel regulatory motifs and mechanisms. *Molecular Cell*, 81(3):584–598, 2021.
23. Tsukasa Fukunaga and Michiaki Hamada. Linalifold and centroidlinalifold: Fast rna consensus secondary structure prediction for aligned sequences using beam search methods. *Bioinformatics Advances*, 2(1):vbac078, 2022.
24. Michiaki Hamada, Hisanori Kiryu, Kengo Sato, Toutai Mituyama, and Kiyoshi Asai. Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25(4):465–473, 2009.
25. Liang Huang, Otso Barron, Apoorv Malik, Sizhen Li, and David H. Mathews. Lazy outside and lazy backward algorithms. in preparation, 2024.
26. Mirela Andronescu, Anne Condon, Holger H Hoos, David H Mathews, and Kevin P Murphy. Computational approaches for rna energy parameter estimation. *RNA*, 16(12):2304–2318, 2010.
27. Liang Huang and David Chiang. Better k-best parsing. *Proceedings of the Ninth International Workshop on Parsing Technologies*, pages 53–64, 2005.
28. Jamie J Cannone, Sankar Subramanian, Murray N Schnare, James R Collett, Lisa M D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V Madabusi, Kirsten M Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R Gutell. The Comparative RNA Web (CRW) Site: An Online Database of Comparative Sequence and Structure Information for Ribosomal, Intron, and Other RNAs. *BioMed Central Bioinformatics*, 3(2), 2002.
29. Kazutaka Katoh and Daron M Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780, 2013.
30. Nima Aghaeepour and Holger H Hoos. Ensemble-based prediction of RNA secondary structures. *BMC Bioinformatics*, 14(139).
31. He Zhang, Sizhen Li, Liang Zhang, David H Mathews, and Liang Huang. Lazysampling and linearsampling: fast stochastic sampling of rna secondary structure with applications to sars-cov-2. *Nucleic acids research*, 51(2):e7–e7, 2022.
32. He Zhang, Sizhen Li, Ning Dai, Liang Zhang, David H Mathews, and Liang Huang. LinearCoFold and LinearCoPartition: linear-time algorithms for secondary structure prediction of interacting RNA molecules. *Nucleic Acids Research*, 51(18):e94–e94, 2023.
33. Liang Huang and Kenji Sagae. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL 2010*, page 1077–1086, Uppsala, Sweden, 2010. ACL.
34. Lei Sun, Pan Li, Xiaohui Ju, Jian Rao, Wenze Huang, Lili Ren, Shaojun Zhang, Tuanlin Xiong, Kui Xu, Xiaolin Zhou, et al. In vivo structural characterization of the SARS-CoV-2 RNA genome identifies host proteins vulnerable to repurposed drugs. *Cell*, 184(7):1865–1883, 2021.
35. James K Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132, 1979.
36. J. S. McCaskill. The equilibrium partition function and base pair probabilities for rna secondary structure. *Biopolymers*, 29:11105–1119, 1990.
37. Liang Huang. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594, 2008.
38. Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2010.
39. Stefan Elbe and Gemma Buckland-Merrett. Data, disease and diplomacy: GISAID's innovative contribution to global health. *Global challenges*, 1(1):33–46, 2017.

LinearAlifold: Linear-Time Consensus Structure Prediction for RNA Alignments

Apoorv Malik, Liang Zhang, Milan Gautam, Ning Dai, Sizhen Li, He Zhang, David H. Mathews, Liang Huang

Supplementary Figures

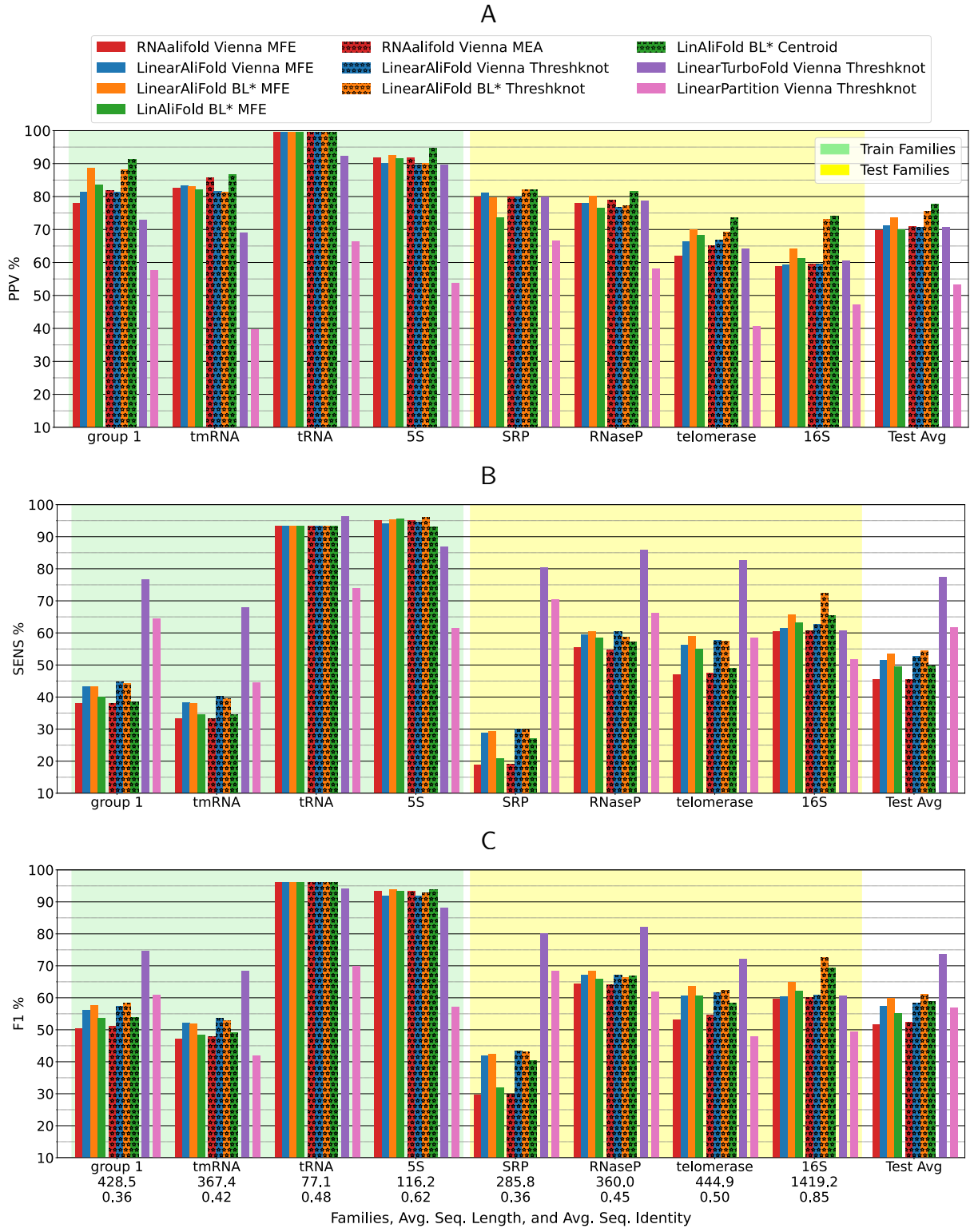


Figure S1. Accuracy comparisons on the RNAstralign dataset, similar to Fig. 2 but including more systems. Each family has 10 samples, and each sample is an MSA with $k=30$ homologs. Align-then-fold systems (RNAalifold, LinearAlifold, and LinAliFold) tend to be inaccurate for low sequence indentity families (e.g., SRP and group 1) and tend to be more accurate for high sequence identity families (e.g., 16S rRNA). Refer to Fig. S2 for a similar figure with 20 samples per family.

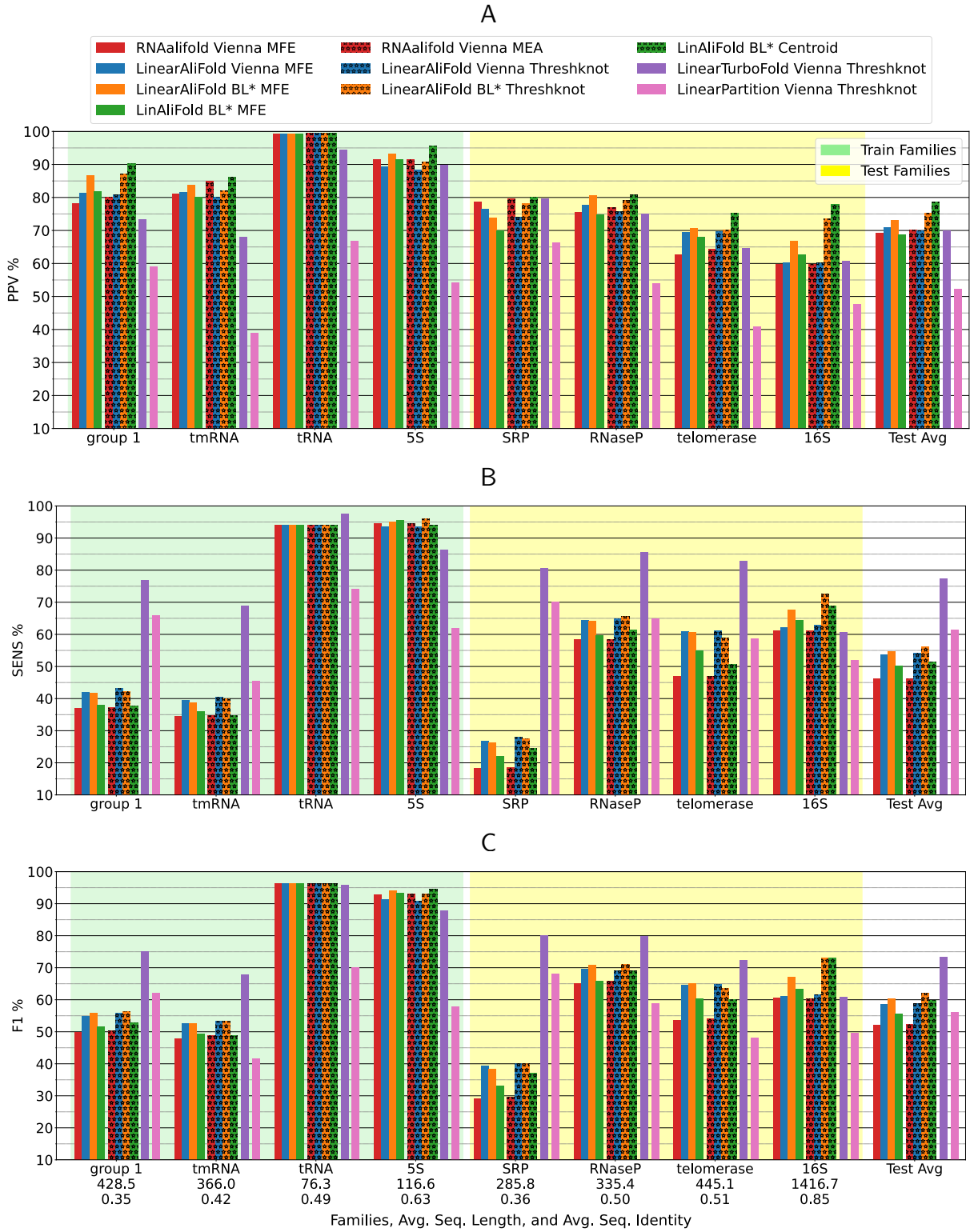


Figure S2. Accuracy comparisons on the RNAstralign dataset. Each family has 20 samples, and each sample is an MSA with $k=30$ homologs. Align-then-fold systems (RNAalifold, LinearAlifold, and LinAliFold) tend to be inaccurate for low sequence indentity families (e.g., SRP and group 1) and tend to be more accurate for high sequence identity families (e.g., 16S rRNA). Refer to Fig. S1 for a similar figure with 10 samples per family.

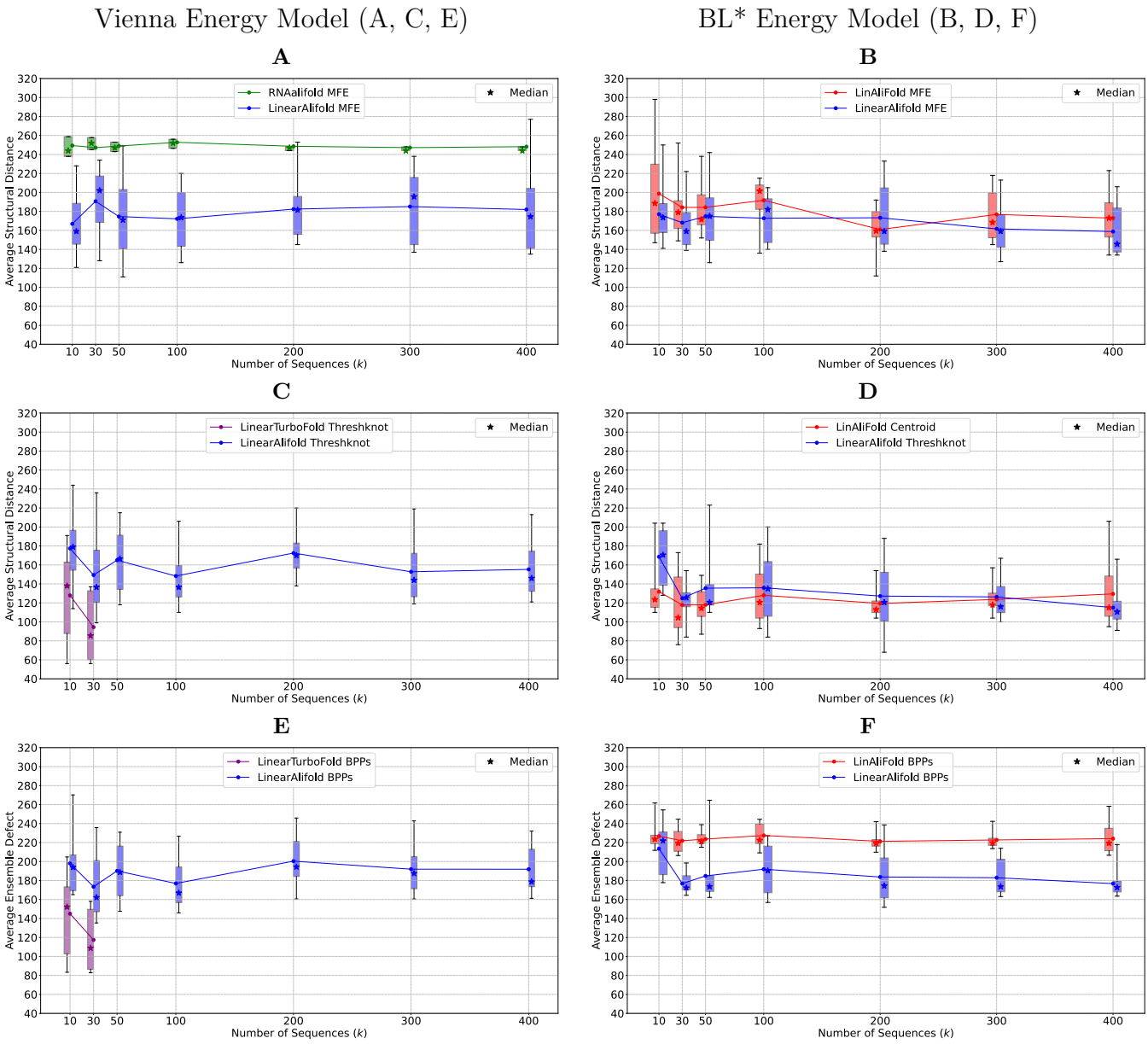


Figure S3. Box plot of structural distance and ensemble defect (of 5' and 3' UTRs) against the number of sequences (k) for different energy models and methods, with the structural distance evaluated using the hybrid COVID structure (Fig. 4H; Huston et al. + Ziv et al.) as the reference (see Fig. S4 for a similar figure with the Huston et al. structure as reference). For each k, we have 10 samples, so the boxes show the 25-75 percentiles, and the whiskers represent the full range of data (1-100 percentile). The curves show the mean values over 10 samples for each k, and the stars denote the medians. See Fig. 3 for another version where the x-axes are time instead of k. A–B: MFE prediction. C–D: partition-based structure prediction. E–F: ensemble quality.

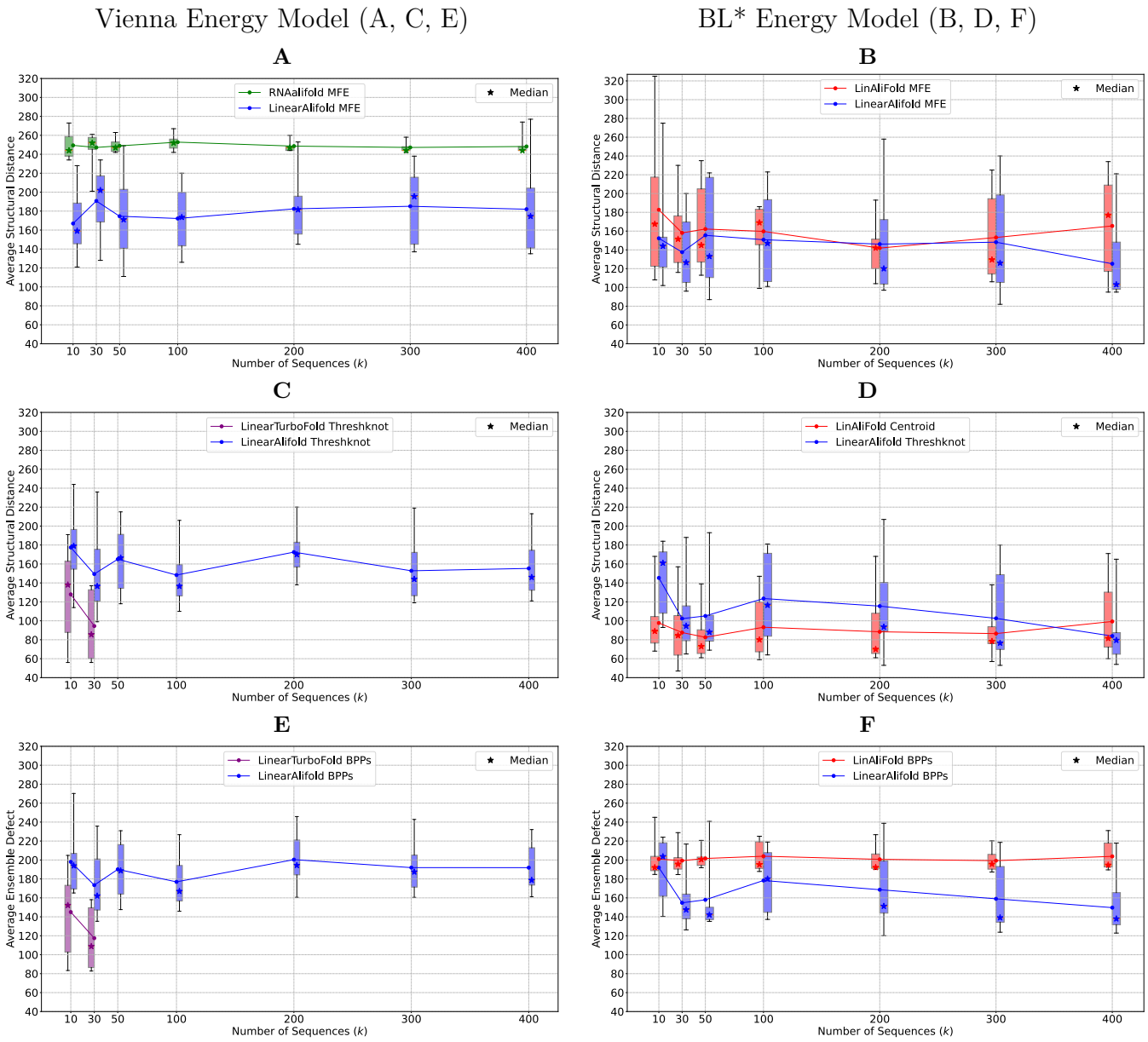
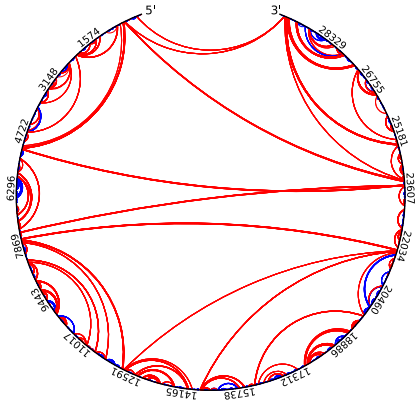


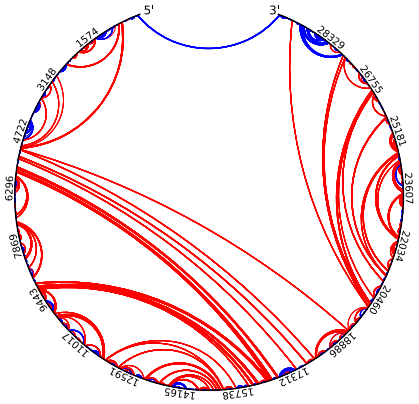
Figure S4. Box plot of structural distance and ensemble defect (of 5' and 3' UTRs) against the number of sequences (k) for different energy models and methods, with the structural distance evaluated using the Huston et al. reference structure as the reference (see Fig. S3 for a similar figure with the hybrid structure as reference). For each k, we have 10 samples, so the boxes show the 25-75 percentiles, and the whiskers represent the full range of data (1-100 percentile). The curves show the mean values over 10 samples for each k, and the stars denote the medians. A–B: MFE prediction. C–D: partition-based structure prediction. E–F: ensemble quality.

A: LinearAlifold Vienna Threshknot



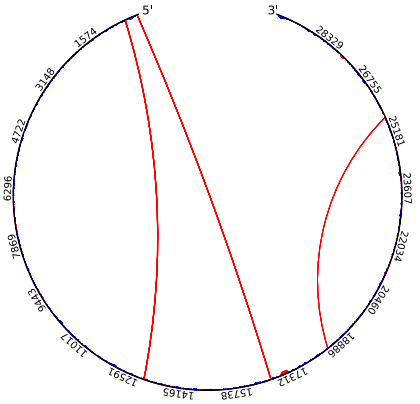
Pairing Distance	Predicted	Matched	Precision
[0, +∞)	9728	7409	76.2%
[0, 100)	7833	6801	86.8%
[100, +∞)	1903	615	32.3%
[29700, +∞)	0	0	0.0%

B: LinearAlifold BL* Threshknot



Pairing Distance	Predicted	Matched	Precision
[0, +∞)	8856	6531	73.7%
[0, 100)	6634	5727	86.3%
[100, +∞)	2229	811	36.4%
[29700, +∞)	14	14	100.0%

C: LinearTurboFold Vienna Threshknot



Pairing Distance	Predicted	Matched	Precision
[0, +∞)	8141	7022	86.3%
[0, 100)	7955	6888	86.6%
[100, +∞)	197	145	73.6%
[29700, +∞)	0	0	0.0%

Figure S5. COVID circular plots with Ziv et al. range precisions for various methods. Evaluated on the COVID $k=30$ sample #5/10. Red arcs do not match any Ziv et al. range, while blue arcs match at least one Ziv et al. range. See also Fig. 4A–C.