LinearTurboFold: Linear-Time Global Prediction of Conserved Structures for RNA Homologs with Applications to SARS-CoV-2

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

[a]School of Electrical Engineering & Computer Science, Oregon State University, Corvallis, OR; [b]Baidu Research, Sunnyvale, CA; [d]Department of Biochemistry & Biophysics, Center for RNA Biology, and Department of Biostatistics & Computational Biology, University of Rochester Medical Center, Rochester, NY

The constant emergence of COVID-19 variants reduces the effectiveness of existing vaccines and test kits. Therefore, it is critical to identify conserved structures in SARS-CoV-2 genomes as potential targets for variant-proof diagnostics and therapeutics. However, the algorithms to predict these conserved structures, which simultaneously fold and align multiple RNA homologs, scale at best cubically with sequence length, and are thus infeasible for coronaviruses, which possess the longest genomes ($\sim$30,000 *nt*) among RNA viruses. As a result, existing efforts on modeling SARS-CoV-2 structures resort to single sequence folding as well as local folding methods with short window sizes, which inevitably neglect long-range interactions that are crucial in RNA functions. Here we present LinearTurboFold, an efficient algorithm for folding RNA homologs that scales *linearly* with sequence length, enabling unprecedented *global* structural analysis on SARS-CoV-2. Surprisingly, on a group of SARS-CoV-2 and SARS-related genomes, LinearTurboFold's purely *in silico* prediction not only is close to experimentally-guided models for local structures, but also goes far beyond them by capturing the end-to-end pairs between 5' and 3' UTRs ($\sim$29,800 *nt* apart) that match perfectly with a purely experimental work. Furthermore, LinearTurboFold identifies novel conserved structures and conserved accessible regions as potential targets for designing efficient and mutation-insensitive small-molecule drugs, antisense oligonucleotides, siRNAs, CRISPR-Cas13 guide RNAs and RT-PCR primers. LinearTurboFold is a general technique that can also be applied to other RNA viruses and full-length genome studies, and will be a useful tool in fighting the current and future pandemics.

RNA secondary structure | homologous folding | conserved structures | structural alignment | SARS-CoV-2

Ribonucleic acid (RNA) plays important roles in many cellular processes (1, 2). To maintain their functions, secondary structures of RNA homologs are conserved across evolution (3–5). These conserved structures provide critical targets for diagnostics and treatments. Thus, there is a need for developing fast and accurate computational methods to identify structurally conserved regions.

Commonly, conserved structures involve compensatory base pair changes, where two positions in primary sequences mutate across evolution and still conserve a base pair, for instance, an AU or a CG pair replaces a GC pair in homologous sequences. These compensatory changes provide strong evidence for evolutionarily conserved structures (6–10). Meanwhile, they make it harder to align sequences when structures are unknown. Initially, the process of determining a conserved structure, termed comparative sequence analysis, was manual and required substantial insight to identify the conserved structure. A notable early achievement was the determination of the conserved tRNA secondary structure (11). Comparative analysis was also demonstrated to be 97% accurate as compared to crystal structures for ribosomal RNAs, where the models were refined carefully over time (12).

To automate comparative analysis, three distinct algorithmic approaches were developed (13, 14). The first, "joint fold-and-align" method, seeks to simultaneously predict structures and a structural alignment for two or more sequences. This was first proposed by Sankoff (15) using a dynamic programming algorithm. The major limitation of this approach is that the algorithm runs in $O(n^{3k})$ against k sequences with the average sequence length n. Several software packages provide implementations of the Sankoff algorithm (16–21) that use simplifications to reduce runtime. The second, "align-then-fold" approach, is to input a sequence alignment and predict the conserved structure that can be identified across sequences in the alignment. This was described by Waterman (22), and was subsequently refined and popularized by RNAalifold (23). The third, "fold-then-align" approach, is to predict plausible structures for the sequences, and then align the structures to determine the sequence alignment and the optimal conserved structures. This was described by Waterman (24) and implemented in RNAforester (25) and MARNA (26) (*SI Appendix*, Fig. S1).

As an alternative, TurboFold II (27), an extension of TurboFold (28), provides a more computationally efficient method to align and fold sequences. Taking multiple unaligned sequences as input, TurboFold II iteratively refines alignments and structure predictions so that they conform more closely to each other and converge on conserved structures. TurboFold II is significantly more accurate than

Significance Statement

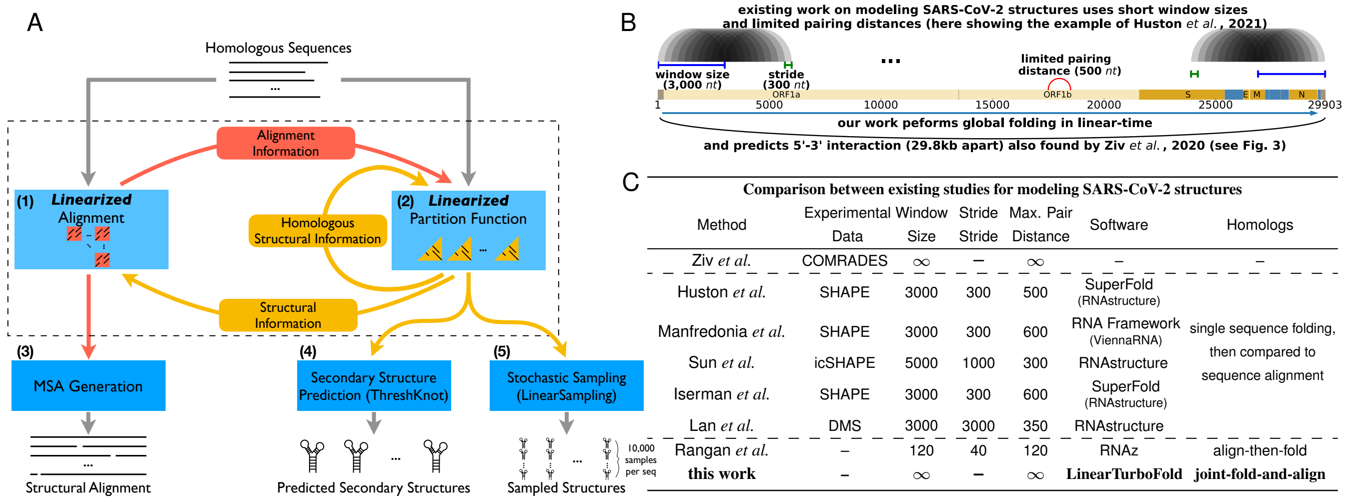
Conserved RNA structures are critical for designing diagnostic and therapeutic tools for many diseases including COVID-19. However, existing algorithms are much too slow to model the global structures of full-length RNA viral genomes. We present LinearTurboFold, a linear-time algorithm that is orders of magnitude faster, making it the first method to simultaneously fold and align whole genomes of SARS-CoV-2 variants, the longest known RNA virus ($\sim$30 kilobases). Our work enables unprecedented *global* structural analysis and captures long-range interactions that are out of reach for existing algorithms but crucial for RNA functions. LinearTurboFold is a general technique for full-length genome studies and can help fight the current and future pandemics.

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. L.K. made the webserver. H.Z., B.L., L.H., and D.H.M. guided the analysis on SARS-CoV-2.

The authors declare no conflict of interest.

* Corresponding authors: liang.huang.sh@gmail.com or David_Mathews@urmc.rochester.edu.

† Lead contact.


A
Homologous Sequences
...
Alignment Information
(1) Linearized Alignment
Homologous Structural Information
(2) Linearized Partition Function
...
Structural Information
(3)
MSA Generation
(4)
Secondary Structure Prediction (ThreshKnot)
(5)
Stochastic Sampling (LinearSampling)
...
Structural Alignment
...
Predicted Secondary Structures
...
10,000 samples per seq
Sampled Structures
B
existing work on modeling SARS-CoV-2 structures uses short window sizes and limited pairing distances (here showing the example of Huston et al. , 2021)
window size (3,000 nt)
stride (300 nt)
...
limited pairing distance (500 nt)
ORF1a
ORF1b
S
E
M
N
1
5000
10000
15000
20000
25000
29903
our work peforms global folding in linear-time
and predicts 5'-3' interaction (29.8kb apart) also found by Ziv et al. , 2020 (see Fig. 3)
C

Comparison between existing studies for modeling SARS-CoV-2 structures

Method	Experimental	Window	Stride	Max. Pair	Software	Homologs
	Data	Size	Stride	Distance		
Ziv *et al.*	COMRADES	∞	–	∞	–	–
Huston *et al.*	SHAPE	3000	300	500	SuperFold (RNAstructure)	single sequence folding, then compared to sequence alignment
Manfredonia *et al.*	SHAPE	3000	300	600	RNA Framework (ViennaRNA)	
Sun *et al.*	icSHAPE	5000	1000	300	RNAstructure	
Iserman *et al.*	SHAPE	3000	300	600	SuperFold (RNAstructure)	
Lan *et al.*	DMS	3000	3000	350	RNAstructure	
Rangan *et al.*	–	120	40	120	RNAz	align-then-fold
this work	–	∞	–	∞	**LinearTurboFold**	**joint-fold-and-align**

Fig. 1. **A**: The LinearTurboFold framework. Like TurboFold II, LinearTurboFold takes multiple unaligned homologous sequences as input and outputs a secondary structures for each sequence, and a multiple sequence alignment (MSA). But unlike TurboFold II, LinearTurboFold employs two linearizations to ensure linear runtime: a *linearized* alignment computation (module **1**) to predict posterior co-incidence probabilities (red squares) for all pairs of sequences (see **Methods §1–4**), and a *linearized* partition function computation (module **2**) to estimate base-pairing probabilities (yellow triangles) for all the sequences (see **Methods §5–6**). These two modules take advantage of information from each other and iteratively refine predictions (*SI Appendix*, Fig. S2). After several iterations, module **3** generates the final multiple sequence alignments (see **Methods §7**), and module **4** predicts secondary structures. Module **5** can stochastically sample structures. **B–C**: Prior studies (31–36) (except for the purely experimental work by Ziv *et al.* (37)) used local folding methods with limited window size and maximum pairing distance. **B** shows the local folding of the SARS-CoV-2 genome by Huston *et al.*, which used a window of 3,000 *nt* that was advanced 300 *nt*. It also limited the distance between nucleotides that can base pair at 500. Some work also used homologous sequences to identify conserved structures, but they only predicted structures for one genome and utilized sequence alignments to identify mutations. By contrast, LinearTurboFold is a global folding method without any limitations on sequence length or paring distance, and it jointly folds and aligns homologs to obtain conserved structures. Consequently, LinearTurboFold can capture long-range interactions even across the whole genome (the long arc in **B** and Fig. 3).

other methods (16, 18, 23, 29, 30) when tested on RNA families with known structures and alignments.

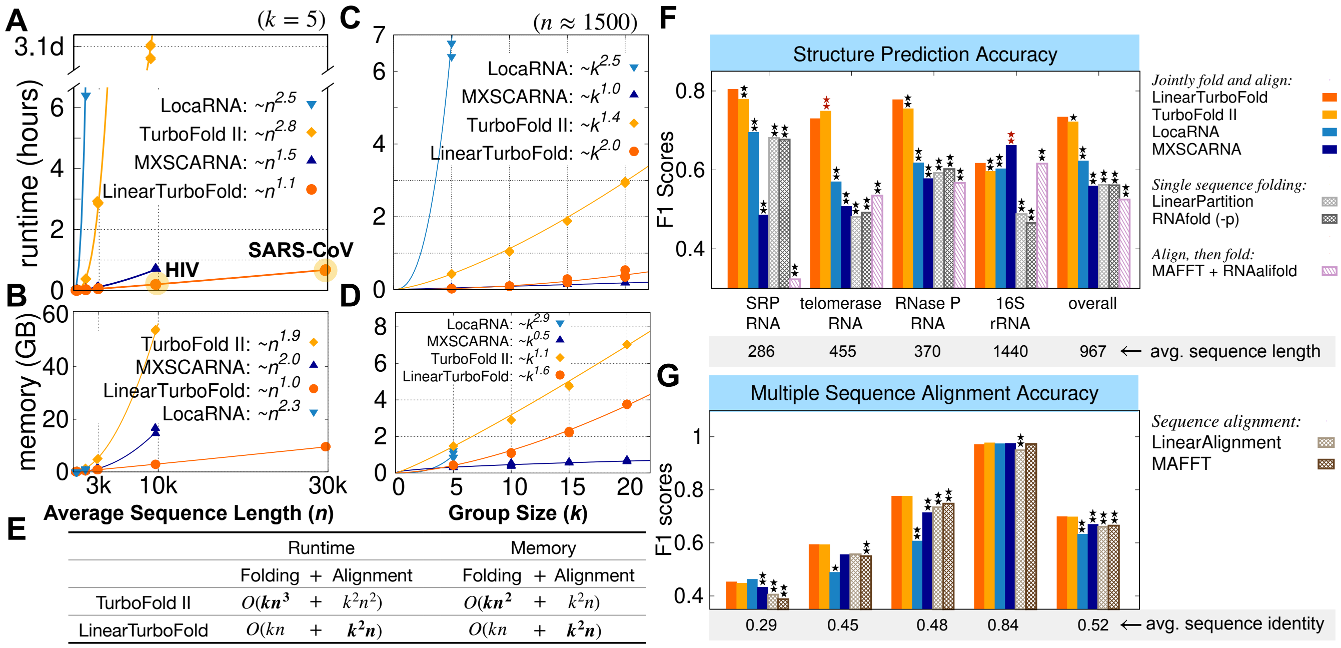
However, the cubic runtime and quadratic memory usage of TurboFold II prevent it from scaling to longer sequences such as full-length SARS-CoV-2 genomes, which contain ∼30,000 nucleotides; in fact, no joint-align-and-fold methods can scale to these genomes, which are the longest among RNA viruses. As a (not very principled) workaround, most existing efforts for modeling SARS-CoV-2 structures (31–36) resort to local folding methods (38, 39) with sliding windows plus a limited pairing distance, abandoning all long-range interactions, and only consider one SARS-CoV-2 genome (Fig. 1B–C), ignoring signals available in multiple homologous sequences. To address this challenge, we designed a linearized version of TurboFold II, *LinearTurboFold* (Fig. 1A), which is a global homologous folding algorithm that scales linearly with sequence length. This linear runtime makes it the first joint-fold-and-align algorithm to scale to full-length coronavirus genomes without any constraints on window size or pairing distance, taking about 13 hours to analyze a group of 25 SARS-CoV homologs. It also leads to significant improvement on secondary structure prediction accuracy as well as an alignment accuracy comparable to or higher than all benchmarks.

Over a group of 25 SARS-CoV-2 and SARS-related homologous genomes, LinearTurboFold predictions are close to the canonical structures (40) and structures modeled with the aid of experimental data (32–34) for several well-studied regions. Thanks to global rather than local folding, LinearTurboFold discovers a long-range interaction involving 5' and 3' UTRs (∼29,800 *nt* apart), which is consistent with recent purely experimental work (35), and yet is out of reach for local folding methods used by existing studies (Fig. 1B–C). In short, our *in silico* method of folding multiple homologs can achieve results similar to, and sometimes more accurate than, experimentally-guided models for one genome. Moreover, LinearTurboFold identifies conserved structures supported by compensatory mutations, which are potential targets for small molecule drugs (41) and antisense oligonucleotides (ASOs) (36). We further identify regions that are (a) sequence-level conserved, (b) at least 15 *nt* long, and (c) accessible (i.e., likely to be completely unpaired) as potential targets for ASOs (42), small interfering RNA (siRNA) (43), CRISPR-Cas13 guide RNA (gRNA) (44) and reverse transcription polymerase chain reaction (RT-PCR) primers (45). LinearTurboFold is a general technique that can also be applied to other RNA viruses (e.g., influenza, Ebola, HIV, Zika, etc.) and full-length genome studies.

Results

The framework of LinearTurboFold has two major aspects (Fig. 1A): linearized structure-aware pairwise alignment estimation (module **1**); and linearized homolog-aware structure prediction (module **2**). LinearTurboFold iteratively refines alignments and structure predictions, specifically, updating pairwise alignment probabilities by incorporating predicted base-pairing probabilities (from module **2**) to form structural alignments, and modifying base-pairing probabilities for each sequence by integrating the structural information from homologous sequences via the estimated alignment probabilities (from module **1**) to detect conserved structures. After several iterations, LinearTurboFold generates the final multiple sequence alignment (MSA) based on the latest pairwise alignment probabilities (module **3**) and predicts secondary structures using the latest pairing probabilities (module **4**).

LinearTurboFold achieves linear time regarding sequence length with two major linearized modules: our recent work LinearPartition (46) (Fig. 1A module **2**), which approximates the RNA partition function (47) and base pairing probabilities in linear time, and a novel algorithm LinearAlignment (module **1**). LinearAlignment aligns two sequences by Hidden Markov Model (HMM) in linear time by applying the same beam search heuristic (48) used by LinearPartition.



	Runtime			Memory		
	Folding	+	Alignment	Folding	+	Alignment
TurboFold II	$O(\boldsymbol{kn^3}$	+	$k^2n^2)$	$O(\boldsymbol{kn^2}$	+	$k^2n)$
LinearTurboFold	$O(kn$	+	$\boldsymbol{k^2n})$	$O(kn$	+	$\boldsymbol{k^2n})$

Fig. 2. End-to-end Scalability and Accuracy Comparisons. **A–B**: End-to-end runtime and memory usage comparisons between benchmarks and LinearTurboFold against the sequence length. LinearTurboFold uses beam size 100 in both partition function and HMM alignment calculation with three iterations to run all groups of data. **C–D**: End-to-end runtime and memory usage comparisons against the group size. LinearTurboFold is the first joint-fold-and-align algorithm to scale to full-length coronavirus genomes (∼30,000 *nt*) due to its linear runtime. **E**: The runtime and space complexity comparisons between TurboFold II and LinearTurboFold. The dominating terms are in bold. **F–G**: The F1 accuracy scores of the structure prediction and multiple sequence alignment (*SI Appendix*, Tab. S1). LocARNA and MXSCARNA are Sankoff-style simultaneous folding and alignment algorithms for homologous sequences. As negative controls, LinearPartition and Vienna RNAfold-predicted structures for each sequence separately; LinearAlignment and MAFFT generated sequence-level alignments; RNAalifold folded pre-aligned sequences (e.g., from MAFFT) and predicted conserved structures. Statistical significances (two-tailed permutation test) between the benchmarks and LinearTurboFold are marked with one star (⋆) on the top of the corresponding bars if $p < 0.05$ or two stars ($\overset{\star}{\star}$) if $p < 0.01$. The benchmarks whose accuracies are significantly lower than LinearTurboFold are annotated with black stars, while benchmarks higher than LinearTurboFold are marked with dark red stars. Overall, on structure prediction, LinearTurboFold achieves significantly higher accuracy than all evaluated benchmarks, and on multiple sequence alignment, it achieves accuracies comparable to TurboFold II and significantly higher than other methods (*SI Appendix*, Tab. S1).

Finally, LinearTurboFold assembles the secondary structure from the final base pairing probabilities using an accurate and linear-time method named ThreshKnot (49) (module **4**).

LinearTurboFold also integrates a linear-time stochastic sampling algorithm named LinearSampling (50) (module **5**), which independently samples structures according to the homolog-aware partition functions and then calculates the probability of being unpaired for regions, which is an important property in, for example, siRNA sequence design (43). Therefore, the overall end-to-end runtime of LinearTurboFold scales linearly with sequence length (**Methods §1–7**). The number of iterations and other hyperparameters were tuned on the training set. As observed previously (27, 28), improvements after three iterations are negligible, therefore the best number of iterations is set to be three. On the testing set, it is observed that LinearTurboFold achieves the most substantial improvements in both structure prediction and MSA accuracy in the first iteration and continues to benefit from the next two iterations (*SI Appendix*, Fig. S5). which is consistent with the observation on the training set. After approximately three iterations, both structure prediction and MSA accuracies remain stable with small fluctuations. To better demonstrate the improvement in each iteration, we visualized both base-pairing probabilities and alignment co-incidence probabilities from LinearTurboFold for a group of five tRNAs across iterations (*SI Appendix*, Fig. S6–S7).

Scalability and Accuracy. To evaluate the efficiency of LinearTurboFold against the sequence length, we collected a dataset consisting of seven families of RNAs with sequence length ranging from 210 *nt* to 30,000 *nt*, including five families from the RNAStrAlign dataset (27) plus 23S ribosomal RNA, HIV genomes and SARS-CoV genomes, and the calculation for each family uses five homologous sequences (**Methods §8**). Fig. 2A compares the running times of LinearTurboFold with TurboFold II and two Sankoff-style simultaneous folding and alignment algorithms, LocARNA and MXSCARNA. Clearly, LinearTurboFold scales linearly with sequence length n, and is substantially faster than other algorithms, which scale superlinearly. The linearization in LinearTurboFold brought orders of magnitude speedup over the cubic-time TurboFold II, taking only 12 minutes on the HIV family (average length 9,686 *nt*) while TurboFold II takes 3.1 days (372× speedup). More importantly, LinearTurboFold takes only 40 minutes on five SARS-CoV sequences while all other benchmarks fail to scale. Regarding the memory usage (Fig. 2B), LinearTurboFold costs linear memory space with sequence length, while other benchmarks use quadratic or more memory. In Fig. 2C–D, we also demonstrate that the runtime and memory usage against the number of homologs using sets of 16S rRNAs about 1,500 *nt* in length. The apparent complexity of LinearTurboFold against the group size k is higher than that of TurboFold II because the runtime of the latter is $O(kn^3 + k^2n^2)$ and is dominated by the $O(kn^3)$ partition function calculation, thus scaling $O(k^{1.4})$ empirically. By contrast, LinearTurboFold linearizes both partition function and alignment modules, so its overall runtime becomes $O(kn + k^2n)$ and is instead dominated by the $O(k^2n)$ alignment module, therefore scaling $O(k^2)$ in practice. A similar analysis holds for memory usage (Fig. 2E).[*]

[*] Theoretically, the alignment part takes $O(k^2n^2)$ space. However, in practice, TurboFold II discards positions whose alignment co-incidence probabilities less than thresholds and only keeps a linear number of positions. (51)

We next compare the accuracies of secondary structure prediction and MSA between LinearTurboFold and several benchmark methods (**Methods §9**). Besides Sankoff-style LocARNA and MXSCARNA, we also consider three types of negative controls: (a) single sequence folding (partition function-based): Vienna RNAfold (39) (-p mode) and LinearPartition; (b) sequence-only alignment: MAFFT (29) and LinearAlignment (a standalone version of the alignment method developed for this work but without structural information); and (c) an align-then-fold method that predicts consensus structures from MSAs (*SI Appendix*, Fig. S1): MAFFT + RNAalifold (23).

For secondary structure prediction, LinearTurboFold, TurboFold II and LocARNA achieve higher F1 scores than single sequence folding methods (Vienna RNAfold and LinearPartition) (Fig. 2F), which demonstrates folding with homology information performs better than folding sequences separately. Overall, LinearTurboFold performs significantly better than all the other benchmarks on structure prediction. For the accuracy of MSAs (Fig. 2G), the structural alignments from LinearTurboFold obtain higher accuracies than sequence-only alignments (LinearAlignment and MAFFT) on all four families, especially for families with low sequence identity. On average, LinearTurboFold performs comparably with TurboFold II and significantly better than other benchmarks on alignments. We also note that the structure prediction accuracy of the align-then-fold approach (MAFFT + RNAalifold) depends heavily on the alignment accuracy, and is the worst when the sequence identity is low (e.g., SRP RNA) and the best when the sequence identity is high (e.g., 16S rRNA) (Fig. 2F–G).

Highly Conserved Structures in SARS-CoV-2 and SARS-related Betacoronaviruses. RNA sequences with conserved secondary structures play vital biological roles and provide potential targets. The current COVID-19 outbreak raises an emergent requirement of identifying potential targets for diagnostics and therapeutics. Given the strong scalability and high accuracy, we used LinearTurboFold on a group of full-length SARS-CoV-2 and SARS-related (SARSr) genomes to obtain global structures and identify highly conserved structural regions.

We used a greedy algorithm to select the 16 most diverse genomes from all the valid SARS-CoV-2 genomes submitted to the Global Initiative on Sharing Avian Influenza Data (GISAID) (52) up to December 2020 (**Methods §11**). We further extended the group by adding 9 SARS-related homologous genomes (5 human SARS-CoV-1 and 4 bat coronaviruses) (53). In total, we built a dataset of 25 full-length genomes consisting of 16 SARS-CoV-2 and 9 SARS-related sequences (*SI Appendix*, Fig. S9).The average pairwise sequence identities of the 16 SARS-CoV-2 and the total 25 genomes are 99.9% and 89.6%, respectively. LinearTurboFold takes about 13 hours and 43 GB on the 25 genomes.

To evaluate the reliability of LinearTurboFold predictions, we first compare them with the Huston *et al.*'s SHAPE-guided models (32) for regions with well-characterized structures across betacoronaviruses. For the extended 5' and 3' untranslated regions (UTRs), LinearTurboFold's predictions are close to the SHAPE-guided structures (Fig. 3A–B), i.e., both identify the stem-loops (SLs) 1–2 and 4–7 in the extended 5' UTR, and the bulged stem-loop (BSL), SL1, and a long bulge stem for the hypervariable region (HVR) including the stem-loop II-like motif (S2M) in the 3' UTR. Interestingly, in our model, the high unpaired probability of the stem in the SL4b indicates the possibility of being single-stranded as an alternative structure, which is supported by experimental studies (33, 36). In addition, the compensatory mutations LinearTurboFold found in UTRs strongly support the evolutionary conservation of structures (Fig. 3A).

The most important difference between LinearTurboFold's prediction and Huston *et al.*'s experimentally-guided model is that LinearTurboFold discovers an end-to-end interaction (29.8 kilobases apart) between the 5' UTR (SL3, 60-82 *nt*) and the 3' UTR (final region, 29845-29868 *nt*), which fold locally by themselves in Huston *et al.*'s model. Interestingly, this 5'-3' interaction matches *exactly* with the one discovered by the purely experimental work of Ziv *et al.* (37) using the COMRADES technique to capture long-range base-pairing interactions (Fig. 3C). These end-to-end interactions have been well established by theoretical and experimental studies (54–56) to be common in natural RNAs, but are far beyond the reaches of local folding methods used in existing studies on SARS-CoV-2 secondary structures (32–35). By contrast, LinearTurboFold predicts secondary structures globally without any limit on window size or base-pairing distance, enabling it to discover long-distance interactions across the whole genome. The similarity between our predictions and the experimental work shows that our *in silico* method of folding multiple homologs can achieve results similar to, if not more accurate than, those experimentally-guided single-genome prediction.

LinearTurboFold can model these end-to-end interactions thanks to three ingredients: (a) linearization, (b) LinearPartition's better modeling power on long sequences and long-range pairs, and (c) homologous folding and soft alignment. Linearization not only enables LinearTurboFold to scale to longer sequences, but also improves the accuracy of modeling long-range interactions benefiting from LinearPartition (46). In addition, homologous folding is also crucial. We observed that LinearPartition can model the same end-to-end interactions detected by Ziv *et al.* for 8 out of 25 sequences (4 out of 16 SARS-CoV-2 and 4 out of 9 SARS-related sequences; see *SI Appendix*, Fig. S12A and the left column of Fig. S13). For the other sequences, however, LinearPartition either cannot predict end-to-end interactions or predicts them in the wrong locations. On the other hand, LinearTurboFold propagates the correct structural information from those eight sequences to other homologs, resulting in all SARS-CoV-2 sequences having the same end-to-end pairs (*SI Appendix*, Fig. S12B and the right column of Fig. S13). By contrast, the align-then-fold approach (MAFFT + RNAalifold), which relies on the input hard alignment and predicts one single consensus structure for all homologs, fails to predict such long-range interactions (*SI Appendix*, Fig. S10B).

The frameshifiting stimulation element (FSE) is another well-characterized region. For an extended FSE region, the LinearTurboFold prediction consists of two substructures (Fig. 4A): the 5' part includes an attenuator hairpin and a stem, which are connected by a long internal loop (16 *nt*) including the slippery site, and the 3' part includes three stem loops. We observe that our predicted structure of the 5' part is consistent with experimentally-guided models (32, 33, 35) (Fig. 4B–D). In the attenuator hairpin, the small internal loop motif (UU) was previously selected as a small molecule binder that stabilizes the folded state of the attenuator hairpin and impairs frameshifting (41). For the long internal loop including the slippery site, we will show in the next section that it is both highly accessible and conserved (Fig. 5), which makes it a perfect candidate for drug design. For the 3' region of the FSE, LinearTurboFold successfully predicts stems 1–2 (but misses stem 3) of the canonical three-stem pseudoknot (40) (Fig. 4E). Our prediction is closer to the canonical structure compared to the experimentally-guided models (32, 33, 35) (Fig. 4B–D); one such model (Fig. 4B) identified the pseudoknot (stem 3) but with an open stem 2. Note that all these experimentally-guided models for the FSE region were estimated for specific local regions. As a result, the models are sensitive to the context and region boundaries (32, 35, 57) (see *SI Appendix*, S11D–F for alternative structures of Fig. 4B–D

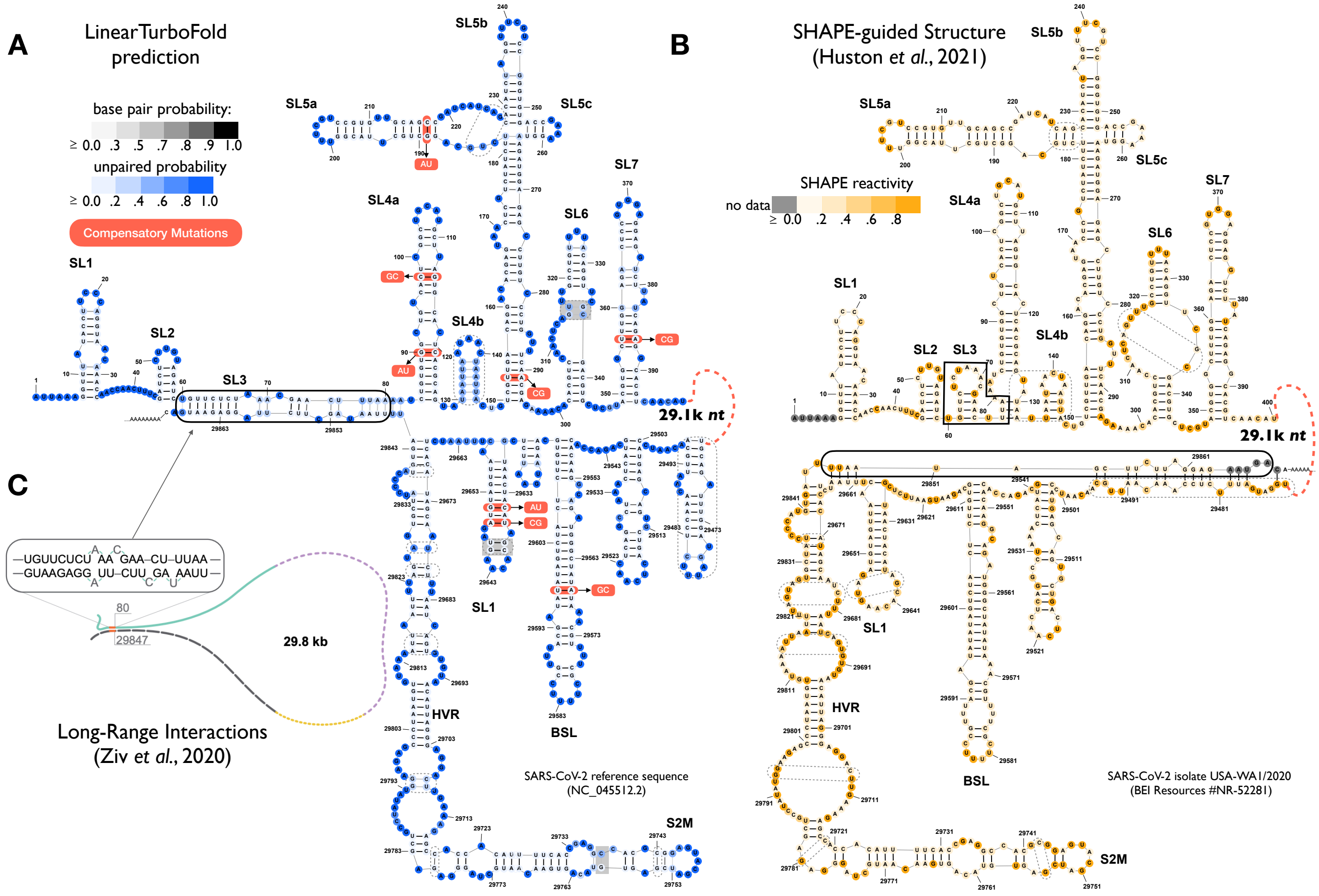

A
LinearTurboFold prediction
base pair probability:
≥ 0.0 .3 .5 .7 .8 .9 1.0
unpaired probability
≥ 0.0 .2 .4 .6 .8 1.0
Compensatory Mutations
SL1
SL2
SL3
SL4a
SL4b
SL5a
SL5b
SL5c
SL6
SL7
29.1k nt
HVR
BSL
S2M
SARS-CoV-2 reference sequence (NC_045512.2)
B
SHAPE-guided Structure (Huston et al., 2021)
SHAPE reactivity
no data
≥ 0.0 .2 .4 .6 .8
SARS-CoV-2 isolate USA-WA1/2020 (BEI Resources #NR-52281)
C
—UGUUCUCU AA GAA–CU–UUAA—
—GUAAGAGG UU–CUU GA AAUU—
80
29847
29.8 kb
Long-Range Interactions (Ziv et al., 2020)

Fig. 3. Secondary structures predictions of SARS-CoV-2 extended 5' and 3' UTRs. **A**: LinearTurboFold prediction.The nucleotides and base pairs are colored by unpaired probabilities and base-pairing probabilities, respectively. The compensatory mutations extracted by LinearTurboFold are annotated with alternative pairs in red boxes (see *SI Appendix*, Tab. S2 for more fully conserved pairs with co-variational changes). **B**: SHAPE-guided model by Huston *et al.* (32) (window size 3000 *nt* sliding by 300 *nt* with maximum pairing distance 500 *nt*). The nucleotides are colored by SHAPE reactivities. Dashed boxes enclose the different structures between **A** and **B**. Our model is close to Huston *et al.*'s, but the major difference is that LinearTurboFold predicts the end-to-end pairs involving 5' and 3' UTRs (solid box in **A**), which is *exactly* the same interaction detected by Ziv *et al.* using the COMRADES experimental technique (37) (**C**). Such long-range interactions cannot be captured by the local folding methods used by prior experimentally-guided models (Fig. 1B). The similarity between models A and B as well as the exact agreement between A and C show that our *in silico* method of folding multiple homologs can achieve results similar to, if not more accurate than, experimentally-guided single-genome prediction. As negative controls (*SI Appendix*, Fig. S10), the align-then-fold (RNAalifold) method cannot predict such long-range interactions. Although the single sequence folding algorithm (LinearPartition) predicts a long-range 5'-3' interaction, the positions are not the same as the LinearTurboFold model and Ziv *et al.*'s experimental result.

with different regions). LinearTurboFold, by contrast, does not suffer from this problem by virtue of global folding without local windows. Besides SARS-CoV-2, we notice that the estimated structure of the SARS-CoV-1 reference sequence (Fig. 4F) from LinearTurboFold is similar to SARS-CoV-2 (Fig. 4A), which is consistent with the observation that the structure of the FSE region is highly conserved among betacoronaviruses (40). Finally, as negative controls, both the single sequence folding algorithm (LinearPartition in Fig. 4G) and the align-then-fold method (RNAalifold in *SI Appendix*, Fig. S11G) predict quite different structures compared with the LinearTurboFold prediction (Fig. 4A) (39%/61% of pairs from the LinearTurboFold model are not found by LinearPartition/RNAalifold).

In addition to the well-studied UTRs and FSE regions, LinearTurboFold discovers 50 conserved structures with identical structures among 25 genomes, and 26 regions are novel compared to previous studies (31, 32) (Fig. 4H, *SI Appendix*, Tab. S3). These novel structures are potential targets for small-molecule drugs (41) and ASOs (36, 58). LinearTurboFold also recovers fully conserved base pairs with compensatory mutations (*SI Appendix*, Tab. S2), which imply highly conserved structural regions whose functions might not have been explored. We provide the complete multiple sequence alignment and predicted structures for 25 genomes from LinearTurboFold (Dataset S1; see *SI Appendix*, Fig. S14 for the format).

Highly Accessible and Conserved Regions in SARS-CoV-2 and SARS-related Betacoronaviruses. Studies show that the siRNA silencing efficiency, ASO inhibitory efficacy, CRISPR-Cas13 knockdown efficiency, and RT-PCR primer binding efficiency, all correlate with the target region's *accessibility* (43–45, 59), which is the probability of a target site being fully unpaired. However, most existing work for designing siRNAs, ASOs, CRISPR-Cas13 gRNAs, and RT-PCR primers does not take this feature into consideration (60, 61) (*SI Appendix*, Tab. S4). Here LinearTurboFold is able to provide more principled design candidates by identifying accessible regions of the target genome. In addition to accessibility, the emerging variants around the world reduce effectiveness of existing vaccines and test kits (*SI Appendix*, Tab. S4), which indicates sequence conservation is another critical aspect for therapeutic and diagnostic design. LinearTurboFold, being a tool for both structural alignment and ho-

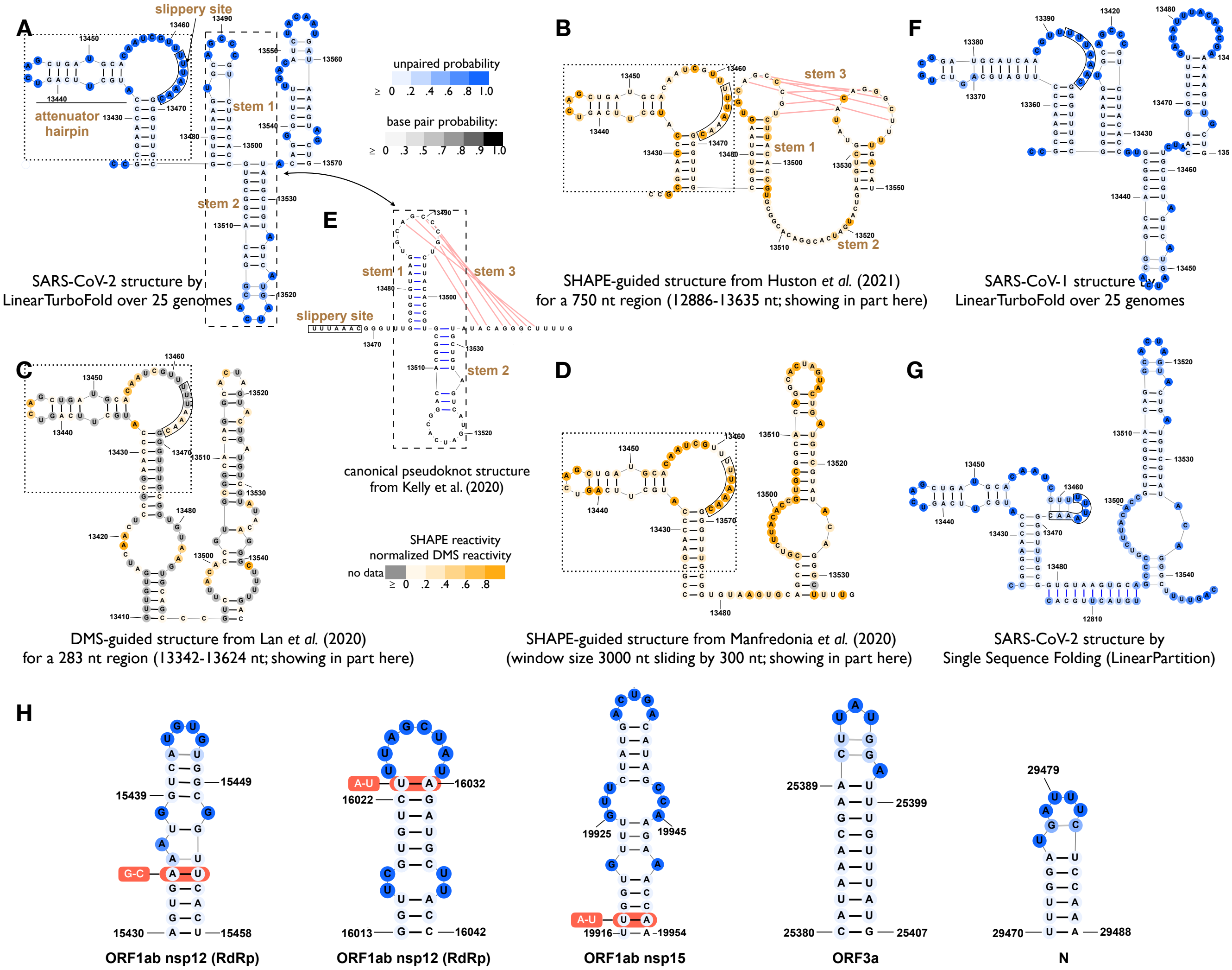
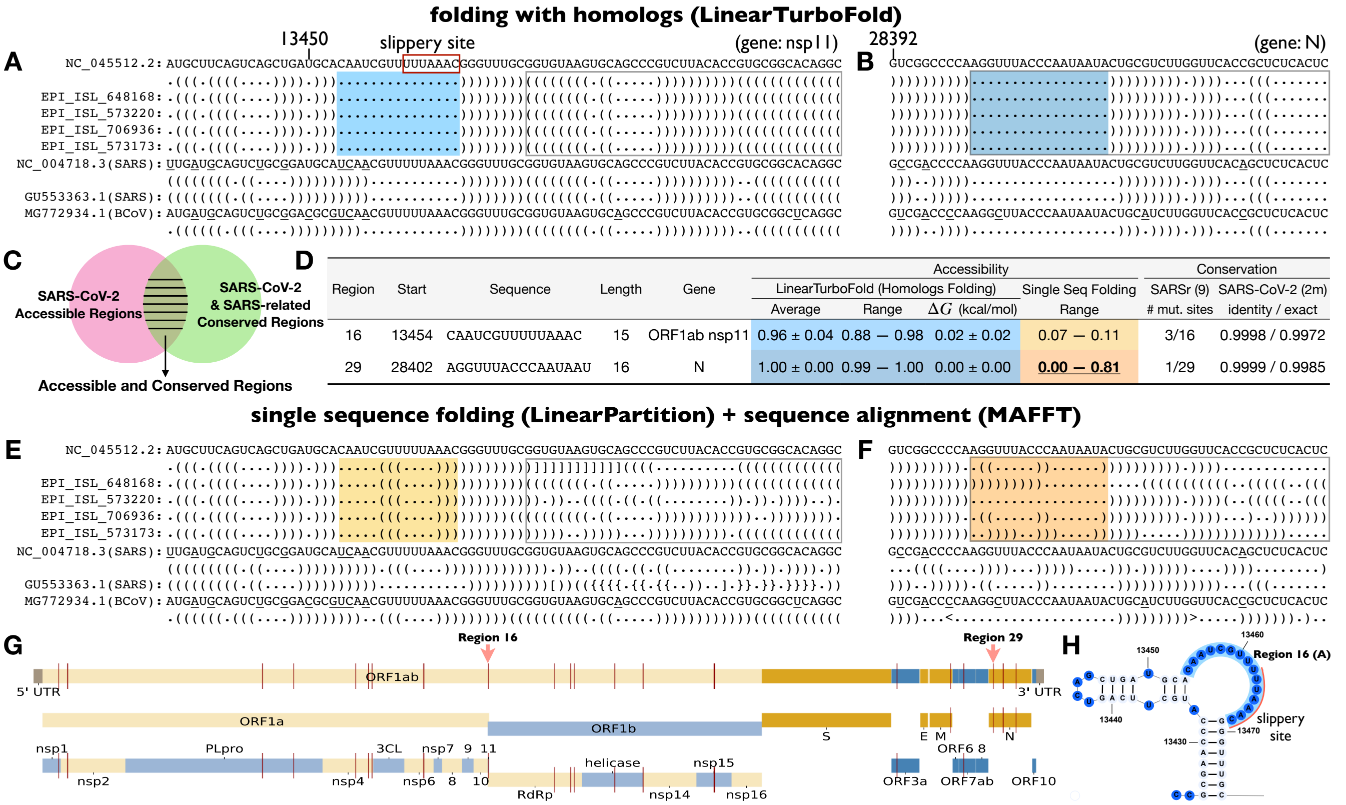


Fig. 4. **A–D**: Secondary structure predictions of SARS-CoV-2 extended frameshifting stimulation element (FSE) region (13425–13545 *nt*). **A**: LinearTurboFold prediction. **B–D**: Experimentally-guided predictions from the literature (32, 33, 35), which are sensitive to the context and region boundaries due to the use of local folding methods (*SI Appendix*, Fig. S11). **E**: The canonical pseudoknot structure by the comparative analysis between SARS-CoV-1 and SARS-CoV-2 genomes (40). For the 5' region of the FSE shown in dotted boxes (attenuator hairpin, internal loop with slippery site, and a stem), the LinearTurboFold prediction (A) is consistent with B–D; for the 3' region of the FSE shown in dashed boxes, our prediction (predicting stems 1–2 but missing 3) is closer to the canonical structure in E compared to B–D. **F**: LinearTurboFold prediction on SARS-CoV-1. **G**: Single sequence folding algorithm (LinearPartition) prediction on SARS-CoV-2, which is quite different from LinearTurboFold's. As another negative control, the align-then-fold method (RNAalifold) predicts a rather dissimilar structure (*SI Appendix*, Fig. S11G). **H**: Five examples from 59 fully conserved structures among 25 genomes (*SI Appendix*, Tab. S3), 26 of which are novel compared with prior work (31, 32).

mologous folding, can identify regions that are both (sequence-wise) conserved and (structurally) accessible, and it takes advantage of not only SARS-CoV-2 variants but also homologous sequences, e.g., SARS-CoV-1 and bat coronavirus genomes, to identify conserved regions from historical and evolutionary perspectives.

To get unstructured regions, Rangan *et al.* (31) imposed a threshold on unpaired probability of each position, which is a crude approximation because the probabilities are not independent of each other. By contrast, the widely-used stochastic sampling algorithm (50, 62) builds a representative ensemble of structures by sampling independent secondary structures according to their probabilities in the Boltzmann distribution. Thus the accessibility for a region can be approximated as the fraction of sampled structures in which the region is single-stranded. LinearTurboFold utilized LinearSampling (50) to generate 10,000 independent structures for each genome according to the modified partition functions after the iterative refinement (Fig. 1A module **5**), and calculated accessibilities for regions at least 15 *nt* long. We then define *accessible regions* that are with at least 0.5 accessibility among all 16 SARS-CoV-2 genomes (Fig. 5A–B). We also measure the free energy to open a target region $[i, j]$ (63), notated: $\Delta G_{\mathrm{u}}[i,j] = -RT(\log Z_{\mathrm{u}}[i,j] - \log Z) = -RT \log P_{\mathrm{u}}[i,j]$ where Z is the partition function which sums up the equilibrium constants of all possible secondary structures, $Z_{\mathrm{u}}[i,j]$ is the partition function over all structures in which the region $[i,j]$ is fully unpaired, R is the universal gas constant and T is the thermodynamic temperature. Therefore $P_{\mathrm{u}}[i,j]$ is the unpaired probability of the target region and can be approximated via sampling by s_0/s, where s is the sample size and s_0 is the number of samples in which the target region is single-stranded. The regions whose free energy changes are close to zero need less free energy to open, thus more accessible to bind with siRNAs, ASOs, CRISPR-Cas13 gRNAs and RT-PCR primers.

Next, to identify *conserved regions* that are highly conserved



Region	Start	Sequence	Length	Gene	Accessibility				Conservation	
					LinearTurboFold (Homologs Folding)			Single Seq Folding	SARSr (9)	SARS-CoV-2 (2m)
					Average	Range	ΔG (kcal/mol)	Range	# mut. sites	identity / exact
16	13454	CAAUCGUUUUUAAAC	15	ORF1ab nsp11	0.96 ± 0.04	0.88 — 0.98	0.02 ± 0.02	0.07 — 0.11	3/16	0.9998 / 0.9972
29	28402	AGGUUUACCCAAUAAU	16	N	1.00 ± 0.00	0.99 — 1.00	0.00 ± 0.00	**0.00 — 0.81**	1/29	0.9999 / 0.9985

Fig. 5. An illustration of accessible and conserved regions that LinearTurboFold identifies. **A–B**: Identified structurally-conserved accessible regions by LinearTurboFold with the help of considering alignment and folding simultaneously. The regions at least 15 *nt* long with accessibility of at least 0.5 among all the 16 SARS-CoV-2 genomes are shaded on blue background. Structures are encoded in dot-bracket notation. "(" and ")" indicates nucleotides pairing in the 3' and 5' direction, respectively. "." indicates an unpaired nucleotide. The positions with mutations compared to the SARS-CoV-2 reference sequence among three different subfamilies (SARS-CoV-2, SARS-CoV-1 and BCoV) are underlined. **C**: Accessible and conserved regions are not only *accessible* among SARS-CoV-2 genomes (pink circle) but also *conserved* (at sequence level) among both SARS-CoV-2 and SARS-related genomes (green circle). **D**: Two examples out of 33 accessible and conserved regions found by LinearTurboFold. Region 16 and Region 29 correspond to the accessible regions in **A** and **B**, respectively. Region 16 is also the long internal loop including the slippery site in the FSE region (**H**). The conservation of these regions on 9 SARS-related genomes is the number of mutated sites. The conservation on the ∼2M SARS-CoV-2 dataset is shown in both average sequence identity with the reference sequence and the percentage of exact matches, respectively. **E–F**: Single sequence folding algorithms predict greatly different structures even if the sequence identities are high (grey boxes). These two regions, fully conserved among SARS-CoV-2 genomes, still fold into different structures due to mutations outside the regions. **G**: The positions of these 33 regions (red bars) across the whole genome (*SI Appendix*, Tab. S5). All the accessible and conserved regions are potential targets for siRNAs, ASOs, CRISPR-Cas13 gRNAs and RT-PCR primers.

among both SARS-CoV-2 and SARS-related genomes, we require that these regions contain at most three mutated sites on the 9 SARS-related genomes compared to the SARS-CoV-2 reference sequence because historically conserved sites are also unlikely to change in the future (64), and the average sequence identity with reference sequence over a large SARS-CoV-2 dataset is at least 0.999 (here we use a dataset of ∼2M SARS-CoV-2 genomes submitted to GISAID up to June 30, 2021[†]; see **Methods §11**). Finally, we identified 33 *accessible and conserved regions* (Fig. 5G and *SI Appendix*, Tab. S5), which are not only structurally accessible among SARS-CoV-2 genomes but also highly conserved among SARS-CoV-2 and SARS-related genomes (Fig. 5C). Because the specificity is also a key factor influencing siRNA efficiency (65), we used BLAST against the human transcript dataset for these regions (*SI Appendix*, Tab. S5). Finally, we also listed the GC content of each region. Among these regions, region 16 corresponds to the internal loop containing the slippery site in the extended FSE region, and it is conserved at both structural and sequence levels (Fig. 5D and 5H). Besides SARS-CoV-2 genomes, the SARS-related genomes such as the SARS-CoV-1 reference sequence (NC_004718.3) and a bat coronavirus (BCoV, MG772934.1) also form similar structures around the slippery site (Fig. 5A). By removing the constraint of conservation on SARS-related genomes, we identified 38 additional candidate regions (*SI Appendix*, Tab. S6) that are accessible but only highly conserved on SARS-CoV-2 variants.

We also designed a negative control by analyzing the SARS-CoV-2 reference sequence alone using LinearSampling, which can also predict accessible regions. However, these regions are not structurally conserved among the other 15 SARS-CoV-2 genomes, resulting in vastly different accessibilities, except for one region in the M gene (*SI Appendix*, Tab. S7). The reason for this difference is that, even with a high sequence identity (over 99.9%), single sequence folding algorithms still predict greatly dissimilar structures for the SARS-CoV-2 genomes (Fig. 5E–F). Both regions (in nsp11 and N genes) are fully conserved among the 16 SARS-CoV-2 genomes, yet they still fold into vastly different structures due to mutations outside the regions; as a result, the accessibilities are either low (nsp11) or in a wide range (N) (Fig. 5D). Conversely, addressing this by folding each sequence with proclivity of base pairing inferred from all homologous sequences, LinearTurboFold structure predictions are more consistent with each other and thus can detect conserved structures (Fig. 5A–B).

[†] The average sequence identity is 0.9987 on that ∼2M dataset (downloaded on July 25, 2021).

Discussion

The constant emergence of new SARS-CoV-2 variants is reducing the effectiveness of exiting vaccines and test kits. To cope with this issue, there is an urgent need to identify conserved structures as promising targets for therapeutics and diagnostics that would work in spite of current and future mutations. Here we presented LinearTurboFold, an end-to-end linear-time algorithm for structural alignment and conserved structure prediction of RNA homologs, which is the first joint-fold-and-align algorithm to scale to full-length SARS-CoV-2 genomes without imposing any constraints on base-pairing distance. We also demonstrate that LinearTurboFold leads to significant improvement on secondary structure prediction accuracy as well as an alignment accuracy comparable to or higher than all benchmarks.

Unlike existing work on SARS-CoV-2 using local folding and single-sequence folding workarounds, LinearTurboFold enables unprecedented global structural analysis on SARS-CoV-2 genomes; in particular, it can capture long-range interactions, especially the one between 5' and 3' UTRs across the whole genome, which matches perfectly with a recent purely experiment work. Over a group of SARS-CoV-2 and SARS-related homologs, LinearTurboFold identifies not only conserved structures supported by compensatory mutations and experimental studies, but also accessible and conserved regions as vital targets for designing efficient small-molecule drugs, siRNAs, ASOs, CRISPR-Cas13 gRNAs and RT-PCR primers. LinearTurboFold is widely applicable to the analysis of other RNA viruses (influenza, Ebola, HIV, Zika, etc.) and full-length genome analysis.

Methods

§1 Pairwise Hidden Markov Model. We use a pairwise Hidden Markov Model (pair-HMM) to align two sequences (51, 66). The model includes three actions (h): aligning two nucleotides from two sequences (ALN), inserting a nucleotide in the first sequence without a corresponding nucleotide in the other sequence (INS1), and a nucleotide insertion in the second sequence without a corresponding nucleotide in the first sequence (INS2). We then define $\mathcal{A}(\mathbf{x}, \mathbf{y})$ as a set of all the possible alignments for the two sequences, and one alignment $a \in \mathcal{A}(\mathbf{x}, \mathbf{y})$ as a sequence of steps (h, i, j) with $m+2$ steps, where (h, i, j) means an alignment step at the position pair (i, j) by the action h. Thus, for the lth step $a_l = (h_l, i_l, j_l) \in a$, the values of i_l and j_l depend on the action h_l and the positions i_{l-1} and j_{l-1} of a_{l-1}:

$$a_l = \begin{cases} (\textsc{aln}, & i_{l-1}+1, & j_{l-1}+1), & h_l = \textsc{aln} \\ (\textsc{ins1}, & i_{l-1}+1, & j_{l-1}), & h_l = \textsc{ins1} \\ (\textsc{ins2}, & i_{l-1}, & j_{l-1}+1), & h_l = \textsc{ins2} \end{cases}$$

with $(\textsc{aln}, 0, 0)$ as the first step, and $(\textsc{aln}, |\mathbf{x}|+1, |\mathbf{y}|+1)$ as the last one. For two sequences $\{\textsc{acaagu}, \textsc{aacug}\}$, one possible alignment $\{-\textsc{acaagu}, \textsc{aac}{-}{-}\textsc{ug}\}$ can be specified as $\{(\textsc{aln}, 0, 0) \rightarrow (\textsc{ins2}, 0, 1) \rightarrow (\textsc{aln}, 1, 2) \rightarrow (\textsc{aln}, 2, 3) \rightarrow (\textsc{ins1}, 3, 3) \rightarrow (\textsc{ins1}, 4, 3) \rightarrow (\textsc{aln}, 5, 4) \rightarrow (\textsc{aln}, 6, 5) \rightarrow (\textsc{aln}, 7, 6)\}$, where a gap symbol $(-)$ represents a nucleotide insertion in the other sequence at the corresponding position (*SI Appendix*, Tab. S3). The action h_l in each step (h_l, i_l, j_l) corresponds to a line segment starting from the previous node (i_{l-1}, j_{l-1}) and stopping at the node (i_l, j_l). Thus the line segment is horizontal, vertical or diagonal towards the top-right corner when h_l is INS1, INS2 or ALN, respectively (*SI Appendix*, Tab. S3).

We initialize the first step with the state ALN of probability 1, thus $p_\pi(\textsc{aln}) = 1$. $p_t(h_2 \mid h_1)$ is the transition probability from the state h_1 to h_2, and $p_e((c_1, c_2) \mid h_1)$ is the probability of the state h_1 emitting a character pair (c_1, c_2) with values from $\{\text{A, G, C, U}, -\}$. Both the emission and transition probabilities were taken from TurboFold II. The function $e()$ yields a character pair based on a_l and the nucleotides of two sequences:

$$e(\mathbf{x}, \mathbf{y}, a_l) = \begin{cases} (x_{i_l}, y_{j_l}), & h_l = \textsc{aln} \\ (x_{i_l}, -), & h_l = \textsc{ins1} \\ (-, y_{j_l}), & h_l = \textsc{ins2} \end{cases}$$

where x_i and y_i are the ith and jth nucleotides of sequences $\mathbf{x}$ and $\mathbf{y}$, respectively. Note that the first step $a_0 = (\textsc{aln}, 0, 0)$ and the last $a_{m+1} = (\textsc{aln}, |\mathbf{x}|+1, |\mathbf{y}|+1)$ do not have emissions.

We denote forward probability $\alpha^h_{i,j}$ encompassing the probability of the partial alignments of $\mathbf{x}$ and $\mathbf{y}$ up to positions i and j, and all the alignments that go through the step (h, i, j):

$$\begin{aligned} \alpha^h_{i,j} &= \sum_{\substack{a \in \mathcal{A}(\mathbf{x},\mathbf{y}) \\ \exists k, a_k = (h,i,j)}} p(\mathbf{x}, \mathbf{y}, a[:k]) \\ &= p_\pi(h_0) \cdot \prod_{l=1}^{k} p_t(h_l \mid h_{l-1}) p_e(e(\mathbf{x}, \mathbf{y}, a_l) \mid h_l) \end{aligned}$$

where $a[:k]$ indicates the partial alignments from the starting node up to the kth step and $a_k = (h, i, j)$. For instance, $\alpha^{\textsc{aln}}_{3,3}$, $\alpha^{\textsc{ins1}}_{3,3}$ and $\alpha^{\textsc{ins2}}_{3,3}$ corresponds to the region circled by the blue dashed lines (*SI Appendix*, Tab. S3B, C and D). Similarly, the backward probability $\beta^h_{i,j}$ assembles the probability of partial alignments $a[k+1:]$ from the $(k+1)$th step up to the end one:

$$\begin{aligned} \beta^h_{i,j} &= \sum_{\substack{a \in \mathcal{A}(\mathbf{x},\mathbf{y}) \\ \exists k, a_k = (h,i,j)}} p(\mathbf{x}, \mathbf{y}, a[k+1:]) \\ &= \left\{ \prod_{l=k+1}^{m} p_t(h_l \mid h_{l-1}) p_e(e(\mathbf{x}, \mathbf{y}, a_l) \mid h_l) \right\} \cdot p_t(h_{m+1} \mid h_m) \end{aligned}$$

For example, $\beta^{\textsc{aln}}_{3,3}$, $\beta^{\textsc{ins1}}_{3,3}$ and $\beta^{\textsc{ins2}}_{3,3}$ are the regions circled by the yellow dashed line (*SI Appendix*, Tab. S3B, C and D). Thus, the probability of observing two sequences $p(\mathbf{x}, \mathbf{y})$ is $\alpha^{\textsc{aln}}_{|\mathbf{x}|+1,|\mathbf{y}|+1}$ or $\beta^{\textsc{aln}}_{0,0}$.

§2 Posterior Co-incidence Probability Computation. Nucleotide positions i and j in two sequences $\mathbf{x}$ and $\mathbf{y}$ are said to be *co-incident* (notated as $i \sim j$) in an alignment a if the alignment path goes through the node (i, j) (51). Since the node (i, j) is reachable by three actions $\mathcal{H} = \{\textsc{aln}, \textsc{ins1}, \textsc{ins2}\}$, the co-incidence probability for a position pair (i, j) given two sequences is:

$$p(i \sim j \mid \mathbf{x}, \mathbf{y}) = \frac{1}{p(\mathbf{x}, \mathbf{y})} \sum_{\substack{a \in \mathcal{A}(\mathbf{x},\mathbf{y}) \\ \exists h, (h,i,j) \in a}} p(\mathbf{x}, \mathbf{y}, a) \qquad [1]$$

where $p(\mathbf{x}, \mathbf{y}, a)$ is the probability of two sequences with the alignment a, and $p(\mathbf{x}, \mathbf{y})$ is the probability of observing two sequences, which is the sum of probability of all the possible alignments:

$$p(\mathbf{x}, \mathbf{y}) = \sum_{a \in \mathcal{A}(\mathbf{x},\mathbf{y})} p(\mathbf{x}, \mathbf{y}, a)$$

The co-incidence probability for positions i and j (Equation 1) can be computed by:

$$p(i \sim j \mid \mathbf{x}, \mathbf{y}) = \frac{\sum_h \alpha^h_{i,j} \cdot \beta^h_{i,j}}{\alpha^{\textsc{aln}}_{|\mathbf{x}|+1,|\mathbf{y}|+1}}$$

§3 LinearAlignment. Unlike a previous method (51) that fills out all the nodes in the alignment matrix by columns (*SI Appendix*, Fig. S3). LinearAlignment scans the matrix based on the *step count* s, which is the sum value of i and j ($s = i + j$) for the partial alignments of $\mathbf{x}_{[1,i]}$ and $\mathbf{y}_{[1,j]}$. As shown in the pseudocode (*SI Appendix*, Fig. S4), the forward phase starts from the node $(0, 0)$ in the state ALN of probability 1, then iterates the step count s from 0 to $|\mathbf{x}| + |\mathbf{y}| - 1$. For each step count s with a specific state h from $\mathcal{H}$, we first collect all the nodes (i, j) with the step count s with $\alpha^h_{i,j}$ existing, which means the position pair (i, j) has been visited via the state h before. Then each node makes transitions to next nodes by there states, and updates the corresponding forward probabilities $\alpha^{\textsc{ins1}}_{i+1,j}$, $\alpha^{\textsc{ins2}}_{i,j+1}$ and $\alpha^{\textsc{aln}}_{i+1,j+1}$, respectively.

The current alignment algorithm is still an exhaustive-search algorithm and costs quadratic time and space for all the $|\mathbf{x}| \times |\mathbf{y}|$ nodes. To reduce the runtime, LinearAlignment uses the beam search heuristic algorithm (48) and keeps a limited number of promising nodes at each step. For each step count s with a state h, LinearAlignment applies the beam search method first over $B(s, h)$, which is the collection of all the nodes (i, j) with step count s and the presence of $\alpha^h_{i,j}$ (*SI Appendix*, Fig. S4 line 6).This algorithm only saves the top b_{aln} nodes with the highest forward scores in $B(s, h)$, and these are subsequently allowed to make transitions to the next states. Here b_{aln} is a user-specified beam size and the default value is 100. In total, $O(b_{aln}n)$ nodes survive because the length of s is $|\mathbf{x}| + |\mathbf{y}|$ and each step count keeps b_{aln} nodes. For simplicity, we show the topological order and the beam search method with alignment examples (*SI Appendix*, Fig. S3A), while the forward-backward algorithm adopts the same idea by summing the probabilities of all the possible alignments.

After the forward phase, the backward phase (*SI Appendix*, Fig. S4) performs in linear time to calculate the co-incidence probabilities automatically because only a linear number of nodes in $B(s, h)$ are stored. Thus by pruning low-scoring candidates at each step in the forward algorithm, we reduce the runtime from $O(n^2)$ to $O(b_{aln}n)$ for aligning two sequences. For k input homologous sequences, LinearTurboFold computes posterior co-incidence probabilities for each pair of sequences by LinearAlignment, which costs $O(k^2 b_{aln} n)$ runtime in total.

§4 Match Scores Computation and Modified LinearAlignment. To encourage the pairwise alignment conforming with estimated secondary structures, LinearTurboFold predicts structural alignments by incorporating the secondary structural conformation. PMcomp (67) first proposed the match score to measure the structural similarity for position pairs between a pair of sequences, and TurboFold II adapts it as a prior. Based on the base pair probabilities $P_{\mathbf{x}}(i, j)$ estimated from the partition function for a sequence $\mathbf{x}$, a position i could be paired with bases upstream, downstream or unpaired, with corresponding probability $P_{\mathbf{x},>}(i) = \sum_{j<i} P_{\mathbf{x}}(i, j)$, $P_{\mathbf{x},<}(i) = \sum_{j>i} P_{\mathbf{x}}(i, j)$ and $P_{\mathbf{x},o}(i) = 1 - P_{\mathbf{x},>}(i) - P_{\mathbf{x},<}(i)$, respectively. The match score $m_{\mathbf{x},\mathbf{y}}(i, j)$ for two positions i and j from two sequences $\mathbf{x}$ and $\mathbf{y}$ is based on the probabilities of these three structural propensities from the last iteration $(t - 1)$:

$$m^{(t)}_{\mathbf{x},\mathbf{y}}(i,j) = \alpha_1 \left[\sqrt{P^{(t-1)}_{\mathbf{x},>}(i)\cdot P^{(t-1)}_{\mathbf{y},>}(j)}\sqrt{P^{(t-1)}_{\mathbf{x},<}(i)\cdot P^{(t-1)}_{\mathbf{y},<}(j)}\right] + \alpha_2\sqrt{P^{(t-1)}_{\mathbf{x},o}(i)\cdot P^{(t-1)}_{\mathbf{y},o}(j)} + \alpha_3$$

where α_1, α_2 and α_3 are weight parameters trained in TurboFold II. The forward-backward phrases integrate the match score as a prior when aligning two nucleotides (*SI Appendix*, Fig. S4 line 10 and line 12).

TurboFold II separately pre-computes match scores for all the $O(n^2)$ position pairs for pairs of sequences before the HMM alignment calculation. However, only a linear number of pairs $O(b_{aln}n)$ survive after applying the beam pruning in LinearAlignment. To reduce redundant time and space usage, LinearTurboFold calculates the corresponding match scores for co-incident pairs when they are first visited in LinearAlignment. Overall, for k homologous sequences, LinearTurboFold reduces the runtime of the whole module of pairwise posterior co-incidence probability computation from $O(k^2n^2)$ to $O(k^2 b_{aln} n)$ by applying the beam search heuristic to the pairwise HMM alignment, and only calculating the match scores for position pairs that are needed.

§5 Extrinsic Information Calculation. To update partition functions for each sequence with the structural information from homologs, TurboFold (28) introduces *extrinsic information* to model the the proclivity for base pairing induced from the other sequences in the input set $\mathcal{S}$. The extrinsic information $e_{\mathbf{x}}(i, j)$ for a base pair (i, j) in the sequence $\mathbf{x}$ maps the estimated base pairing probabilities of other sequences to the target sequence via the co-incident nucleotides between each pair of sequences:

$$\sum_{\mathbf{y}\in\{\mathcal{S}\backslash\mathbf{x}\}}(1-s_{\mathbf{x},\mathbf{y}})\sum_{k,l} p^{(t-1)}_{\mathbf{y}}(k,l)\cdot p^{(t)}_{\mathbf{x},\mathbf{y}}(i\sim k)\cdot p^{(t)}_{\mathbf{x},\mathbf{y}}(j\sim l)$$

where $p^{(t-1)}_{\mathbf{y}}(k, l)$ is the base pair probability for a base pair (k, l) in the sequence $\mathbf{y}$ from $(t - 1)$th iteration. $p^{(t)}_{\mathbf{x},\mathbf{y}}(i \sim k)$ and $p^{(t)}_{\mathbf{x},\mathbf{y}}(j \sim l)$ are the posterior co-incidence probabilities for position pairs (i, k) and (j, l), respectively, from (t)th iteration. The extrinsic information $e^{(t)}_{\mathbf{x}}(i, j)$ first sums all the base pair probabilities of alignable pairs from another one sequence with the co-incidence probabilities and then iterates over all the other sequences. $s_{\mathbf{x},\mathbf{y}}$ is the sequence identity for sequences $\mathbf{x}$ and $\mathbf{y}$. The sequences with a low identity contribute more to the extrinsic information than sequences of higher identity. The sequence identity is defined as the fraction of nucleotides that are aligned and identical in the alignment.

§6 LinearPartition for Base Pairing Probabilities Estimation with Extrinsic Information. 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 (46) algorithm approximates the partition function and base paring probability matrix computation in linear time. LinearPartition is significantly faster, and correlates better with the ground truth structures than the traditional cubic partition function calculation. Thus LinearTurboFold uses LinearPartition to predict base pair probabilities instead of the traditional $O(n^3)$-time partition function.

TurboFold introduces the extrinsic information $e^{(t)}_{\mathbf{x}}(i, j)$ in the partition function as a pseudo-free energy term for each base pair (i, j). Similarly, in LinearPartition, for each span $[i, j]$, which is the subsequence $x_i...x_j$, and its associated partition function $Q(i, j)$, the partition function is modified as $\tilde{Q}(i, j) = Q(i, j)e^{(t)}_{\mathbf{x}}(i, j)^{\lambda}$ 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 $Q(i, j)e^{(t)}_{\mathbf{x}}(i, j)^{\lambda}$ 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.

Similarly, 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. Overall, for k homologous sequences, LinearTurboFold reduces the runtime of base pair probabilities estimation for each sequence from $O(kn^3 + k^2n^2)$ to $O(kb^2_{aln}n + k^2 b_{fld} n)$ by applying the beam search b_{fld} to the partition function calculation, and only calculating extrinsic information for the saved base pairs.

§7 MSA Generation and Secondary Structure Prediction. After several iterations, TurboFold II builds the multiple sequence alignment using a probabilistic consistency transformation, generating a guide tree and performing progressive alignment over the pairwise posterior co-incidence probabilities (30). The whole procedure is accelerated in virtue of the sparse matrix by discarding alignment pairs of probability smaller than a threshold (0.01 by default). Since LinearAlignment uses the beam search method and only saves a linear number of co-incident pairs, the MSA generation in LinearTurboFold costs linear runtime against the sequence length straightforwardly.

Estimated base pair probabilities are fed into downstream methods to predict secondary structures. To maintain the end-to-end linear-time property, LinearTurboFold uses ThreshKnot (49), which is a thresholded version of ProbKnot (68) and only considers base pairs of probability exceeding a 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, 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 (*SI Appendix*, Fig. S8B).

§8 Efficiency and Scalability Datasets. Four datasets are built and used for measuring efficiency and scalability. To evaluate the efficiency and 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 a fixed group size 5. Sequences are sampled from RNAStrAlign dataset (27), the Comparative RNA Web (CRW) Site (69), the Los Alamos HIV database (http://www.hiv.lanl.gov/) and the SARS-related betacoronaviruses (SARS-related) (53). RNAStrAlign, aggregated and released with TurboFold II, is an RNA alignment and structure database. Sequences in RNAStrAlign are categorized into families, i.e. sets of homologs, 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 SARS-related sequences (of sequence length ranging from 29,484 *nt* to 29,903 *nt*). All the sequences in one group belong to the same subfamily or subtype. We sampled five groups for each family and obtained 35 groups in total. Due to the runtime and memory limitations, we did not run TurboFold II on SARS-CoV-2 groups (Fig. 2, A and D).

To assess the runtime and memory usage of LinearTurboFold with group size, we fixed the sequence length around 1,500 *nt*, and sampled 5 groups of sequences from the small subunit ribosomal RNA (Alphaproteobacteria subfamily) with group size 5, 10, 15 and 20, respectively (Fig. 2, B and F). We used 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.

We built a test set from the RNAStrAlign dataset to measure and compare the performance between LinearTurboFold and other methods. 60 groups of input sequences consisting of five homologous sequences were randomly selected from the small subunit ribosomal RNA (rRNA) (Alphaproteobacteria subfamily), SRP RNA (Protozoan subfamily), RNase P RNA (bacterial type A subfamily) and telomerase RNA families from RNAStrAlign dataset. We removed sequences shorter than 1,200 *nt* for the small subunit rRNA to filter out subdomains, and removed sequences that are shorter than 200 *nt* for SRP RNA following the TurboFold II paper to filter out less reliable sequences. We resampled the test set five times and show the average PPV, Sensitivity and F1 scores over the five samples (Fig. 2, C and F).

An RNAStrAlign training set was built to compare accuracies between MEA and ThreshKnot. 40 groups of 3, 5 and 7 homologs were randomly sampled from 5S ribosomal RNA (Eubacteria subfamily), group I intron (IC1 subfamily), tmRNA, and tRNA families from RNAStrAlign dataset. We chose $\theta = 0.1, 0.2, 0.3, 0.4$ and 0.5 for ThreshKnot, and $\gamma = 1, 1.5, 2, 2.5, 3, 3.5, 4, 8$ and 16 for MEA. We reported the average secondary structure prediction accuracies (PPV and Sensitivity) across all training families (*SI Appendix*, Fig. S8B).

§9 Benchmarks. The Sankoff algorithm (15) 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 length n. Both LocARNA (16) and MXSCARNA (18) are Sankoff-style algorithms.

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 (70) 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 $O(k^2n^2)$ with a limited searching space of folding and alignment. Both MXSCARNA and LocARNA uses pre-computed base pair probabilities for each sequence as structural input. All the benchmarks use the default options and hyper-parameters 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 ($b_{aln} = b_{fld} = 100$) in LinearAlignment and LinearPartition, then predicts structures with ThreshKnot ($\theta = 0.3$).

§10 Significance Test. We use a paired, two-tailed permutation test (71) to measure the significant difference. Following the common practice, the repetition number is 10,000, and the significance threshold α is 0.05.

§11 SARS-CoV-2 Datasets. We used two large SARS-CoV-2 datasets. The first dataset is used to draw a representative sample of most diverse SARS-CoV-2 genomes. We downloaded all the genomes submitted to GISAID (52) by December 29, 2020 (downloaded on December 29, 2020), and filtered out low-quality genomes (with more than 5% unknown characters and degenerate bases, shorter than 29,500 *nt*, or with framing error in the coding region), and we also discard genomes with more than 600 mutations compared with the SARS-CoV-2 reference sequence (NC_0405512.2) (72). After preprocessing, this dataset includes about 258,000 genomes. To identify a representative group of samples with more variable mutations, we designed a greedy algorithm to select 16 most diverse genomes genomes found at least twice in the 258,000 genomes. The general idea of the greedy algorithm is to choose genomes one by one with the most new mutations compared with the selected samples, which consists of only the reference sequence at the beginning.

The second, larger, dataset is to evaluate the conservation of regions with respect to more up-to-date variants. We did the same preprocessing as the first dataset on all the genomes submitted to GISAID by June 30, 2021 (downloaded on July 25, 2021). This resulted in a dataset of ∼2M genomes, which was used to evaluate conservation in Figure 5 and *SI Appendix*, Tab. S4–S6.

§12 Data Availability. Our code, data and complete results for 25 SARS-CoV-2 and SARS-related genomes are released at: https://github.com/LinearFold/LinearTurboFold.

ACKNOWLEDGMENTS. This work is supported in part by National Institutes of Health (R01 GM132185 to D.H.M.) and National Science Foundation (IIS-1817231 to L.H.).

1. SR Eddy., Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.* **2**, 919–929 (2001).
2. JA Doudna, TR Cech, The chemical repertoire of natural ribozymes. *Nature* **418**, 222–228 (2002).
3. EP Nawrocki, SR Eddy, Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* **29**, 2933–2935 (2013).
4. EA Brown, H Zhang, LH Ping, SM Lemon, Secondary structure of the 5' nontranslated regions of hepatitis C virus and pestivirus genomic RNAs. *Nucleic Acids Res.* **20**, 5041–5045 (1992).
5. J Ritz, JS Martin, A Laederach, Evolutionary evidence for alternative structure in RNA sequence co-variation. *PLoS Comput. Biol.* **9**, e1003152–e1003152 (2013).
6. E Rivas, J Clements, SR Eddy, Estimating the power of sequence covariation for detecting conserved RNA structure. *Bioinformatics* **36**, 3072–3076 (2020).
7. RW Holley, et al., Structure of a ribonucleic acid. *Science* pp. 1462–1465 (1965).
8. HF Noller, et al., Secondary structure model for 23S ribosomal RNA. *Nucleic Acids Res.* **9**, 6167–6189 (1981).
9. NR Pace, DK Smith, GJ Olsen, BD James, Phylogenetic comparative analysis and the secondary structure of ribonuclease P RNA—a review. *Gene* **82**, 65–75 (1989).
10. K Williams, D Bartel, Phylogenetic analysis of tmRNA secondary structure. *RNA* **2**, 1306–1310 (1996).
11. M Levitt, Detailed molecular model for transfer ribonucleic acid. *Nature* **224**, 759–763 (1969).
12. RR Gutell, JC Lee, JJ Cannone, The accuracy of ribosomal RNA comparative structure models. *Curr. opinion structural biology* **12**, 301–310 (2002).
13. JH Havgaard, J Gorodkin, RNA structural alignments, part I: Sankoff-based approaches for structural alignments in *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods.* (Springer), pp. 275–290 (2014).
14. K Asai, M Hamada, RNA structural alignments, part II: non-Sankoff approaches for structural alignments in *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods.* (Springer), pp. 291–301 (2014).
15. D Sankoff, Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. on Appl. Math.* **45**, 810—-825 (1985).
16. S Will, K Reiche, IL Hofacker, PF Stadler, R Backofen, Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.* **3**, e65 (2007).
17. JH Havgaard, E Torarinsson, J Gorodkin, Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Comput. Biol.* **3**, 1896–1908 (2007).
18. Y Tabei, H Kiryu, T Kin, K Asai, A fast structural multiple alignment method for long RNA sequences. *BMC Bioinforma.* **9**, 33 (2008).
19. Z Xu, DH Mathews, Multilign: an algorithm to predict secondary structures conserved in multiple RNA sequences. *Bioinformatics* **27**, 626–632 (2011).
20. DH Mathews, DH Turner, Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.* **317**, 191–203 (2002).
21. K Sato, Y Kato, T Akutsu, K Asai, Y Sakakibara, DAFS: simultaneous aligning and folding of RNA sequences via dual decomposition. *Bioinformatics* **28**, 3218–3224 (2012).
22. MS Waterman, Computer analysis of nucleic acid sequences in *Methods in enzymology.* (Elsevier) Vol. 164, pp. 765–793 (1988).
23. SH Bernhart, IL Hofacker, S Will, AR Gruber, PF Stadler, RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinforma.* **9**, 1–13 (2008).
24. MS Waterman, Consensus methods for folding single-stranded nucleic acids. *Math. methods for DNA sequences/editor, Michael S. Waterman* (1989).
25. M Hochsmann, T Toller, R Giegerich, S Kurtz, Local similarity in RNA secondary structures in *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003.* (IEEE), pp. 159–168 (2003).
26. S Siebert, R Backofen, MARNA: A server for multiple alignment of RNAs. in *German Conference on Bioinformatics.* pp. 135–140 (2003).
27. Z Tan, Y Fu, G Sharma, DH Mathews, TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res.* **45**, 11570–11581 (2017).
28. AO Harmanci, G Sharma, DH Mathews, TurboFold: iterative probabilistic estimation of secondary structures for multiple RNA sequences. *BMC Bioinforma.* **12**, 108 (2011).
29. K Katoh, DM Standley, MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.* **30**, 772–780 (2013).
30. CB Do, MS Mahabhashyam, M Brudno, S Batzoglou, ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.* **15**, 330–340 (2005).
31. R Rangan, et al., RNA genome conservation and secondary structure in SARS-CoV-2 and SARS-related viruses: a first look. *RNA* **26**, 937–959 (2020).
32. NC Huston, et al., Comprehensive in vivo secondary structure of the SARS-CoV-2 genome reveals novel regulatory motifs and mechanisms. *Mol. cell* **81**, 584–598 (2021).
33. I Manfredonia, et al., Genome-wide mapping of SARS-CoV-2 RNA structures identifies therapeutically-relevant elements. *Nucleic Acids Res.* **48**, 12436–12452 (2020).
34. C Iserman, et al., Genomic RNA elements drive phase separation of the SARS-CoV-2 nucleocapsid. *Mol. cell* **80**, 1078–1091 (2020).
35. TC Lan, et al., Structure of the full SARS-CoV-2 RNA genome in infected cells. *BioRxiv* (2020).
36. L Sun, et al., In vivo structural characterization of the SARS-CoV-2 rna genome identifies host proteins vulnerable to repurposed drugs. *Cell* **184**, 1865–1883 (2021).
37. O Ziv, et al., The short- and long-range RNA-RNA interactome of SARS-CoV-2. *Mol. cell* **80**, 1067–1077 (2020).
38. JS Reuter, DH Mathews, RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinforma.* **11**, 1–9 (2010).
39. R Lorenz, et al., ViennaRNA package 2.0. *Algorithms for Mol. Biol.* **6**, 1 (2011).
40. JA Kelly, et al., Structural and functional conservation of the programmed- 1 ribosomal frameshift signal of sars coronavirus 2 (sars-cov-2). *J. Biol. Chem.* **295**, 10741–10748 (2020).
41. HS Haniff, et al., Targeting the SARS-CoV-2 RNA genome with small molecule binders and

ribonuclease targeting chimera (RIBOTAC) degraders. *ACS Cent. Sci.* **6**, 1713–1721 (2020).

42. ZJ Lu, DH Mathews, Fundamental differences in the equilibrium considerations for siRNA and antisense oligodeoxynucleotide design. *Nucleic Acids Res.* **36**, 3738–3745 (2008).
43. S Schubert, A Grünweller, VA Erdmann, J Kurreck, Local RNA target structure influences siRNA efficacy: systematic analysis of intentionally designed binding regions. *J. Mol. Biol.* **348**, 883–893 (2005).
44. OO Abudayyeh, et al., RNA targeting with CRISPR–Cas13. *Nature* **550**, 280–284 (2017).
45. SA Bustin, T Nolan, Pitfalls of quantitative real-time reverse-transcription polymerase chain reaction. *J. Biomol. Tech. JBT* **15**, 155 (2004).
46. H Zhang, L Zhang, DH Mathews, L Huang, LinearPartition: linear-time approximation of RNA folding partition function and base-pairing probabilities. *Bioinformatics* **36**, i258–i267 (2020).
47. JS McCaskill, The equilibrium partition function and base pair probabilities for RNA secondary structure. *Biopolymers* **29**, 11105–1119 (1990).
48. L Huang, K Sagae, Dynamic programming for linear-time incremental parsing in *Proceedings of ACL 2010.* (ACL, Uppsala, Sweden), p. 1077–1086 (2010).
49. L Zhang, H Zhang, DH Mathews, L Huang, ThreshKnot: Thresholded probknot for improved RNA secondary structure prediction. *BioRxiv* (2019).
50. H Zhang, L Zhang, S Li, D Mathews, L Huang, LinearSampling: Linear-time stochastic sampling of RNA secondary structure with applications to SARS-CoV-2. *BioRxiv* (2020).
51. AO Harmanci, G Sharma, DH Mathews, Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. *BMC Bioinforma.* **8**, 130 (2007).
52. S Elbe, G Buckland-Merrett, Data, disease and diplomacy: GISAID's innovative contribution to global health. *Glob. Challenges* **1**, 33–46 (2017).
53. C Ceraolo, FM Giorgi, Genomic variance of the 2019-nCoV coronavirus. *J. Med. Virol.* **92**, 522–528 (2020).
54. MG Seetin, DH Mathews, RNA structure prediction: an overview of methods in *Bacterial Regulatory RNA.* (Springer), pp. 99–122 (2012).
55. TJ Li, CM Reidys, The rainbow spectrum of RNA secondary structures. *Bull. Math. Biol.* **80**, 1514–1538 (2018).
56. WJC Lai, et al., mRNAs and lncRNAs intrinsically form secondary structures with short end-to-end distances. *Nat. Commun.* **9**, 4328 (2018).
57. R Rangan, et al., De novo 3D models of SARS-CoV-2 RNA elements from consensus experimental secondary structures. *Nucleic Acids Res.* **49**, 3092–3108 (2021).
58. V Lulla, et al., The stem loop 2 motif is a site of vulnerability for SARS-CoV-2. *BioRxiv* pp. 2020–09 (2021).
59. ZJ Lu, DH Mathews, Efficient siRNA selection using hybridization thermodynamics. *Nucleic Acids Res.* **36**, 640–647 (2008).
60. SA Bustin, et al., The MIQE guidelines: Minimum information for publication of quantitative real-time pcr experiments. *Clin. Chem.* **55**, 611–622 (2009).
61. M Park, J Won, BY Choi, CJ Lee, Optimization of primer sets and detection protocols for SARS-CoV-2 of coronavirus disease 2019 (COVID-19) using PCR and real-time PCR. *Exp. & Mol. Medicine* **52**, 963–977 (2020).
62. Y Ding, CE Lawrence, A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.* **31**, 7280–7301 (2003).
63. U Mückstein, et al., Thermodynamics of RNA–RNA binding. *Bioinformatics* **22**, 1177–1182 (2006).
64. SR Eddy, R Durbin, RNA sequence analysis using covariance models. *Nucleic Acids Res.* **22**, 2079–2088 (1994).
65. E Fakhr, F Zare, L Teimoori-Toolabi, Precise and efficient siRNA design: a key point in competent gene silencing. *Cancer Gene Ther.* **23**, 73–82 (2016).
66. R Durbin, S Eddy, A Krogh, G Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* (Cambridge University Press, Cambridge, UK), (1998).
67. IL Hofacker, SH Bernhart, PF Stadler, Alignment of RNA base pairing probability matrices. *Bioinformatics* **20**, 2222–2227 (2004).
68. S Bellaousov, DH Mathews, ProbKnot: fast prediction of RNA secondary structure including pseudoknots. *RNA* **16**, 1870–1880 (2010).
69. JJ Cannone, et al., The comparative RNA web (CRW) site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BioMed Cent. Bioinforma.* **3** (2002).
70. Y Tabei, K Tsuda, T Kin, K Asai, SCARNA: fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments. *Bioinformatics* **22**, 1723–1729 (2006).
71. N Aghaeepour, HH Hoos, Ensemble-based prediction of RNA secondary structures. *BMC Bioinforma.* **14** (2013).
72. F Wu, et al., A new coronavirus associated with human respiratory disease in China. *Nature* **579**, 265–269 (2020).