Accurate Multiple Sequence Alignment of Ultramassive Genome Sets

Xiao Lai[1†], Haixin Luan[1†], Pu Tian[1,2*]

[1]School of Life Sciences, Jilin University, Changchun & 130012, China

[2]School of Artificial Intelligence, Jilin University, Changchun & 130012, China

[*]Corresponding author. Email: tianpu@jlu.edu.cn [†]These authors contributed equally to this work.

With ever increasing sequencing efficiency, there is a pressing need to tackle a presently intractable task of accurate multiple sequence alignment (MSA) for ultramassive genome sets of millions and beyond. Additionally, efficient graph and probabilistic representations for downstream analysis are in dire lack. In light of these challenges, we develop a set of linearly scalable algorithms, including that for constructing directed acyclic graphs, for training profile hidden Markov models and for conducting alignment on such graphs. The astounding power of these algorithms is demonstrated by both significantly improved accuracy and tremendous acceleration of SAR-CoV-2 MSA by *three observed to five projected orders of magnitude* for genome set sizes ranging from 40,000 to more than 4 million when compared with widely utilized MAFFT. Future application to other viral species and extension to more complex genomes will prove this algorithm set as a cornerstone for the coming era of ultramassive genome sets.

Introduction

The SARS-CoV-2 pandemic brought us a genome dataset amount to nearly 20 million (with more than 4 million high quality sequences). This provides a potential opportunity for us to in-

vestigate evolution of a virus with unprecedented resolution (*1*). With ever increasing efficiency and improving accuracy of sequencing, more ultramassive genome sets are likely to be generated. For example, the number of publicly available human genomes is approaching 1 million (https://www.ukbiobank.ac.uk, https://allofus.nih.gov), and is projected to reach half billion in 2030 (https://www.ga4gh.org/news_item/ga4gh-releases-2018-strategic-roadmap/). Somatic mutation studies are expected to generate large sets of single cell genomes for many individuals (*2*), and extension of such sequencing investigations to more important species is warranted. Therefore we are inevitably entering an upcoming era of ultramassive genome sets. To reliably extract valuable information on evolution and molecular interactions buried in genome diversity and correlations among different segments there in, it is essential to perform accurate multiple sequence alignment (MSA), which is unfortunately computationally intractable for ultramassive data sets with presently available algorithms.

The goal of MSA computation is to align residues of the same evolutional and/or (functional, structural) origin from different sequences to the same column as illustrated in Figure 1(a,b). As we never have a chance to find out the real evolutional path of biological sequences, the most widely utilized analysis strategy is to find the maximum likelihood solution as an effective approximation, which is still intractable for hundreds of or more sequences as its computational cost scales exponentially with the number of sequences (n). There has been a long history of approximate MSA algorithm development (*3–9*), which are based on dynamic programming (*4, 8*), hidden Markov models (HMMs) (*3, 10–12*), or some form of their mixtures (*13, 14*). Presently, consistency based algorithms (*15*) are considered to be sufficiently accurate for practical purposes, it has a computational complexity of $\sim O(n^3L^2)$ (Simultaneous alignment of p sequences results in $\sim n^p$ complexity), which amount to at least 3.7×10^{30} float-point operations for the particular case of more than 4 million SARS-CoV-2 genomes, and takes more than $1,000$ years to complete on a exa-scale supercomputer. MAFFT (*16*) is widely utilized (*17–21*) for its much higher efficiency and acceptable accuracy. Regressive alignment (*22*) utilizes present mainstream methods (e.g. MAFFT) with a dividing and conquer strategy to speed up MSA calculation. Recent FM-index alignment made some progress by removing a small portion of tremendous repetitive computation (*23*), which is ubiquitous for all present algorithms when applied on highly similar sequences (e.g. SARS-CoV-2 or human genome sets). Unfortunately, performance of all above algorithms are far from practically

feasible for such ultramassive genome sets. Presently, the only routinely viable option is to use a selected reference sequence and align all other sequences against it (SR-pair), which is an extremely coarse treatment with relationship among all non-reference sequences ignored. Besides difficulty in obtaining accurate MSA for currently available sequences, we are in dire lack of efficient and succinct graph and probabilistic representations for downstream tasks of viral genome analysis. These tasks include but not limited to comparative genomics, rapid (joint) variant calling for large data sets, and monitoring and accurate phylogenetic characterization of continuously arising new genomes in real time.

We set to tackle these challenges by developing a set of new algorithms, assembly of which dramatically accelerate MSA by essentially removing all repetition of computation engendered by present alignment algorithms for similar sequences. Of special importance is the explicit utility of fragment topological order (FTO, illustrated in Figure 1(e,f)), which while being implicit in all present algorithms, plays a pivotal role in our construction of directed acyclic graphs (DAG). Correspondingly, graphs constructed in this work are termed FTO-DAGs. In strong contrast to usual tradeoff between accuracy and efficiency, our strategy achieves significantly improved accuracy besides sharply increased efficiency. In the particular case of SARS-CoV-2 genomes, their MSA is accelerated by *three observed and five projected orders of magnitude* for genome set sizes ranging from $40,000$ to more than 4 million when compared with the fastest option of MAFFT. These algorithms therefore lay an important cornerstone for the upcoming era of ultramassive genome sets by transforming similar tasks *from intractable to routine*. The overall pipeline of the MSA calculation for a ultramassive genome set with this algorithm set is illustrated in Figure 2.

Results

Construction of directed acyclic graphs

Subgraph construction and graph merging

There are two potential strategies to remove repetitions in MSA computation. One is to cache (i.e. to catalog results for distinct blocks of alignment computation and directly use them when the corresponding repetitions occur), the other is to merge identical segments in all sequences before

performing alignment. However, due to the fact that common segments shared by distinct groups of sequences are different and overlapping, both strategies are difficult to implement as one need to face tough decisions on selecting segments of caching/merging in earlier stages of computation without knowing what comes later on. Additionally, the cache approach takes excessively large memory space. de Bruijin graph has been widely utilized in genome assembly (*24*), and may effectively compress identical parts from different genomes (*25*). Unfortunately, cycles in such graphs prevent effective alignment and their removal is computationally challenging for graphs housing large number of sequences. Based on the above consideration, we decided to develop a series of algorithms and procedures for potential cycle detection and prevention (PCDP) for each sequence being added, so as no cycle will ever form in the first place for a target FTO-DAG. This turns out to be an effective decision. The overall pipeline is illustrated in Figure 3b. More specifically, an incoming sequence is first transformed into a sequence path (as shown in Fig. 3(a)) before its merging into a target FTO-DAG. Mergeable nodes from the target FTO-DAG and the incoming sequence path, as defined by identity of enclosed sequence fragment, are denoted as initial anchor and matching set respectively (Figure 3(c)). For bijective initial anchor and matching sets, formation of potential cycle occurred if their respective FTO were in conflict and subsequent merging directly conducted (Figure 3(c,d)). Final anchor and matching sets after potential cycle prevention are illustrated in Figure 3(e), and the new merged FTO-DAG is shown in Figure 3(f). When these two sets are not bijective, direct merging of matching node pairs would cause formation of potential cycles and violation of FTO, a series of procedures are developed to prevent such violation in such scenarios. FTO in both directions, including that from 5' to 3' and that from 3' to 5' of a genome sequence, are utilized in our algorithm. To realize parallel computation, the target dataset is split into subsets (see Figure 2), sequences within each subset are utilized to construct a corresponding subgraph, which are subsequently merged recursively to form the total graph.

The sequence fragment length L_{frag} and its adjustment

The sequence fragment length L_{frag} is the only adjustable parameter in the FTO-DAG construction process, with its upper and lower limit being the minimal sequence length L_{min} in a given sequence set and 1 respectively. If $L_{frag} = L_{min}$, there would be negligible fragment repetition for distinct sequences and the FTO-DAG constructing process is essentially useless as no significant compu-

tation repetition will be removed. On the other hand, when $L_{frag} = 1$, there would be numerous occurrences for each single-character "fragment". This will lead to non-bijective initial anchor (in target FTO-DAG) and matching (in a incoming sequence path) sets housing a large number of potential cycles. Consequently the computation engendered by PCDP becomes highly expensive. So excessively small fragment lengths may result in prohibitive computational cost of PCDP while too large fragment lengths may significantly decrease the repetition reduction. To avoid such dilemma caused by a fixed value of L_{frag}, a recursive reduction procedure is developed. Specifically, a sufficiently large L_{frag} is selected first to reduce PCDP computation at the global genome level, such that relative positions of different genomes are roughly determined, the resulting FTO-DAG may be further compressed locally recursively by stepwise decrease of L_{frag} so as no overwhelming PCDP ever occurs. The extreme case is to reduce L_{frag} to 1, which is termed atomization of FTO-DAG. The extent of repetition reduction for a given L_{frag} may be quantitatively described by the compression ratio η:

$$\eta = \frac{\text{number of nodes in individual sequence paths}}{\text{number of nodes in the corresponding FTO-DAG}} \tag{1}$$

Profile HMM training and alignment on FTO-DAG

Formulation and parallelization of profile HMM on FTO-DAG

All present MSA algorithms were developed for aligning individual sequences and/or profiles. To take advantage of tremendous compression ratio of FTO-DAGs constructed above, development of MSA algorithm on which becomes essential. Highly complex graph structures engender great difficulty for dynamic programming procedures. Fortunately, the local focus of profile HMM (pHMM) algorithm (*26*) is a great tool for overcoming global structural complexity of FTO-DAGs. Besides, probabilistic representations of pHMM provide great convenience for downstream analyses. We therefore derive the formulation of pHMM on FTO-DAG, develop a graph traversal based parallel strategy, and name the algorithms as DAG-pHMM as schematically demonstrated in Figure 4.

pHMM training windows and parameter initialization

A FTO-DAG is essentially a partial alignment. In addition to accomplishing near-complete repetition riddance, it accelerates and improves pHMM training by providing valuable information for parameter initialization and tremendously reducing computation/memory requirement. In the original pHMM (*26, 27*), the computational complexity for a single training iteration for each sequence of length L is $O(L_{pHMM}L)$, with L_{pHMM} being the number of match columns in the pHMM. In practice, L_{pHMM} is usually larger than the average of L (here we note as $\bar{L}$), hence the computational complexity is at least of $O(\bar{L}^2)$. Based on the relative position property of FTO-DAG, each node i has a potential range of match positions in the model with window length w_i, which is determined by sequence similarity and extent of conservation, and is significantly smaller than and independent of L_{pHMM}. Utilizing this property, we successfully reduce the computational complexity of pHMM to $O(L_{pHMM}\bar{w})$, with $\bar{w}$ being the weighted average pHMM window sizes w_i. In the particular case of SARS-Cov-2 genome sets, $\bar{w}$ (plus an additional safety window extension) is about 250, which (considering that $\bar{L}$ is greater than 30000) decreases both computation and memory requirement by more than 120 times without causing any notable reduction of accuracy.

Viterbi algorithms on FTO-DAG

Viterbi alignment algorithm is significantly more complicated for a FTO-DAG constructed from large datasets than for a single sequence. The reason is that a sequence fragment within a single node of a FTO-DAG may be shared by different genomes, each of which at the position of the shared sequence may correspond to different hidden states in pHMM, bookkeeping of sequence originating genomes and hidden states in such nodes becomes more challenging with increasing graph complexity and sizes, and the accompanying cost may eventually overtake savings from increased compression ratio. Therefore, in reality we chose to carry out Viterbi alignment on subgraphs to achieve maximum efficiency.

Alignment accuracy and efficiency

To assess the accuracy of our algorithms, we utilized two group of SARS-CoV-2 genome sets sampled with significantly different time intervals as shown in Figure 5a. These data sets are

aligned with three types of methods, including the most coarse approximation of SR-pair, the most widely utilized MAFFT with "auto" option (the most accurate option is unrealistic to test for more than a few hundred sequences), and our DAG-pHMM. To assess efficiency, we compared among MAFFT, SR-pair and DAG-pHMM for datasets ranging from 50 to 40000 in sizes. For datasets larger than $40,000$, CPU time for MAFFT may only be extrapolated as direct computation becomes extremely time consuming. As shown in Fig. 5b, our method achieves acceleration of three observed and five projected orders of magnitude for data sets ranging from 40000 to more than 4 million. When data set sizes is greater than 40000, our DAG-pHMM surpasses the most coarse SR-pair approximation. We did not try regular pHMM (*26*) as it was demonstrated that a mere alignment of 1000 genomes by a trained model took $\sim$ 50minutes (*28*) on a cluster (detailed parameters were not reported). As the training is the most expensive part, so regular pHMM is not feasible for ultramassive genome set.

As shown in Figure 5(c-j), except in the smallest dataset of 50 and 100 genome sequences for genome set sampled within $1-$day sampling window (Figure 5(g-h)), our algorithm attains consistently higher accuracy as measured by SP scores and entropy (see Fig.1(b,c,d) for definition). Despite tremendously more computational cost, for the large LSW group (800 days), MAFFT exhibits consistently inferior SP scores (see Figure 5(c)) and higher entropy (see Figure 5(d)) than SR_pair for 50-sequence small sets, and similarly inferior SP scores (see Figure 5(e)) but comparable entropy (see Figure 5(f)). The error bars in these figures are variances among different datasets calculated separately for positive and negative values with respect to the corresponding averages, their large magnitude it caused by significant differences among different datasets. More specifically, while the conserved part of the genome is essentially very similar for all genomes, the variable regions are very different with distinct genome sets.

???

Discussions

It is important to note that our algorithms are targeted for similar sequences, not as a general substitute for many excellent present algorithms (e.g.MUSCLE, MAFFT, $\cdots$, regression). Nonetheless, it is certain that there will be more and more ultramassive sets of similar (genome) sequences

awaiting accurate MSA. Two key factors contributing to the observed dramatic MSA accelerations are the FTO-DAG compression ratio (η) and average window sizes ($\bar{w}$) in pHMM training, with a larger η and a smaller $\bar{w}$ corresponds higher efficiency. Both factors depend critically on the extent of sequence similarity within the target data set, with η decreases and $\bar{w}$ increases with reduction of sequence similarity.

Regarding accuracy, due to the fact that training of DAG-pHMM is an optimization process in high dimensional space of its parameters, more attempts are likely to locate for better local maximum of likelihood and therefore higher accuracy. As cost of each locally converged training is not expensive (a few minutes on a single 128-core node for 4-million genome set), higher accuracy will be obtained by simply conducting training from more parameter starting points with a little more computing budget. Strategies of achieving highest accuracy with given computing budget for such ultramassive genome sets will be investigated further in the future.

The major output of our algorithms are FTO-DAGs, pHMM models and MSAs for selected datasets. While both FTO-DAGs and pHMMs are constructed mainly to achieve accurate MSA, they both have additional important utilities. As ultramassive genome sets being generated for more and more important species, identification of the closest genome for a new sequence becomes more difficult, and comparative genomics questions can not be tackled by naive alignment of single sequences anymore. FTO-DAGs generated by our algorithms may be utilized for accurate and rapid identification of the closest previous genome of continuously arising new ones, and for variant calling or other similar tasks. Probabilistic models such as pHMMs are suitable tools to achieve significantly more comprehensive comparison between different species. Accurate MSAs generated for large genome sets in this work may be utilized to perform better evolution analysis (e.g. phylogenetic and recombination investigations), and to carry out genome segments correlation analysis, which may reveal difficult-to-discover molecular interactions and disclose presently unknown encoding segments and regulation pathways. When extended to more complex genomes (e.g. yeast, mouse, human), FTO-DAGs and MSAs together may serve as a reliable framework and high resolution map for integrating information from other types of sequencing data (e.g. RNA-seq, epigenetic modifications, single cell data, spatially resolved transcriptomics). With ever increasing quantity and improving quality of sequence data, which are loyal recordings of evolution experiments in hundreds of millions years, language models () will certainly player more and more important roles

in our future investigation of biology. Accurate MSAs and genome segments correlations derived from which may provide important step stones for training and aligning future language models.

Our FTO-DAGs provide an alternative to current pan-genome graphs with cycles (*29–32*). Besides the demonstrated advantage of facilitating lightning fast alignment, the absence of cycles is also helpful in rapid sifting of virtual paths formed by stretches from distinct genomes.Both types of graphs have respective advantages and limitations, we may select either or both in our analysis based on specific goals and data sets.

Conclusion

In summary, we developed a set of new algorithms, including that for construction of FTO-DAGs, for training of pHMMs and subsequent Viterbi alignment on such graphs. These algorithmic and implementational efforts jointly realized acceleration of accurate MSA of more than 4 million SARS-CoV-2 genomes by an estimated five orders of magnitude, thus transforming similar tasks from intractable to routine. The DAGs, pHMM models and alignments generated for many data sets in this study may be utilized by the scientific community for further analysis of evolution and for discovering of molecular interactions. With this set of algorithms, handling of continuously rising variants becomes computationally trivial in real time despite current ultramassive genome sets. Application of these algorithms to other viral genome sets is straight froward. We are actively work on extension of these algorithms to more complex genomes of bacteria, human and more species. Completion of the present and planned future work are expected to lay an important cornerstone and open up new frontiers for the upcoming era of ultramassive genome sets.

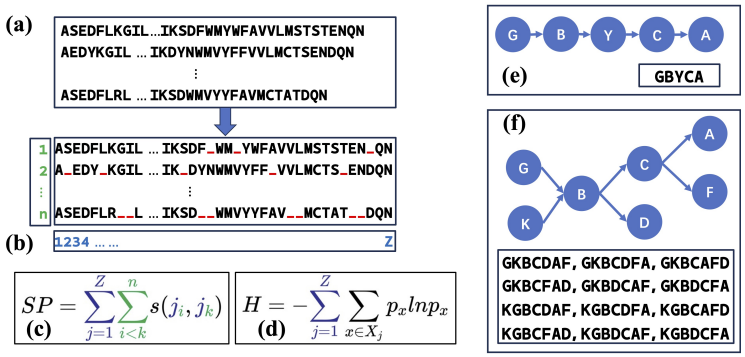

Figure 1: **Schematic illustration of MSA and topological order.** (a) n unaligned sequences. (b) n sequences after alignment with Z columns. The alignment process is to find most suitable position combination to place gaps (represented by red underscore in (b)) such that resulting alignment scoring the best. Considering the astonishing number of combinations of gaps and residues within each and among different sequences, the challenges are apparent (see ???). The quality of MSA is usually characterized by (c) sum of pair score (SP) (with $s(j_i, j_k)$ being the score of the i and the kth sequence at the jth column) and/or (d) entropy (H) (with X_j being the number of different residues in the jth column) as shown in the two equations, with larger SP and smaller H corresponding to better MSA. One critical property of MSA is that after alignment, the topological order for all segments/residues in each sequence remain the same as illustrated in (e) for a linear chain of nodes, topological order is the unique order of appearance specified by the directed edges as shown in the text box, and in (f) for a directed graph of nodes, topological order dictates that a parent node has to precede its child nodes, as each node may have multiple parent/child nodes, so topological order is not unique as shown in the text box.

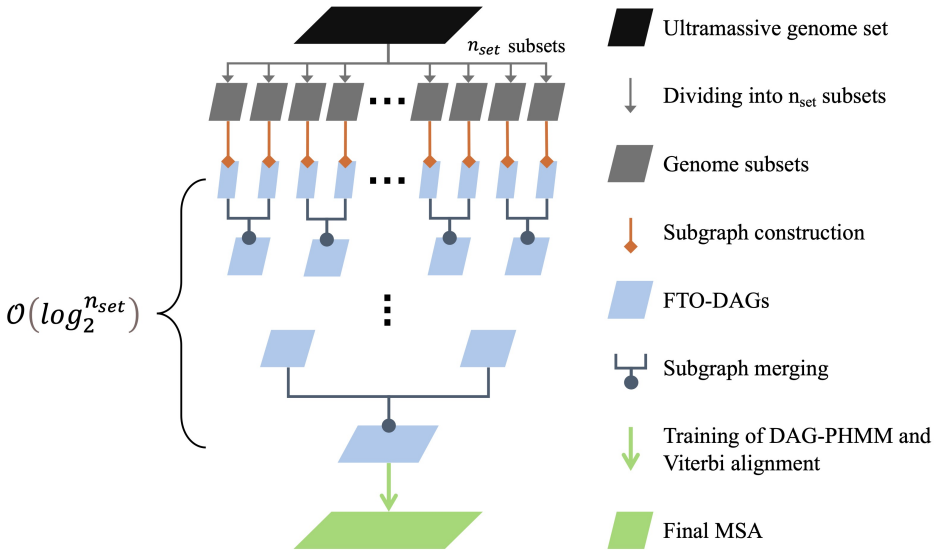

Figure 2: **Schematic illustration of the overall algorithm pipeline for MSA of a ultramassive genome set.** Shapes represent data, including genome sets, graphs and MSA. Lines represent algorithms and their corresponding manipulation of relevant data.

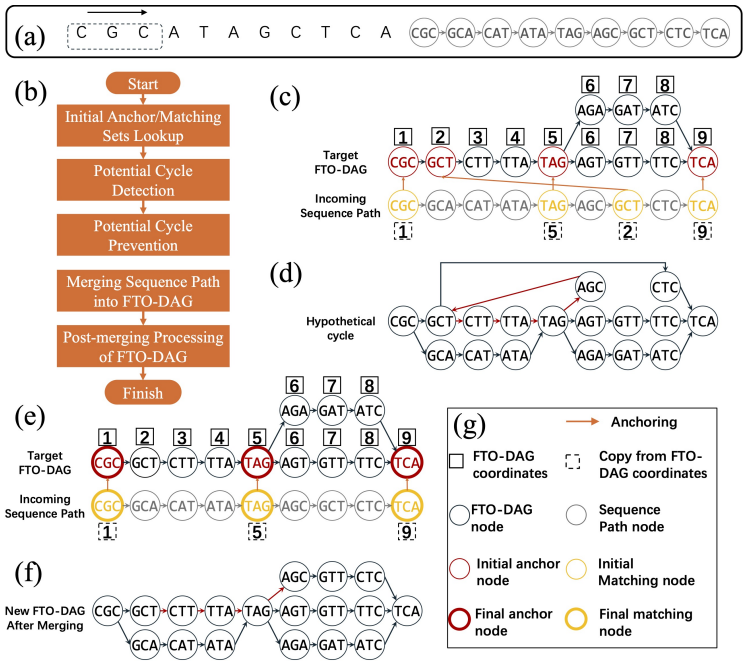

Figure 3: **The overall pipeline and one key scenario of potential cycle detection and prevention.** (a) Conversion of a sequence into a sequence path by sliding a window of length $L_{frag} = 3$. (b)The overall pipeline of FTO-DAG subgraph constraction. (c) The initial anchor set within the target FTO-DAG indicated in red and the initial matching set in the incoming sequence path indicated in yellow. It is apparent that there is FTO conflicts between these two sets, embodied as the copy of FTO coordinates from the initial anchor set to the initial matching set in the incoming sequence path being $1 \Rightarrow 5 \Rightarrow 2 \Rightarrow 9$, which is not in strict increasing order as dictated by FTO. (d) A hypothetical merged graph with a cycle indicated red directed edges, such cycle would have been formed if the incoming sequence path was directly merged into the target FTO-DAG based on the initial anchor/matching sets. (e) Final anchor and matching set indicated by thicker circles in red and yellow respectively, this is obtained from a potential cycle prevention procedure, which removes "GCT" from initial anchor/matching sets with a dynamic programming procedure. (f) The new FTO-DAG after assimilating the incoming sequence path. (g) annotation of shapes in (c),(d),(e),(f)

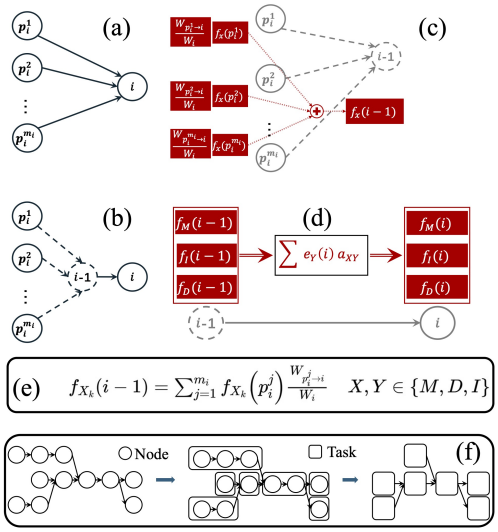

Figure 4: **Schematic illustration of parallelization and forward probability calculation for nodes with multiple parent nodes in pHMM training on a FTO-DAG. Backward calculation for nodes with multiple children are calculated similarly (not shown for brevity).** (a) A node i with m_i parent nodes in a FTO-DAG. (b) In forward probability calculation of regular pHMM (), a node i has a preceding node $(i-1)$, we therefore creating a virtual preceding node $(i-1)$ shown in a dashed circle, corresponding probability terms of this virtual node are constructed as shown in (c) and (e). (d) With a virtual node $(i-1)$, calculation of probability terms of node i is calculated by multiplication of transition probability matrix and emission probability as in a regular pHMM. (f) Parallelization scheme of pHMM training on FTO-DAG, linear path segments are stored as separate computing tasks by a graph traversal, dependence among these tasks are controlled by their topological order (See Fig. 1(f) for illustration.

(a)

NS	50	100	500	1000	5000	10000	20000	40000
ND	32	32	32	32	32	16	8	4
LSW(Large)	800	800	800	800	800	800	800	800
LSW(Small)	1	1	1	1	1	100	200	300

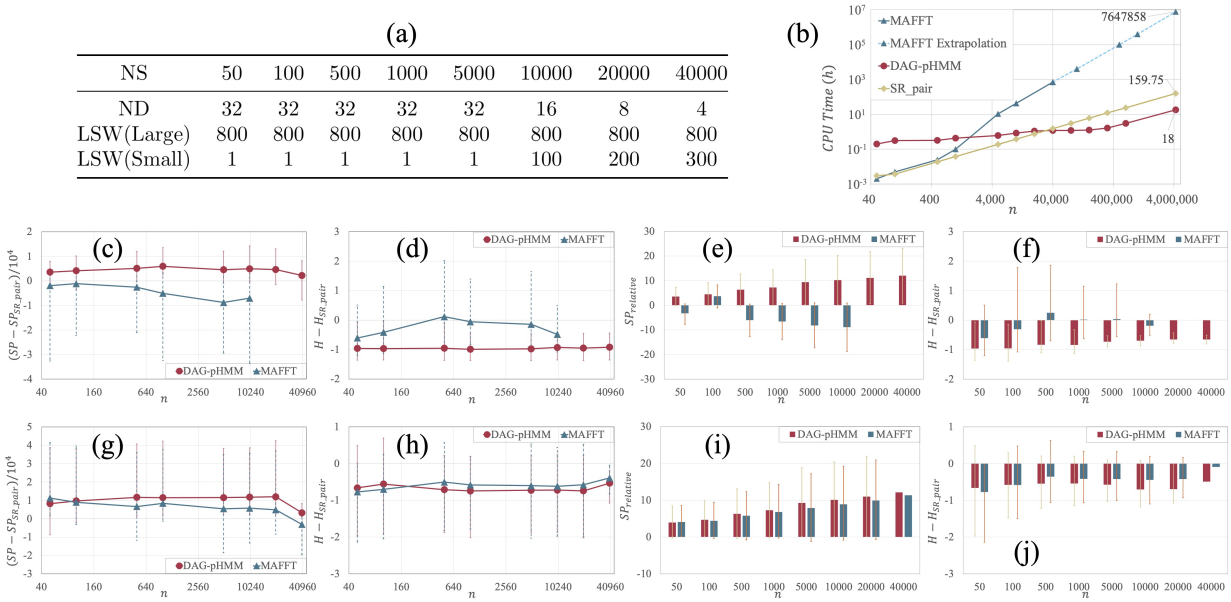

Figure 5: **Efficiency and accuracy of MAFFT and DAG-pHMM based alignment on selected datasets in comparison with the SR-pair approximation.** (a) Basic information on the two groups of datasets with different length of sampling windows (LSW, in unit of days), LSW (large) and LSW (small), termed below as LSWL and LSWS groups respectively. NS: Number of sequences, ND: Number of genome datasets. (b) Average computing time (in hours) on a single node (dual CPU, 128 cores at 2.90GHz with 500G memory) for calculating MSA for different sizes of genome sets by MAFFT, SR_pair and DAG-pHMM (variances are smaller than symbol sizes). (c) and (g): SP scores relative to that of SR_pair results for 50-sequence genome sets in the LSWL and LSWS groups by themselves or when embedded in variously sized larger genome sets shown by horizontal axis. (d) and (h): Entropy relative to that of SR_pair results for the 50-sequence genome sets in the LSWL and LSWS groups by themselves or when embedded in variously sized larger genome sets shown by horizontal axis. (e) and (i): SP scores relative to that of SR_pair results for the LSWL and LSWS groups for variously sized genome sets, $SP_{relative} = sign(SP - SP_pair)log|SP - SP_pair|$. (f) and (j): Entropy relative to that of SR_pair results for the LSWL and LSWS groups for variously sized genome sets.

References and Notes

1. P. V. Markov, *et al.*, The evolution of SARS-CoV-2. *Nature Reviews Microbiology* **21** (June), 361–379 (2023), doi:10.1038/s41579-023-00878-2.

2. K. Walsh, NIH project probes the human body's multitude of genomes. *Science* **381** (6659), 719–720 (2023), doi:10.1126/science.add3503.

3. R. Durbin, S. R. Eddy, A. Krogh, G. Mitchison, *Biological sequence analysis, Probabilistic models of proteins and nucleic acids* (Cambridge University Press, Cambridge, UK) (1998).

4. D. W. Mount, Using Progressive Methods for Global Multiple Sequence Alignment. *Cold Spring Harb Protoc* **4** (7), 1–7 (2009), doi:10.1101/pdb.top43.

5. R. C. Edgar, S. Batzoglou, Multiple sequence alignment. *Current opinion in structural biology* **16**, 368–373 (2006), doi:10.1016/j.sbi.2006.04.004.

6. C. Kemena, C. Notredame, Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics* **25** (19), 2455–2465 (2009), doi:10.1093/bioinformatics/btp452.

7. M. Chatzou, *et al.*, Multiple sequence alignment modeling : methods and applications. *Briefings in Bioinformatics* **17** (6), 1009–1023 (2016), doi:10.1093/bib/bbv099.

8. D. W. Mount, Using Iterative Methods for Global Multiple Sequence Alignment. *Cold Spring Harb Protoc* **4** (7), 1–6 (2009), doi:10.1101/pdb.top44.

9. Y. Zhang, Q. Zhang, J. Zhou, Q. Zou, A survey on the algorithm and development of multiple sequence alignment. *Briefings in Bioinformatics* **23**, 1–16 (2022).

10. R. D. Finn, J. Clements, S. R. Eddy, HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research* **39** (suppl_2), W29–W37 (2011), doi:10.1093/nar/gkr367, `https://doi.org/10.1093/nar/gkr367`.

11. D. W. Mount, D. W. Mount, Using Hidden Markov Models to Align Multiple Sequences Using Hidden Markov Models to Align Multiple Sequences. *Cold Spring Harb Protoc* **4** (7), 1–8 (2009), doi:10.1101/pdb.top41.

12. Y. Liu, B. Schmidt, D. L. Maskell, MSAProbs : multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics* **26** (16), 1958–1964 (2010), doi:10.1093/bioinformatics/btq338.

13. C. B. Do, M. S. P. Mahabhashyam, M. Brudno, S. Batzoglou, ProbCons : Probabilistic consistency-based multiple sequence alignment. *Genome Research* **15**, 330–340 (2005), doi: 10.1101/gr.2821705.1994.

14. U. Roshan, D. R. Livesay, Probalign : multiple sequence alignment using partition function posterior probabilities. *Bioinformatics* **22** (22), 2715–2721 (2006), doi:10.1093/bioinformatics/btl472.

15. T. Ridgeway, L. Nw, T-Coffee : A Novel Method for Fast and Accurate Multiple Sequence Alignment. *Journal of molecular biology* **302**, 205–217 (2000), doi:10.1006/jmbi.2000.4042.

16. K. Katoh, K. Misawa, K.-i. Kuma, T. Miyata, MAFFT : a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research* **30** (14), 3059–3066 (2002).

17. H.-l. Zeng, genomes reveals epistasis between eight viral genes. *Proceedings of the National Academy of Sciences* **117** (December), 31519–31526 (2020), doi:10.1073/pnas.2012331117.

18. A. B. Gussow, *et al.*, Genomic determinants of pathogenicity in SARS-CoV-2 and other human coronaviruses. *Proceedings of the National Academy of Sciences* **117** (26), 15193–15199 (2020), doi:10.1073/pnas.2008176117, `https://www.pnas.org/doi/abs/10.1073/pnas.2008176117`.

19. H.-y. Hu, W.-w. Zhou, Intercontinental transmission and local demographic expansion of SARS-CoV-2. *Epidemiology and Infection* **149** (e94), 1–5 (2021).

20. D. Mtetwa, T. Manjengwa, Z. Chikwambi, Genetic diversity and spread dynamics of SARS-CoV-2 variants present in African populations. *Royal Society Open Science* **11** (6), 231326 (2024), doi:10.1098/rsos.231326, `https://royalsocietypublishing.org/doi/abs/10.1098/rsos.231326`.

21. M. P. Khurana, *et al.*, High-resolution epidemiological landscape from ~290,000 SARS-CoV-2 genomes from Denmark. *Nature Communications* **15** (1), 7123 (2024), doi:10.1038/s41467-024-51371-0, https://doi.org/10.1038/s41467-024-51371-0.

22. E. Garriga, *et al.*, Large multiple sequence alignments with a root-to-leaf regressive method. *Nature Biotechnology* **37** (December), 1466–1470 (2019), doi:10.1038/s41587-019-0333-6, http://dx.doi.org/10.1038/s41587-019-0333-6.

23. L. Huan, Z. Quan, X. Yun, A novel fast multiple nucleotide sequence alignment method based on FM-index. *Briefings in Bioinformatics* **23**, 1–9 (2022).

24. P. E. C. Compeau, P. A. Pevzner, G. Tesler, primer How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology* **29** (11), 987–991 (2011), doi:10.1038/nbt.2023, http://dx.doi.org/10.1038/nbt.2023.

25. I. Minkin, S. Pham, P. Medvedev, Sequence analysis TwoPaCo : an efficient algorithm to build the compacted de Bruijn graph from many complete genomes. *Bioinformatics* **33** (24), 4024–4032 (2017), doi:10.1093/bioinformatics/btw609.

26. S. R. Eddy, Profile hidden Markov models. *Bioinformatics* **14**, 755–763 (1998).

27. F. Lemoine, L. Blassel, J. Voznica, O. Gascuel, COVID-Align: accurate online alignment of hCoV-19 genomes using a profile HMM. *Bioinformatics* **37** (12), 1761–1762 (2020), doi:10.1093/bioinformatics/btaa871, https://doi.org/10.1093/bioinformatics/btaa871.

28. F. Lemoine, L. Blassel, J. Voznica, O. Gascuel, COVID-Align: accurate online alignment of hCoV-19 genomes using a profile HMM. *Bioinformatics* **37** (12), 1761–1762 (2020), doi:10.1093/bioinformatics/btaa871, https://doi.org/10.1093/bioinformatics/btaa871.

29. A. Guarracino, E. Garrison, S. Heumos, S. Nahnsen, Genome analysis ODGI : understanding pangenome graphs. *Bioinformatics* **38** (May), 3319–3326 (2022).

30. J. M. Eizenga, *et al.*, Pangenome Graphs. *Annual Review of Genomics and Human Genetics* **21** (Volume 21, 2020), 139–162 (2020), doi:https://doi.org/

10.1146/annurev-genom-120219-080406, https://www.annualreviews.org/content/journals/10.1146/annurev-genom-120219-080406.

31. W.-W. Liao, *et al.*, A draft human pangenome reference. *Nature* **617** (7960), 312–324 (2023), doi:10.1038/s41586-023-05896-x, https://doi.org/10.1038/s41586-023-05896-x.

32. Y. Gao, *et al.*, A pangenome reference of 36 Chinese populations. *Nature* **619** (7968), 112–121 (2023), doi:10.1038/s41586-023-06173-7, https://doi.org/10.1038/s41586-023-06173-7.

Acknowledgments

We thank professors Hao Ge at Peiking University, Yang Ji at Zhejiang University and Hao Wu at Wenzhou Institute University of Chinese Academy of Sciences for enlightening discussions in preparation of the manuscript.

Funding: This study is partially supported by the Interdisciplinary Integration and Innovation Project of JLU with grant number JLUXKJC2021ZZ05

Author contributions: X.L. prepared SARS-CoV-2 data sets, developed and implemented the DAG construction algorithms. X.L. and H.L. jointly performed development and implementation of graph data structure, profile HMM training and viterbi alignment on graphs. X.L and H.L participated in the manuscript preparation. P.T. conceived the study, designed the algorithm framework (FTO guided construction of DAG, followed by pHMM training and alignment on such graphs), and wrote the paper.

Competing interests: Patents application are filed for algorithms developed in this work.

Data and materials availability: Organization of prepared datasets, FTO-DAGs, trained pHMMs and MSAs obtained in this study, and executables of programs are undergoing and will be made publicly available upon formal publication of this work.