




CMAPLE: Efficient Phylogenetic Inference in the Pandemic Era

Nhan Ly-Trong ¹, Chris Bielow,² Nicola De Maio ³, and Bui Quang Minh ^{1,*}

¹School of Computing, College of Engineering, Computing and Cybernetics, Australian National University, Canberra, ACT 2600, Australia

²Bioinformatics Solution Center, Freie Universität Berlin, 14195 Berlin, Germany

³European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Hinxton, UK

*Corresponding author: E-mail: m.bui@anu.edu.au.

Associate editor: Andrey Rzhetsky

Abstract

We have recently introduced MAPLE (MAXimum Parsimonious Likelihood Estimation), a new pandemic-scale phylogenetic inference method exclusively designed for genomic epidemiology. In response to the need for enhancing MAPLE's performance and scalability, here we present two key components: (i) CMAPLE software, a highly optimized C++ reimplementations of MAPLE with many new features and advancements, and (ii) CMAPLE library, a suite of application programming interfaces to facilitate the integration of the CMAPLE algorithm into existing phylogenetic inference packages. Notably, we have successfully integrated CMAPLE into the widely used IQ-TREE 2 software, enabling its rapid adoption in the scientific community. These advancements serve as a vital step toward better preparedness for future pandemics, offering researchers powerful tools for large-scale pathogen genomic analysis.

Key words: phylogenetics, phylogenomics, epidemiology, maximum likelihood, models of sequence evolution.

Introduction

Phylogenetic analysis plays a vital role in genomic epidemiology, as exemplified during the COVID-19 pandemic (Gonzalez-Reiche et al. 2020; Lu et al. 2020; Hodcroft et al. 2021; Vöhrringer et al. 2021). Phylogenetic tools help unveil the origins and transmission of pathogens, monitor the emergence of new variants, and inform vaccine development. For instance, the widely used IQ-TREE 2 software (Minh et al. 2020) has been at the core of the COVID-19 pandemic response, being employed, for example, in Nextstrain (Hadfield et al. 2018). To deal with the pandemic, we recently introduced MAPLE (De Maio et al. 2023), a novel likelihood-based phylogenetic inference method tailored for genomic epidemiological analyses. To address the ever-looming threat of new pandemics, the need for further improvements regarding both the performance and scalability of MAPLE has become increasingly apparent.

Here, we present CMAPLE, a C++ reimplementations of MAPLE highly optimized for performance and scalability. CMAPLE is 3-fold faster and more memory efficient than MAPLE, reducing the runtime to analyze 200,000 SARS-CoV-2 sequences (McBroome et al. 2021) using one CPU core from 2.4 d to 19 h. Notably, CMAPLE can reconstruct a phylogenetic tree of 1 million SARS-CoV-2 genomes, taking 11 d and 15.4 GB RAM. Additionally, we incorporate a plethora of new protein models (Minh

et al. 2021; Dang et al. 2022) to improve CMAPLE's versatility in analyzing a broader spectrum of pathogen genomic data. We also developed a suite of application programming interfaces (APIs) and have successfully incorporated CMAPLE into IQ-TREE version 2.3.4.cmaple (<https://github.com/iqtree/iqtree2/releases>). Additionally, we developed an adaptive mechanism that automatically selects CMAPLE or IQ-TREE search algorithms to minimize the runtime. These advancements facilitate rapid dissemination and widespread adoption of CMAPLE.

To efficiently control pandemics, authorities need to make urgent decisions, such as applying new preventive measures to tackle new virus variants. CMAPLE enables a quicker and more accurate tracking of transmission and discovery of new variants and mutations, which is critical for public health decisions, vaccine designs, and better preparedness for future pandemics.

In the following, we highlight the improvements and key new features in CMAPLE and discuss a few potential directions for further enhancements.

The Overall Workflow of the CMAPLE Tree Search Algorithm

CMAPLE is a C++20 reimplementations of the MAPLE algorithm (v0.1.4) implemented in Python. MAPLE takes advantage of low sequence divergence in pathogen data to

Received: February 09, 2024. Revised: May 15, 2024. Accepted: June 21, 2024

© The Author(s) 2024. Published by Oxford University Press on behalf of Society for Molecular Biology and Evolution.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Open Access

optimize memory usage. More specifically, MAPLE compresses an input sequence alignment in FASTA or PHYLIP format into the so-called MAPLE format (De Maio et al. 2023), which represents each sequence by a set of the differences to a reference sequence. The reference sequence can be specified by users or automatically computed from the input alignment as the consensus sequence. This results in a very compact data structure, and we have taken a similar approach for the phylogenetic likelihood vectors, allowing much faster (although more approximate) likelihood calculation than the standard Felsenstein's (1973) pruning algorithm. Our new implementation of CMAPLE focuses on further reducing memory allocation operations, optimizing memory access patterns, and minimizing CPU cache misses.

The CMAPLE algorithm involves three main steps: (1) building an initial tree using the sample placement algorithm (see below), (2) optimizing the tree topology using SPR moves (Felsenstein 1989; Swofford and Olsen 1990), and finally (3) optimizing the branch lengths. Users can choose to skip steps 2 and 3 (-search fast) to speed up the analysis, in which case CMAPLE operates like USHER (Turakhia et al. 2021). CMAPLE also supports updating a user-provided input tree, which might not contain all taxa. In this case, the first step of CMAPLE will add the missing sequences to the existing tree using the placement algorithm, and then the second step only applies SPR moves to the newly added sequences. Users can choose to skip it (-search fast) or to perform a thorough search considering all SPR moves (-search exhaustive).

CMAPLE Implementation and Performance Optimization

In general, high-performance computing and optimization is a very broad topic with multiple factors such as hardware capabilities, programming language, compiler version, choice of algorithm, and data structure, to name a few. We applied many code optimization techniques to enhance the time and memory efficiency of CMAPLE. In brief, while implementing the MAPLE algorithm, we tried to leverage optimization techniques such as SIMD for matrix and vector operations, compact data structures (for tree nodes) with minimal padding, and optimal cache-line efficiency (hot/cold splitting), using bit fields and devirtualization where possible. We also tried to minimize branches, especially in hot loops, to use stack objects instead of heap objects, preallocations, and C++ move semantics. It is hard to quantify the individual effect of these techniques since there is interplay between them, yet they all contributed significantly to performance. Last but not least, we employ a high-performance library for memory allocation, jemalloc, to further save runtimes for Linux and macOS. As jemalloc can be activated during runtime, its effect can be quantified more easily and leads to a time saving of up to around 8% (supplementary table S2, Supplementary Material online).

CMAPLE relies on three third-party libraries: ncl (Lewis 2003), simd (<https://github.com/simd-everywhere/simd>), and zlib (<http://zlib.net/>).

Fast (Online) Sample Placement onto an Existing Tree

During the COVID-19 pandemic, many new viral genome sequences have been obtained and shared on a daily basis. Researchers and healthcare professionals have leveraged fast sample placement onto continually updated and maintained global SARS-CoV-2 phylogenetic trees for identifying new variants, recombinations, and reconstruction transmissions. Therefore, CMAPLE implements a fast sample placement algorithm that allows adding new sequences into a phylogenetic tree of existing samples as follows.

CMAPLE applies the stepwise addition method (Swofford et al. 1996), which iteratively adds new samples to the existing tree one at a time according to their similarity to the reference sequence, such that the most closely related sequence will be added first. For each new sequence, CMAPLE tries to insert it into every branch of the tree and reevaluates the tree's likelihood. The branch with the highest likelihood will be chosen. This process is repeated until all new sequences are added to the tree. In our tests, CMAPLE takes 14 min to insert 10,000 randomly sampled SARS-COV-2 sequences into an existing 500,000-sample tree. The runtime does not depend on the genome size, but on the divergence level of the added sequences to the reference: a higher divergence level leads to a longer running time.

Reversible and Nonreversible DNA and Protein Substitution Models

CMAPLE supports two reversible DNA substitution models, JC (Jukes and Cantor 1969) and GTR (Tavaré 1986), and the general nonreversible model UNREST (Yang 1994a). Additionally, we have implemented 40 empirical reversible and nonreversible protein models from the IQ-TREE 2 software (Minh et al. 2021; Dang et al. 2022), which are all listed at <https://github.com/iqtree/cmapple/wiki#supported-substitution-models>. Those models enable CMAPLE to analyze a broader spectrum of pathogen data, including bacterial genomes (Parks et al. 2018).

Fast Branch Tests

Phylogenetic inference typically involves branch support assessment of the inferred trees. To facilitate that task, CMAPLE incorporates the Shimodaira–Hasegawa-like approximate likelihood ratio test (SH-aLRT; Guindon et al. 2010). To take advantage of multicore CPUs, the SH-aLRT implementation is parallelized using OpenMP (Chapman et al. 2007), which speeds up the SH-aLRT calculation nearly linearly with the number of CPU cores

used. For instance, assessing branch support for a phylogenetic tree with 100,000 tips requires 5.1 h on a single core (AMD EPYC 7551) but only 14 min on 32 cores, a 22-fold speedup.

API and Integration into the IQ-TREE 2 Software

In addition to a standalone software package, we provide a C++ API with comprehensive documentation at <http://iqtree.org/cmaple/>. The CMAPLE API provides three main C++ classes: Alignment, Model, and Tree, representing the input sequence alignment, the substitution models, and the phylogenetic tree, respectively. The API facilitates the integration of the CMAPLE's algorithm into existing phylogenetic software such as IQ-TREE (Nguyen et al. 2015), RAxML (Stamatakis 2014; Kozlov et al. 2019), and PHYML (Guindon et al. 2010). In fact, we already incorporated CMAPLE into IQ-TREE version 2.3.4.cmaple (<https://github.com/iqtree/iqtree2/releases>), which users can use via "--pathogen-force" option.

Automatically Selecting CMAPLE or IQ-TREE Search Algorithms to Minimize the Runtime

The performance of CMAPLE and MAPLE strongly relies on low sequence divergence (De Maio et al. 2023), i.e. the MAPLE algorithm only works well on closely related sequences, such as SARS-CoV-2 genomes. Therefore, we provide a feature to decide if the CMAPLE algorithm is effective for any user-given input alignment: (i) by default, every sequence must be at most 6.7% different from the reference sequence and (ii) the average sequence divergence from the reference must be at most 2%. We set these criteria based on the results of De Maio et al. (2023), who benchmarked MAPLE against popular phylogenetic methods using alignments simulated with different levels of divergence; at levels of divergence around 20 times higher than typical SARS-CoV-2 data sets, they found that MAPLE becomes less accurate or less efficient than traditional approaches, so we set these as thresholds in our method.

Within IQ-TREE version 2.3.4.cmaple, users can use the option "--pathogen," which will automatically invoke either the CMAPLE or the original IQ-TREE search algorithm, depending on the effectiveness defined above. For the API, developers can use the function `cmaple::isEffective()`.

Benchmarking CMAPLE against Existing Software

We benchmarked the sequential version of CMAPLE (under the default setting and using the jemalloc library) against MAPLE on a server with an AMD EPYC 7551 32-core Processor. To generate a testing data set, we sub-sampled 5K, 10K, 50K, 100K, and 200K real SARS-CoV-2 sequences from a global data set of 4.3 million genomes

available on 2022 April 2 (McBroome et al. 2021). CMAPLE is about three times faster (Fig. 1a) and requires over three times less memory than MAPLE (Fig. 1b) while yielding equally high likelihood trees as MAPLE (see the Software Validation section). This is due to the memory efficiency of C++ code over Python and several optimization techniques (see the CMAPLE Implementation and Performance Optimization section). MAPLE required 2.4 d and 11.5 GB RAM (Fig. 1) to reconstruct a tree of 200,000 SARS-CoV-2 sequences, whereas CMAPLE took only 19 h and 3.6 GB RAM.

Thanks to these improvements, we ran CMAPLE on an alignment of 1 million SARS-CoV-2 genomes (McBroome et al. 2021), an input that MAPLE and other existing maximum likelihood methods cannot handle currently. CMAPLE took 11 d and 15.4 GB RAM. Therefore, CMAPLE is highly efficient.

Analyzing these large alignments using existing maximum likelihood software is impractical. Therefore, we benchmarked CMAPLE against IQ-TREE 2 (v2.2.5) and FastTree 2 (Double-precision executable for nearly-identical sequences, v2.1.11; Price et al. 2010) on two smaller alignments with 5K and 10K real SARS-CoV-2 sequences. CMAPLE is 23× and 300× faster and requires 62 and 51 times less memory than FastTree 2 and IQ-TREE 2, respectively, while also producing trees with higher likelihoods (supplementary table S1, Supplementary Material online). CMAPLE took 8 min and 0.24 GB RAM to reconstruct a tree from 10K sequences, while FastTree 2 and IQ-TREE 2 took 3 and 40.5 h and 15.22 and 12.54 GB RAM, respectively (Fig. 1).

Software Validation

We validated our implementation by using IQ-TREE to compute the likelihoods of the trees inferred by CMAPLE and MAPLE from the real SARS-CoV-2 alignments of our testing data sets above. We found that the likelihoods of the trees inferred by CMAPLE and MAPLE are at most 0.003% different from each other (supplementary table S1, Supplementary Material online).

We validated our branch support implementation in CMAPLE with IQ-TREE 2 on the same testing data sets. The SH-aLRT branch supports computed by both programs on the CMAPLE-inferred trees have a Pearson correlation coefficient of 0.995 to 0.997 (supplementary fig. S1, Supplementary Material online). We observed 1.98% to 3.76% and 0.17% to 0.33% of branches where the two support values differed by more than 10% and 20%, respectively. We anticipate that these differences are due to the approximate nature of the CMAPLE likelihoods.

We also examined the code quality using SoftWipe (Zapletal et al. 2021) and achieved an overall absolute score of 8.0/10, ranked third out of 53 computational tools examined at <https://github.com/adrianzap/softwipe/wiki/Code-Quality-Benchmark> (access date: 2024 February 2). All data sets and the testing scripts are provided in the Supplementary Material.

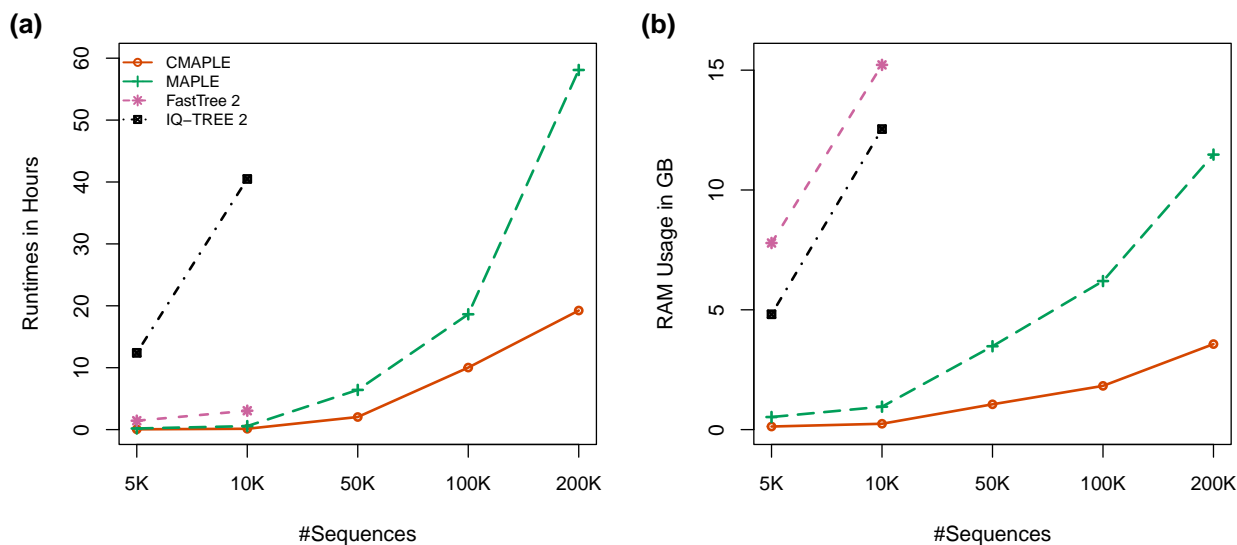


Fig. 1. Runtimes a) and peak memory consumptions b) of CMAPLE, MAPLE, FastTree 2, and IQ-TREE 2 on analyzing 5K, 10K, 50K, 100K, and 200K SARS-CoV-2 genomes subsampled from a global data set of 4.3 million genomes available on 2022 April 2 (McBroom et al. 2021).

Documentation and User Support

CMAPLE is open source and freely available at <https://github.com/iqtree/cmaple>. We provide two executables: “cmaple” and “cmaple-aa” for DNA and protein data, respectively. A comprehensive user manual and command reference of the CMAPLE software are available at <https://github.com/iqtree/cmaple/wiki>. User support, bug reports, and feature requests can be conveniently submitted at <https://github.com/iqtree/cmaple/issues>. API documentation and instructions on how to use the CMAPLE library are available at <http://iqtree.org/cmaple/>.

Discussions

Existing phylogenetic software, such as IQ-TREE, RAxML, and FastTree 2, was primarily developed for general tree reconstructions, thus may become slow and inefficient when handling large alignments with closely related sequences, as demonstrated in our benchmark and by De Maio et al. (2023). CMAPLE was specifically designed to address that problem.

We have here presented a sequential tree search algorithm, CMAPLE, which successfully improves upon MAPLE for larger pandemic-scale phylogenetic reconstruction. We plan to further reduce CMAPLE’s runtime by parallelizing tree search using OpenMP (Chapman et al. 2007) and/or message passing interface (MPI; Gropp et al. 1998). During tree search, CMAPLE seeks SPR moves for internal branches in a sequential manner. To expedite this computationally intensive task, OpenMP can employ multithreading on a single machine, while MPI allows multiprocessing in a high-performance computing cluster. Besides, we will also consider other parallelization approaches, such as parallel tree searches,

parallelizing the likelihood calculation over genome list entries (i.e. groups of sites, see De Maio et al. 2023; similar to the approaches of PLL [Flouri et al. 2015], BEAGLE [Ayres et al. 2012], IQ-TREE, and RAxML) and potentially utilizing GPUs (Smith et al. 2024). These ideas require substantial efforts in design and implementation, thus beyond the scope of the current study.

CMAPLE currently applies GTR for DNA and LG for protein data as default choices. It would be desirable to devise a model selection mechanism to automatically choose the most appropriate substitution model for a given data set, for example, by the Bayesian information criterion (Schwarz 1978) and Akaike information criterion (Akaike 1974). Besides, the current MAPLE algorithm assumes all sites in the alignment evolve at the same substitution rate. We plan to relax this assumption by implementing models of rate heterogeneity across sites (Yang 1994b).

Another key challenge in genomic epidemiological analysis is to deal with sequencing errors (De Maio et al. 2020; Turakhia et al. 2020). Not accounting for sequencing errors can lead to inaccurate inference (Turakhia et al. 2020). We plan to implement sequence error models (Felsenstein 2004; Chen et al. 2022) to enhance the robustness and accuracy of CMAPLE.

Apart from providing CMAPLE as a standalone program, we also provide it as an API that can be readily deployed in any C++ code. The API provides the `cmaple::isEffective()` function to quickly test the efficiency of the CMAPLE algorithm for a data set at hand. If efficient, developers can invoke the CMAPLE library as provided in the tutorial. Besides, we plan to create a Python package for CMAPLE to approach a wider user base (e.g. similar to Wang et al. 2023).

The CMAPLE library complements existing phylogenetic libraries, PLL and BEAGLE, but cannot replace them

because the CMAPLE algorithm only works on low divergence sequences. In the future, we plan to combine the methods in CMAPLE with classical phylogenetic approaches (such as those implemented within the libraries PLL and BEAGLE) to develop a unifying approach that would work efficiently at any level of divergence, for example, switching from classical algorithms and data structures to those of CMAPLE as one moves from long branches in the tree into densely sampled clades.

Supplementary Material

Supplementary material is available at *Molecular Biology and Evolution* online.

Acknowledgments

The computational results have been obtained on the cluster at the Center for Integrative Bioinformatics Vienna (CIBIV). We thank Arndt von Haeseler for providing access to the CIBIV cluster and Prabhav Kalaghatgi, Robert Lanfear, and Thomas Wong for their valuable comments and discussions.

Funding

This work was supported by a Chan-Zuckerberg Initiative grant for open-source software for science (EOSS4-0000000312 to B.Q.M.); an Australian Research Council Discovery grant (DP200103151 to B.Q.M.); a Moore-Simons Foundation grant (735923LPI [<https://doi.org/10.46714/735923LPI>] to B.Q.M.); and partly by a Vingroup Science and Technology Scholarship (VGRS20042M to N.L.T.).

Data Availability

The data underlying this article are available in the Zenodo Repository at <https://doi.org/10.5281/zenodo.11180100>.

References

De Maio N, Walker C, Borges R, Weilguny L, Slodkowicz G, Goldman N. 2020. Issues with SARS-CoV-2 sequencing data. *IEEE Trans Automat Contr*. 1974;19(6):716–723. <https://doi.org/10.1109/TAC.1974.1100705>.

Akaike H. A new look at the statistical model identification. *IEEE Trans Automat Contr*. 1974;19(6):716–723. <https://doi.org/10.1109/TAC.1974.1100705>.

Ayres DL, Darling A, Zwickl DJ, Beerli P, Holder MT, Lewis PO, Huelsenbeck JP, Ronquist F, Swofford DL, Cummings MP, et al. BEAGLE: an application programming interface and high-performance computing library for statistical phylogenetics. *Syst Biol*. 2012;61(1):170–173. <https://doi.org/10.1093/sysbio/syr100>.

Chapman B, Jost G, van der Pas R. *Using OpenMP: portable shared memory parallel programming (scientific and engineering computation)*. Cambridge: The MIT Press; 2007.

Chen K, Moravec JC, Gavryushkin A, Welch D, Drummond AJ. Accounting for errors in data improves divergence time estimates in single-cell cancer evolution. *Mol Biol Evol*. 2022;39(8):1–12. <https://doi.org/10.1093/molbev/msac143>.

Dang CC, Minh BQ, McShea H, Masel J, James JE, Vinh LS, Lanfear R. nQMaker: estimating time nonreversible amino acid

substitution models. *Syst Biol*. 2022;71(5):1110–1123. <https://doi.org/10.1093/sysbio/syac007>.

De Maio N, Kalaghatgi P, Turakhia Y, Corbett-Detig R, Minh BQ, Goldman N. Maximum likelihood pandemic-scale phylogenetics. *Nat Genet*. 2023;55(5):746–752. <https://doi.org/10.1038/s41588-023-01368-0>.

Felsenstein J. Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Syst Biol*. 1973;22(3):240–249. <https://doi.org/10.1093/sysbio/22.3.240>.

Felsenstein J. PHYLIP-Phylogeny Inference Package (Version 3.2). *Cladistics*. 1989;5:164–166.

Felsenstein J. *Inferring phylogenies*. Sunderland: Sinauer Associates, Inc; 2004.

Flouri T, Izquierdo-Carrasco F, Darriba D, Aberer AJ, Nguyen LT, Minh BQ, Von Haeseler A, Stamatakis A. The phylogenetic likelihood library. *Syst Biol*. 2015;64(2):356–362. <https://doi.org/10.1093/sysbio/syu084>.

Gonzalez-Reiche AS, Hernandez MM, Sullivan MJ, Ciferri B, Alshammery H, Obla A, Fabre S, Kleiner G, Polanco J, Khan Z, et al. Introductions and early spread of SARS-CoV-2 in the New York City area. *Science* (1979). 2020;369(6501):297–301. <https://doi.org/10.1126/science.abc1917>.

Gropp W, Huss-Lederman S, Lumsdaine A, Lusk E, Nitzberg B, Saphir W, Snir M. *MPI—the complete reference*. Cambridge: The MIT Press; 1998.

Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W, Gascuel O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol*. 2010;59(3):307–321. <https://doi.org/10.1093/sysbio/syq010>.

Hadfield J, Megill C, Bell SM, Huddlestone J, Potter B, Callender C, Sagulenko P, Bedford T, Neher RA. NextStrain: real-time tracking of pathogen evolution. *Bioinformatics*. 2018;34(23):4121–4123. <https://doi.org/10.1093/bioinformatics/bty407>.

Hodcroft EB, Zuber M, Nadeau S, Vaughan TG, Crawford KHD, Althaus CL, Reichmuth ML, Bowen JE, Walls AC, Corti D, et al. Spread of a SARS-CoV-2 variant through Europe in the summer of 2020. *Nature*. 2021;595(7869):707–712. <https://doi.org/10.1038/s41586-021-03677-y>.

Jukes TH, Cantor CR. Evolution of protein molecules. *Mammalian protein metabolism*. Cambridge: Academic Press; 1969. p. 21–132.

Kozlov AM, Darriba D, Flouri T, Morel B, Stamatakis A. RAXML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*. 2019;35(21):4453–4455. <https://doi.org/10.1093/bioinformatics/btz305>.

Lewis PO. NCL: a C++ class library for interpreting data files in NEXUS format. *Bioinformatics*. 2003;19(17):2330. <https://doi.org/10.1093/bioinformatics/btg319>.

Lu J, du Plessis L, Liu Z, Hill V, Kang M, Lin H, Sun J, François S, Kraemer MUG, Faria NR, et al. Genomic epidemiology of SARS-CoV-2 in Guangdong Province, China. *Cell*. 2020;181(5):997–1003.e9. <https://doi.org/10.1016/j.cell.2020.04.023>.

McBroome J, Thornlow B, Hinrichs AS, Kramer A, De Maio N, Goldman N, Haussler D, Corbett-Detig R, Turakhia Y. A daily-updated database and tools for comprehensive SARS-CoV-2 mutation-annotated trees. *Mol Biol Evol*. 2021;38(12):5819–5824. <https://doi.org/10.1093/molbev/msab264>.

Minh BQ, Dang CC, Vinh LS, Lanfear R. QMaker: fast and accurate method to estimate empirical models of protein evolution. *Syst Biol*. 2021;70(5):1046–1060. <https://doi.org/10.1093/sysbio/syab010>.

Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, Lanfear R. IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era. *Mol Biol Evol*. 2020;37(5):1530–1534. <https://doi.org/10.1093/molbev/msaa015>.

Nguyen LT, Schmidt HA, von Haeseler A, Minh, QB. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-

- likelihood phylogenies. *Mol Biol Evol.* 2015;**32**(1):268–274. <https://doi.org/10.1093/molbev/msu300>.
- Parks DH, Chuvochina M, Waite DW, Rinke C, Skarshewski A, Chaumeil PA, Hugenholtz P. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nat Biotechnol.* 2018;**36**(10):996. <https://doi.org/10.1038/nbt.4229>.
- Price MN, Dehal PS, Arkin AP. FastTree 2—approximately maximum-likelihood trees for large alignments. *PLoS One.* 2010;**5**(3):e9490. <https://doi.org/10.1371/journal.pone.0009490>.
- Schwarz G. Estimating the dimension of a model. *Annal Statist.* 1978;**6**(2):461–464. <https://doi.org/10.1214/aos/1176344136>.
- Smith K, Ayres D, Neumaier R, Worheide G, Hohna S. Bayesian phylogenetic analysis on multi-core compute architectures: implementation and evaluation of BEAGLE in RevBayes with MPI. *Syst Biol.* 2024;**In Press**:syae005. <https://doi.org/10.1093/sysbio/syae005>.
- Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics.* 2014;**30**(9):1312–1313. <https://doi.org/10.1093/bioinformatics/btu033>.
- Swofford DL, Olsen GJ. Phylogeny reconstruction. In: Hillis DM, Moritz Eds C, editors. *Molecular systematics*. Sunderland: Sinauer Associates; 1990. p. 411–501.
- Swofford DL, Olsen GJ, Waddell PJ, Hillis DM. Phylogenetic inference. *Molecular systematics*. Sunderland: Sinauer Associates; 1996. p. 407–514.
- Tavaré SMR. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures Mathematics Life Sci.* 1986;**17**:57–86.
- Turakhia Y, de Maio N, Thornlow B, Gozashti L, Lanfear R, Walker CR, Hinrichs AS, Fernandes JD, Borges R, Slodkowitz G, et al. Stability of SARS-CoV-2 phylogenies. *PLoS Genet.* 2020;**16**(11):1–34. <https://doi.org/10.1371/journal.pgen.1009175>.
- Turakhia Y, Thornlow B, Hinrichs AS, De Maio N, Gozashti L, Lanfear R, Haussler D, Corbett-Detig R. Ultrafast Sample placement on Existing tRees (USHER) enables real-time phylogenetics for the SARS-CoV-2 pandemic. *Nat Genet.* 2021;**53**(6):809–816. <https://doi.org/10.1038/s41588-021-00862-7>.
- Vöhringer HS, Sanderson T, Sinnott M, De Maio N, Nguyen T, Goater R, Schwach F, Harrison I, Hellewell J, Ariani CV, et al. Genomic reconstruction of the SARS-CoV-2 epidemic in England. *Nature.* 2021;**600**(7889):506–511. <https://doi.org/10.1038/s41586-021-04069-y>.
- Wang W, Barbetti J, Wong T, Thornlow B, Corbett-Detig R, Turakhia Y, Lanfear R, Minh BQ. DecentTree: scalable neighbour-joining for the genomic era. *Bioinformatics.* 2023;**39**(9):btad536. <https://doi.org/10.1093/bioinformatics/btad536>.
- Yang Z. Estimating the pattern of nucleotide substitution. *J Mol Evol.* 1994a;**39**(1):105–111. <https://doi.org/10.1007/BF00178256>.
- Yang Z. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol.* 1994b;**39**(3):306–314. <https://doi.org/10.1007/BF00160154>.
- Zapletal A, Höhler D, Sinz C, Stamatakis A. The SoftWipe tool and benchmark for assessing coding standards adherence of scientific software. *Sci Rep.* 2021;**11**(1):8–13. <https://doi.org/10.1038/s41598-021-89495-8>.