The Journal of Open Source Software

BTE: a Python module for pandemic-scale mutation-annotated phylogenetic trees

Jakob McBroome [1,2¶], **Yatish Turakhia** [3], **and Russell Corbett-Detig** [1,2]

1 Department of Biomolecular Engineering, University of California Santa Cruz. Santa Cruz, CA 95064, USA **2** Genomics Institute, University of California Santa Cruz, Santa Cruz, CA 95064, USA **3** Department of Electrical and Computer Engineering, University of California, San Diego; San Diego, CA 92093, USA ¶ Corresponding author

DOI: 10.21105/joss.04433

Software

- Review
- Repository
- Archive

Editor: Jacob Schreiber

Reviewers:

- @bede
- @tijeco

Submitted: 17 May 2022
Published: 05 September 2022

Summary

The massive scale of the worldwide SARS-CoV-2 genome sequencing effort has driven substantial innovation in bioinformatics (Hodcroft et al., 2021), including the development of new highly-scalable approaches to construction and storage of phylogenetic trees. Among these critical innovations is the mutation-annotated tree (MAT), which can compress millions of samples into a highly compact data structure (McBroome et al., 2021; Turakhia et al., 2021). MAT-storing protocol buffers are now regularly being created and used for tracking SARS-CoV-2 lineages (McBroome et al., 2021), and can be freely obtained from UCSC by anyone, but there are relatively few tools capable of directly working with these files.

Big Tree Explorer (BTE) is a Python extension of the highly optimized Mutation Annotated Tree (MAT) C++ library, which underlies the popular and highly effective phylogenetics package UShER. BTE is written in Cython and provides an efficient and intuitive interface for traversing and manipulating mutation-annotated trees in a Python environment. It can load a mutation-annotated tree structure directly from a MAT protocol buffer file or from a Auspice-format JSON. Alternatively, it can automatically infer mutation annotations and create a MAT from a Variant Call Format (VCF) file and a Newick format tree file, or even load and manipulate a tree without mutation annotations from a Newick alone. BTE provides several options for traversal and tree manipulation, including node and mutation creation, relocation, and deletion. BTE also provides native support for node-level clade and lineage annotations, such as those included in UCSC SARS-CoV-2 MAT protocol buffers. BTE's Cython code also includes functionality not present in the original MAT library, such as nucleotide diversity estimation. Altogether, BTE provides much of the same basic tree-level functionality as competing packages, while also supporting mutation and lineage annotations, allowing any user to take advantage of the powerful MAT data structure.

BTE is intended for use by developers, epidemiologists, and data scientists. It can be a critical element in prototyping new phylogenetic methods, support scripts for manipulation and extraction of epidemiological information from SARS-CoV-2 genetic sequences, and much more.

Statement of Need

While there are multiple Python packages for phylogenetics available (Huerta-Cepas et al., 2016; Talevich et al., 2012), none are designed for handling MATs or to scale to tens of millions of tips. When attempting to use these packages with mutation-annotated trees, cumbersome file conversions to Newick and separate storage of mutations and tree structures adds substantial overhead to any analysis. BTE is designed explicitly for working with extremely large mutation-

annotated parsimony phylogenetic trees. It is both more computationally efficient than competing packages (Figure 1) and stores mutations and the tree within a streamlined data structure. BTE makes MATs and pandemic-scale phylogenies in general more accessible and useful to developers worldwide, helping to widen the SARS-CoV-2 bioinformatics bottleneck.

Comparison with Gold-Standard Alternatives

We compared performance of BTE as compared to two other popular packages for Python phylogenetics, ETE3 and Biopython.Phylo (Huerta-Cepas et al., 2016; Talevich et al., 2012). Benchmarking was performed by extracting random subtrees of the specified size from one of UCSC's global SARS-CoV-2 MATs, converting the subtree to Newick format, and performing the specified operation with each package. Most operations scale linearly with tree size in both runtime and memory use. The majority of memory use in a phylogenetics procedure comes from initally loading the data object into memory. Biopython.Phylo uses a partial load implementation that uses less memory than BTE or ETE, but is substantially slower in general (Figure 1). Generally, BTE loads and traverses a tree more quickly than competitors, with a particular advantage at large tree sizes. In terms of runtime, heavy use of multithreading and optimized structures gives BTE up to an order of magnitude advantage across loading and traversal operations. Notably, the operation "path time" (the inference of the complete ancestry and associated haplotype of a specific node) is several orders of magnitude improved, as it uses pointers to parent nodes instead of searching through the tree for the parent node at each step. Overall, BTE is substantially faster and scales better to extremely large phylogenies compared to competing packages. All code for benchmarking is available at a dedicated repository.

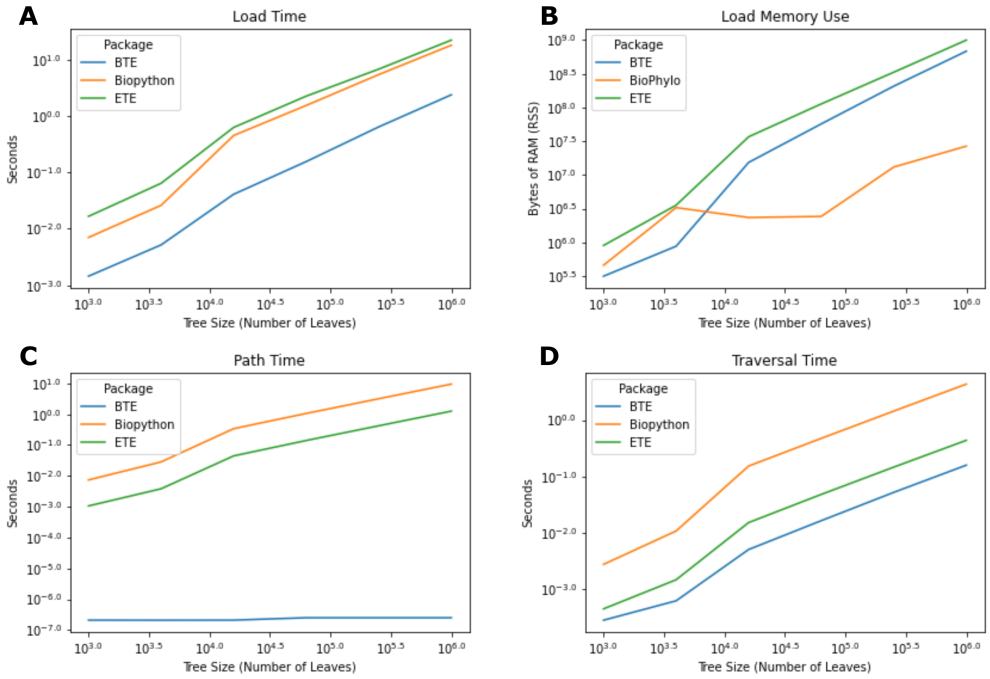

Figure 1: Runtime Comparison. A: Time to load a tree of the indicated size from disk. B: Total memory consumed by loading a tree of the indicated size with each package. C: Time to traverse a tree of the indicated size (depth-first/postorder). D: Time to trace the complete ancestry of a single randomly selected leaf on a tree of the indicated size.

Acknowledgements

We gratefully acknowledge our early adopters, including Alex Kramer and Will Dumm, who provided invaluable feedback and suggestions to make this package as effective, intuitive, and bug-free as possible. This work was supported by T32HG008345 and partially by the CDC award BAA 200-2021-11554.

References

Hodcroft, E. B., De Maio, N., Lanfear, R., MacCannell, D. R., Minh, B. Q., Schmidt, H. A., Stamatakis, A., Goldman, N., & Dessimoz, C. (2021). Want to track pandemic variants faster? Fix the bioinformatics bottleneck. *Nature*, *591*(7848), 30–33. https://doi.org/10.1038/d41586-021-00525-x

Huerta-Cepas, J., Serra, F., & Bork, P. (2016). ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular Biology and Evolution*, *33*(6), 1635–1638. https://doi.org/10.1093/molbev/msw046

McBroome, J., Thornlow, B., Hinrichs, A. S., Kramer, A., De Maio, N., Goldman, N., Haussler, D., Corbett-Detig, R., & Turakhia, Y. (2021). A Daily-Updated Database and Tools for Comprehensive SARS-CoV-2 Mutation-Annotated Trees. *Molecular Biology and Evolution*, *msab264*. https://doi.org/10.1093/molbev/msab264

Talevich, E., Invergo, B. M., Cock, P. J., & Chapman, B. A. (2012). Bio.Phylo: A unified toolkit for processing, analyzing and visualizing phylogenetic trees in Biopython. *BMC Bioinformatics*, *13*(1), 1–9. https://doi.org/10.1186/1471-2105-13-209

Turakhia, Y., Thornlow, B., Hinrichs, A. S., De Maio, N., Gozashti, L., Lanfear, R., Haussler, D., & Corbett-Detig, R. (2021). Ultrafast Sample placement on Existing tRees (UShER) enables real-time phylogenetics for the SARS-CoV-2 pandemic. *Nature Genetics*, *53*(6), 809–816. https://doi.org/10.1038/s41588-021-00862-7