The Journal of Open Source Software

phylo2vec: a library for vector-based phylogenetic tree manipulation

Neil Scheidwasser [1*], **Ayush Nag** [2*], **Matthew J Penn** [1], **Anthony MV Jakob** [4], **Frederik Mølkjær Andersen** [1], **Mark P Khurana** [1], **Landung Setiawan** [2], **Madeline Gordon** [2], **David A Duchêne** [1], **and Samir Bhatt** [1,3¶]

1 Section of Health Data Science and AI, University of Copenhagen, Copenhagen, Denmark **2** eScience Institute, University of Washington, Seattle, United States **3** MRC Centre for Global Infectious Disease Analysis, Imperial College London, London, United Kingdom **4** Independent researcher ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Phylogenetics is a fundamental component of evolutionary analysis frameworks in biology (Yang, 2014) and linguistics (Atkinson & Gray, 2005). Recently, the advent of large-scale genomics and the SARS-CoV-2 pandemic has highlighted the necessity for phylogenetic software to handle large datasets (Attwood et al., 2022; Kapli et al., 2020; Khurana et al., 2024; Kraemer et al., 2025). While significant efforts have focused on scaling optimisation algorithms (De Maio et al., 2023; Sanderson, 2021; Turakhia et al., 2021), visualization (Sanderson, 2022), and lineage identification (McBroome et al., 2024), an emerging body of research has been dedicated to efficient representations of data for genomes (Deorowicz et al., 2023) and phylogenetic trees (Chauve et al., 2025; Penn et al., 2024; Richman et al., 2025). Compared to the traditional Newick format which represents trees using strings of nested parentheses (Felsenstein, 2004), modern tree representations utilize integer vectors to define the tree topology traversal. This approach offers several advantages, including easier manipulation, increased memory efficiency, and applicability to machine learning.

Here, we present the latest release of `phylo2vec` (or Phylo2Vec), a high-performance software package for encoding, manipulating, and analysing binary phylogenetic trees. At its core, the package is based on the phylo2vec (Penn et al., 2024) representation of binary trees, and is designed to enable fast sampling and tree comparison. This release features a core implementation in Rust for improved performance and memory efficiency (Figure 1), with wrappers in R and Python (superseding the original release (Penn et al., 2024)), making it accessible to a broad audience in the bioinformatics community.

The Journal of Open Source Software

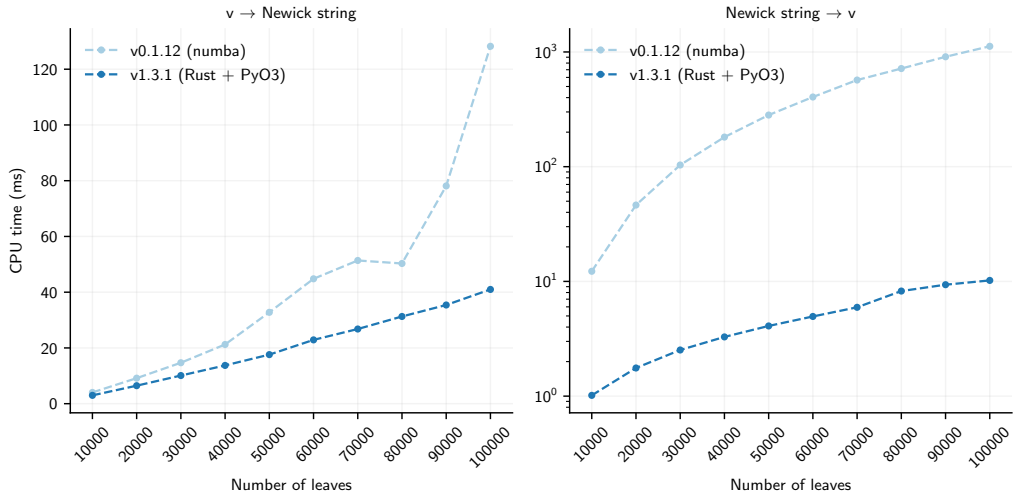

Figure 1: Benchmark times for converting a phylo2vec vector to a Newick string (left) and vice versa (right). Execution time was measured over at least 20 runs per size, comparing Python functions in the latest release (via Rust bindings with PyO3) against the previous release (Penn et al., 2024) based on `Numba` (Lam et al., 2015). All benchmarks were performed on a workstation equipped with an AMD Ryzen Threadripper PRO 5995WX (64 cores, 2.7 GHz) and 256 GB of RAM.

Statement of need

The purpose of the `phylo2vec` library is threefold. First, it provides robust phylogenetic tree manipulation in Rust, complementing other efforts such as `light_phylogeny` (Duchemin et al., 2018) for reconciled phylogenies (Nakhleh, 2013), and `rust-bio` (Köster, 2016), which does not yet cover phylogenetics. Second, it complements existing libraries such as ape (Paradis & Schliep, 2019) in R, and `ete3` (Huerta-Cepas et al., 2016) and `DendroPy` (Moreno et al., 2024) in Python, by providing fast tree sampling, fast tree comparison and efficient tree data compression (Penn et al., 2024). Third, the phylo2vec representation offers a pathway to using new optimisation frameworks for phylogenetic inference. A notable example is GradME (Penn et al., 2023), a gradient descent-based algorithm which uses a continuous relaxation of the phylo2vec representation.

Features

The presented release of `phylo2vec` addresses optimisations limitations of (Penn et al., 2024) with $\mathcal{O}(n \log n)$ implementations for vector-to-Newick and Newick-to-vector conversions, leveraging Adelson-Velsky and Landis (AVL) trees (Adelson-Velsky & Landis, 1962) and Fenwick trees (Fenwick, 1994), respectively.

New features include an extension of the vector representation to support branch length annotations (Figure 2), leaf-level operations (pruning, placement, MRCA identification), fast cophenetic distance matrix calculation, and various optimisation schemes based on phylo2vec tree representations, notably hill-climbing (Penn et al., 2024) and GradME (Penn et al., 2023). We also propose a likelihood function for Bayesian MCMC inference that leverages tree representation similarities with `BEAGLE` (Ayres et al., 2012; Suchard & Rambaut, 2009). Finally, user-friendliness is enhanced with step-by-step demos of phylo2vec's representations and core functions.

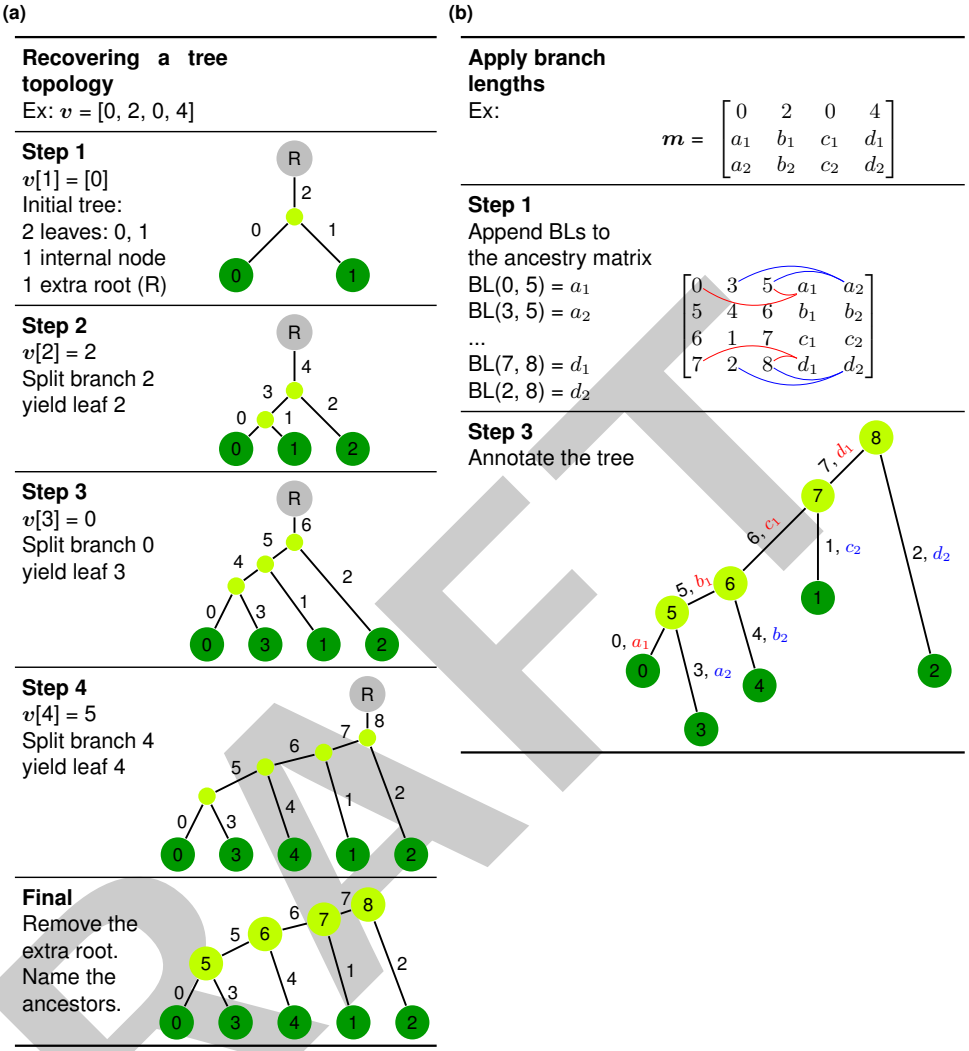

Figure 2: Recovering a tree from a phylo2vec vector: example for $v = [0, 2, 0, 4]$. (a) Leaf placement algorithm from (Penn et al., 2024). (b) Augmenting v into a matrix m with branch lengths. We use an intermediary ancestry matrix whereby each row encodes a cherry (two children + parent) with two extra columns for branch lengths. The node with the smallest descendant has the branch length in the second column, and the other the branch length in the third column.

Maintenance

A strong focus of this release is to support long-term maintenance through implementing recommended software practices into its project structure and development workflow. The project is structured with a Rust API containing core algorithms with language bindings to avoid tight coupling and enable easy language additions. Additionally, we have established a robust continuous integration (CI) pipeline using GitHub Actions, which features:

- Unit test frameworks for Rust (cargo), Python (pytest), and R (testthat (Wickham, 2011))
- Benchmarking on the Rust code (criterion) and its Python bindings (pytest-benchmark)

Lastly, to complement Jupyter Notebook demos, comprehensive documentation is provided using Jupyter Book and Read The Docs.

Acknowledgements

S.B. acknowledges funding from the MRC Centre for Global Infectious Disease Analysis (reference MR/X020258/1), funded by the UK Medical Research Council (MRC). This UK funded award is carried out in the frame of the Global Health EDCTP3 Joint Undertaking. S.B. acknowledges support from the Danish National Research Foundation via a chair grant (DNRF160, also supporting N.S. and M.P.K.), The Eric and Wendy Schmidt Fund For Strategic Innovation via the Schmidt Polymath Award (G-22-63345, also supporting M.J.P. and F.M.A.), and the Novo Nordisk Foundation via The Novo Nordisk Young Investigator Award (NNF20OC0059309). D.A.D. is supported by a Data Science - Emerging researcher award from Novo Nordisk Fonden (NNF23OC0084647).

References

Adelson-Velsky, Georgii, & Landis, E. (1962). An algorithm for the organization of information. *Proc. USSR Acad. Sci.*, *146*, 263–266.

Atkinson, Q. D., & Gray, R. D. (2005). Curious parallels and curious connections—phylogenetic thinking in biology and historical linguistics. *Syst. Biol.*, *54*(4), 513–526. https://doi.org/10.1080/10635150590950317

Attwood, S. W., Hill, S. C., Aanensen, D. M., Connor, T. R., & Pybus, O. G. (2022). Phylogenetic and phylodynamic approaches to understanding and combating the early SARS-CoV-2 pandemic. *Nat. Rev. Genet.*, *23*(9), 547–562. https://doi.org/10.1038/s41576-022-00483-8

Ayres, D. L., Darling, A., Zwickl, D. J., Beerli, P., Holder, M. T., Lewis, P. O., Huelsenbeck, J. P., Ronquist, F., Swofford, D. L., Cummings, M. P., & others. (2012). BEAGLE: An application programming interface and high-performance computing library for statistical phylogenetics. *Syst. Biol.*, *61*(1), 170–173. https://doi.org/10.1093/sysbio/syr100

Chauve, C., Colijn, C., & Zhang, L. (2025). A vector representation for phylogenetic trees. *Philos. Trans. R. Soc. B*, *380*(1919), 20240226. https://doi.org/10.1098/rstb.2024.0226

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

Deorowicz, S., Danek, A., & Li, H. (2023). AGC: Compact representation of assembled genomes with fast queries and updates. *Bioinformatics*, *39*(3), btad097. https://doi.org/10.1093/bioinformatics/btad097

Duchemin, W., Gence, G., Arigon Chifolleau, A.-M., Arvestad, L., Bansal, M. S., Berry, V., Boussau, B., Chevenet, F., Comte, N., Davín, A. A., & others. (2018). RecPhyloXML: A format for reconciled gene trees. *Bioinformatics*, *34*(21), 3646–3652. https://doi.org/10.1093/bioinformatics/bty389

Felsenstein, J. (2004). *Inferring phylogenies* (Vol. 2). Sinauer Associates.

Fenwick, P. M. (1994). A new data structure for cumulative frequency tables. *Softw. Pract. Exp.*, *24*(3), 327–336. https://doi.org/10.1002/spe.4380240306

Huerta-Cepas, J., Serra, F., & Bork, P. (2016). ETE 3: Reconstruction, analysis, and visualization of phylogenomic data. *Mol. Biol. Evol.*, *33*(6), 1635–1638. https://doi.org/10.1093/molbev/msw046

Kapli, P., Yang, Z., & Telford, M. J. (2020). Phylogenetic tree building in the genomic age. *Nat. Rev. Genet.*, *21*(7), 428–444. https://doi.org/10.1038/s41576-020-0233-0

Khurana, M. P., Curran-Sebastian, J., Scheidwasser, N., Morgenstern, C., Rasmussen, M., Fon-

ager, J., Stegger, M., Tang, M.-H. E., Juul, J. L., Escobar-Herrera, L. A., & others. (2024). High-resolution epidemiological landscape from~ 290,000 SARS-CoV-2 genomes from Denmark. *Nat. Commun.*, *15*(1), 7123. https://doi.org/10.1038/s41467-024-51371-0

Köster, J. (2016). Rust-bio: A fast and safe bioinformatics library. *Bioinformatics*, *32*(3), 444–446. https://doi.org/10.1093/bioinformatics/btv573

Kraemer, M. U., Tsui, J. L.-H., Chang, S. Y., Lytras, S., Khurana, M. P., Vanderslott, S., Bajaj, S., Scheidwasser, N., Curran-Sebastian, J. L., Semenova, E., & others. (2025). Artificial intelligence for modelling infectious disease epidemics. *Nature*, *638*(8051), 623–635. https://doi.org/10.1038/s41586-024-08564-w

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based Python JIT compiler. *Proc. 2nd Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6. https://doi.org/10.1145/2833157.2833162

McBroome, J., Bernardi Schneider, A. de, Roemer, C., Wolfinger, M. T., Hinrichs, A. S., O'Toole, A. N., Ruis, C., Turakhia, Y., Rambaut, A., & Corbett-Detig, R. (2024). A framework for automated scalable designation of viral pathogen lineages from genomic data. *Nat. Microbiol.*, *9*(2), 550–560. https://doi.org/10.1038/s41564-023-01587-5

Moreno, M. A., Holder, M. T., & Sukumaran, J. (2024). DendroPy 5: a mature Python library for phylogenetic computing. *J. Open Source Softw.*, *9*(101), 6943. https://doi.org/10.21105/joss.06943

Nakhleh, L. (2013). Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends Ecol. Evol.*, *28*(12), 719–728. https://doi.org/10.1016/j.tree.2013.09.004

Paradis, E., & Schliep, K. (2019). Ape 5.0: An environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, *35*, 526–528. https://doi.org/10.1093/bioinformatics/bty633

Penn, M. J., Scheidwasser, N., Khurana, M. P., Duchêne, D. A., Donnelly, C. A., & Bhatt, S. (2024). Phylo2Vec: A vector representation for binary trees. *Syst. Biol.*, syae030. https://doi.org/10.1093/sysbio/syae030

Penn, M. J., Scheidwasser, N., Penn, J., Donnelly, C. A., Duchêne, D. A., & Bhatt, S. (2023). Leaping through tree space: Continuous phylogenetic inference for rooted and unrooted trees. *Genome Biol. Evol.*, *15*(12), evad213. https://doi.org/10.1093/gbe/evad213

Richman, H., Zhang, C., & IV, F. A. M. (2025). Vector encoding of phylogenetic trees by ordered leaf attachment. In *arXiv*. https://doi.org/10.48550/arXiv.2503.10169

Sanderson, T. (2021). Chronumental: Time tree estimation from very large phylogenies. In *bioRxiv*. Cold Spring Harbor Laboratory. https://doi.org/10.1101/2021.10.27.465994

Sanderson, T. (2022). Taxonium, a web-based tool for exploring large phylogenetic trees. *eLife*, *11*. https://doi.org/10.7554/eLife.82392

Suchard, M. A., & Rambaut, A. (2009). Many-core algorithms for statistical phylogenetics. *Bioinformatics*, *25*(11), 1370–1376. https://doi.org/10.1093/bioinformatics/btp244

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. *Nat. Genet.*, *53*(6), 809–816. https://doi.org/10.1038/s41588-021-00862-7

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, *3*(1), 5–10. https://doi.org/10.32614/rj-2011-002

Yang, Z. (2014). *Molecular evolution: A statistical approach*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199602605.001.0001