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], **Don 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 **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Published: unpublished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 many analysis frameworks in computational and evolutionary biology (Yang, 2014) as well as linguistics (Atkinson & Gray, 2005). Recently, the advent of large-scale genomics and the SARS-CoV-2 pandemic has underscored the necessity to scale phylogenetic software to handle large datasets of genomes or phylogenetic trees (Attwood et al., 2022; Kapli et al., 2020; Khurana et al., 2024; Kraemer et al., 2025). While significant efforts have focused on scaling phylogenetic inference (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 such as phylo2vec (Penn et al., 2024), HOP (Chauve et al., 2025), and OLA (Richman et al., 2025). Compared to traditional tree representations such as the Newick format (Felsenstein, 2004), which describes a phylogenetic tree as a string of nested parentheses enclosing pairs of leaves or subtrees, these modern representations utilize integer vectors to define the tree topology traversal. This approach offers several advantages, including easier manipulability, increased memory efficiency, and applicability to downstream tasks such as machine learning (Penn et al., 2024).

Here, we present the new release of `phylo2vec`, a high-performance software package for encoding, manipulating, and analyzing binary phylogenetic trees. At its core, the package is based on the phylo2vec (Penn et al., 2024) representation of binary trees, which defines a bijection from any tree topology with n leaves into an integer vector of size $n-1$. Compared to the traditional Newick format, phylo2vec was designed with fast sampling and rapid tree comparison in mind. This release features a core implementation in Rust, providing significant performance improvements and memory efficiency (Figure 1), while remaining available in Python (superseding the release described in the original paper (Penn et al., 2024)) and R via dedicated wrappers, making it accessible to a broad audience in the bioinformatics community.

The Journal of Open Source Software

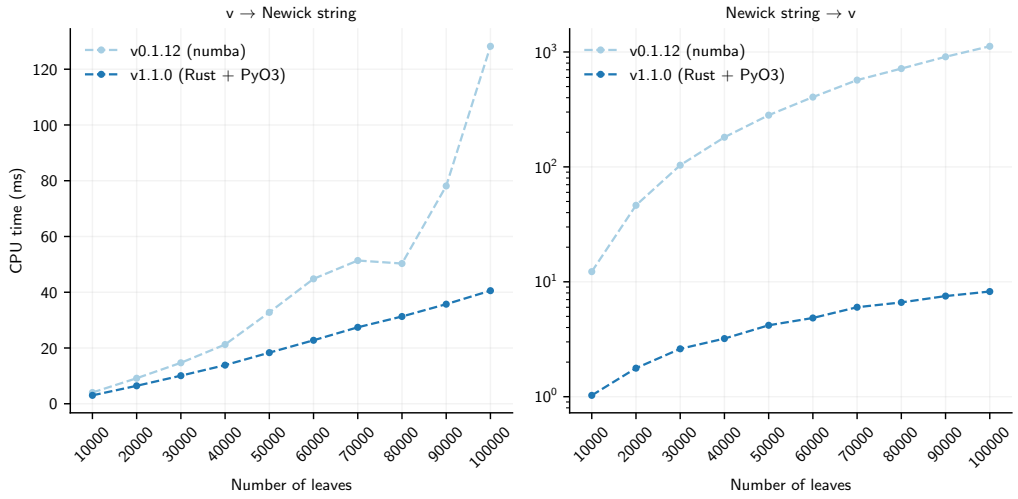

Figure 1: Benchmark times for converting a phylo2Vec vector to a Newick string (left) and vice versa (right). For each size, we evaluated the execution time for a minimum of 20 rounds using `pytest-benchmark`. We compare the execution time of the Python functions in the latest release, which rely on Rust bindings via PyO3, with the previous release (Penn et al., 2024), which make use of just-in-time (JIT) compilation of Python functions using Numba (Lam et al., 2015)

Statement of need

The purpose of the `phylo2vec` library is threefold. First, the core of the library aims at providing a robust phylogenetic tree manipulation library in Rust, complementing other efforts such as `light_phylogeny` (Duchemin et al., 2018), which focuses on tree visualization and manipulation of reconciled phylogenies (Nakhleh, 2013), and `rust-bio` (Köster, 2016), a comprehensive bioinformatics library which does not yet cover phylogenetics. Second, `phylo2vec` aims at complementing existing phylogenetic 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 inherent tree representation of phylo2vec offers a pathway to gradient-based optimization frameworks for phylogenetic inference. A notable example is GradME (Penn et al., 2023), which relaxes the vector representation of phylo2vec into a continuous space.

Features

The presented release of `phylo2vec` addresses several limitations of (Penn et al., 2024). In particular, it allows for branch length annotations, extending the vector representation of size $n-1$ to a matrix of size $(n-1)\times 3$, where n denotes the number of leaves (or taxa) in a tree (Figure 2), a $\mathcal{O}(n\log n)$ implementation of vector-to-Newick conversion based on Adelson-Velsky and Landis (AVL; (Adelson-Velsky & Landis, 1962)) trees, and a $\mathcal{O}(n\log n)$ implementation of Newick-to-vector conversion making use of Fenwick trees (Fenwick, 1994) during the vector construction. Moreover, the current release features several new additions, including several leaf-level operations (pruning, placement, MRCA finding), fast cophenetic distance matrix calculation, and a skeleton for Bayesian phylogenetic inference using Markov Chain Monte Carlo (MCMC) in the highly optimised `Beagle` library that underpins a number of phylogenetic software (Ayres et al., 2012; Suchard & Rambaut, 2009). The inference framework leverages similarities between phylo2vec and `BEAGLE`'s inner representation of post-order traversal. Lastly, user-friendliness is enhanced by step-by-step demos of the inner workings of phylo2vec's vector representation.

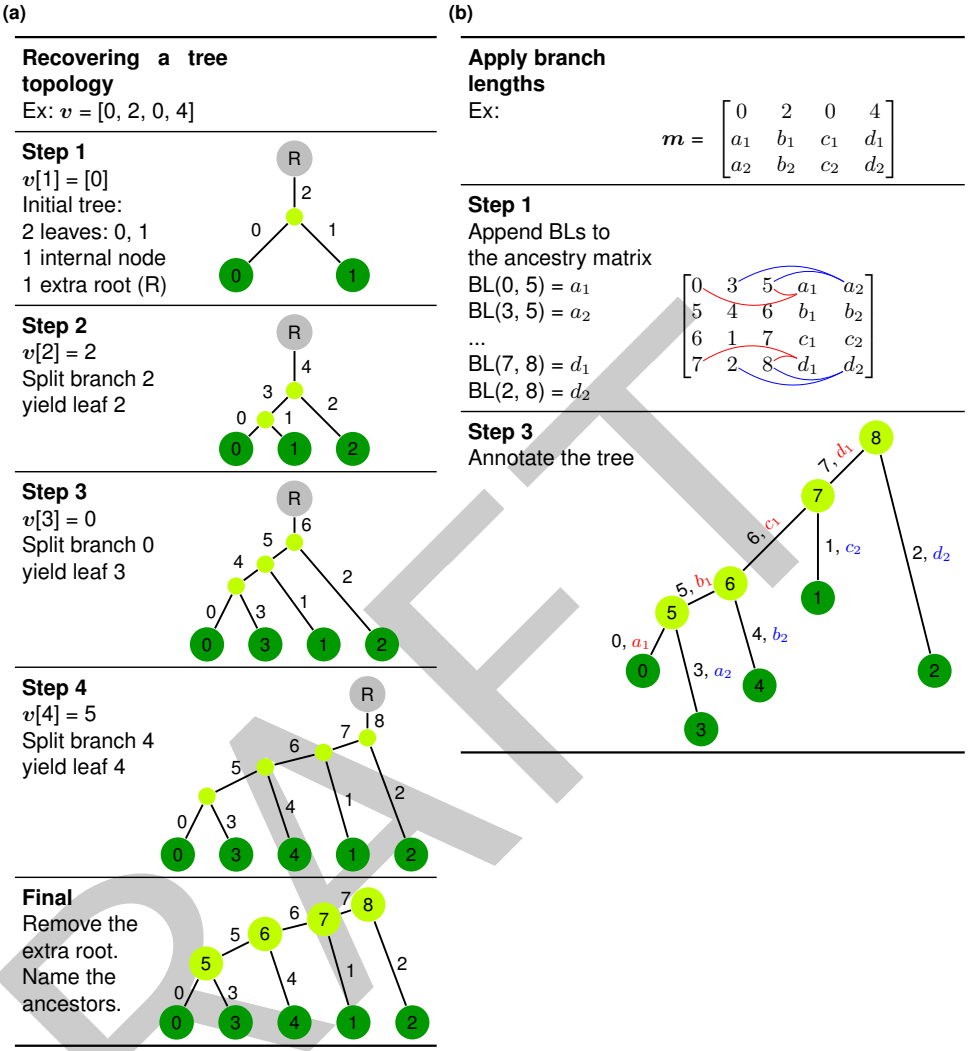

Figure 2: Recovering a tree from a phylo2vec vector: example for $v = [0, 2, 0, 4]$. (a) Main algorithm for leaf placement described in (Penn et al., 2024). (b) Augmenting the phylo2vec vector into a matrix m with branch lengths. We use an intermediary ancestry matrix whereby each row describes a cherry (two children nodes and the parent node), which we augment with two columns of branch lengths. These columns denote the length of the branch connecting each parent and their two children nodes, respectively.

Maintenance

With Phylo2Vec, we aim to support long-term maintenance through implementing recommended software practices explicitly into the structure of the project and development workflow, rather than leaving them implied. This avoids human error as the repo's structure itself enforces good practices, rather than placing the responsibility solely on code contributors. More specifically, we have structured the project such that the Rust API contains the core algorithms, and all other language components are APIs that bind to the Rust functions. This avoids tight coupling, as it allows for the possibility of adding new languages to bind to the Rust API's, without needing to change anything in the Rust project itself. 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 Jupyterbook and Rustdoc for Python and Rust components, respectively.

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) which also supports N.S. and M.P.K. S.B. acknowledges support from The Eric and Wendy Schmidt Fund For Strategic Innovation via the Schmidt Polymath Award (G-22-63345) which also supports M.P. and F.M.A. S.B. acknowledges support from 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. *Soviet Math.*, *3*, 1259–1263.

Atkinson, Q. D., & Gray, R. D. (2005). Curious parallels and curious connections—phylogenetic thinking in biology and historical linguistics. *Systematic Biology*, *54*(4), 513–526.

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. *Nature Reviews Genetics*, *23*(9), 547–562.

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. *Systematic Biology*, *61*(1), 170–173.

Chauve, C., Colijn, C., & Zhang, L. (2025). A vector representation for phylogenetic trees. *Philosophical Transactions B*, *380*(1919), 20240226.

De Maio, N., Kalaghatgi, P., Turakhia, Y., Corbett-Detig, R., Minh, B. Q., & Goldman, N. (2023). Maximum likelihood pandemic-scale phylogenetics. *Nature Genetics*, *55*(5), 746–752.

Deorowicz, S., Danek, A., & Li, H. (2023). AGC: Compact representation of assembled genomes with fast queries and updates. *Bioinformatics*, *39*(3), 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., Dessimoz, C., Dylus, D., Hasic, D., Mallo, D., Planel, R., Posada, D., Scornavacca, C., Szöllősi, G., Zhang, L., … Daubin, V. (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. *Software: Practice and Experience*, *24*(3), 327–336.

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.

Kapli, P., Yang, Z., & Telford, M. J. (2020). Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, *21*(7), 428–444.

Khurana, M. P., Curran-Sebastian, J., Scheidwasser, N., Morgenstern, C., Rasmussen, M., Fonager, 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. *Nature Communications*, *15*(1), 7123.

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

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.

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A llvm-based python jit compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6.

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. *Nature Microbiology*, *9*(2), 550–560.

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

Nakhleh, L. (2013). Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology & Evolution*, *28*(12), 719–728.

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

Penn, M. J., Scheidwasser, N., Khurana, M. P., Duchêne, D. A., Donnelly, C. A., & Bhatt, S. (2024). Phylo2Vec: A vector representation for binary trees. *Systematic Biology*, 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 Biology and Evolution*, *15*(12), evad213.

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

Sanderson, T. (2021). Chronumental: Time tree estimation from very large phylogenies. In *bioRxiv* (pp. 2021–2010). Cold Spring Harbor Laboratory.

Sanderson, T. (2022). Taxonium, a web-based tool for exploring large phylogenetic trees. *Elife*, *11*.

Suchard, M. A., & Rambaut, A. (2009). Many-core algorithms for statistical phylogenetics. *Bioinformatics*, *25*(11), 1370–1376.

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.

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, *3*(1), 5–10.

Yang, Z. (2014). *Molecular evolution: A statistical approach*. Oxford University Press.