




# PhyloX: A Python package for complete phylogenetic network workflows

Remie Janssen <sup>1</sup>

<sup>1</sup> National Institute for Public Health and the Environment, Bioinformatics and Computing group, Bilthoven, The Netherlands

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Frederick Boehm](#) 

## Reviewers:

- [@abhishektiwari](#)
- [@bgiori](#)

Submitted: 20 January 2024

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

PhyloX is a Python package with tools for generating, manipulating, and analyzing phylogenetic networks. It uses the NetworkX package ([Hagberg et al., 2008](#)) for basic graph operations. This has the added benefit that the powerful graph tools from networkX can be used directly on the phylogenetic networks as well. The aim of the package is to be of general use to phylogenetic network researchers, with a current focus on I/O, random generation of networks, cherry-picking methods, rearrangement operations, and the identification of classes and properties of networks.

## Statement of Need

Currently, there is no Python package that enables a full workflow for analysing properties and methods of phylogenetic networks. Isolated scripts for this purpose do appear on GitHub or as pseudo-code regularly, most often as part of publications studying one method or one property ([Remie Janssen et al., 2020](#); [R. Janssen, 2021](#); [Remie Janssen & Murakami, 2020](#); [Pons et al., 2019](#); [Zhang et al., 2018](#)). Combining such scripts requires quite some work, for example because the phylogenetic networks themselves are represented by different Python classes with their own methods.

This package, PhyloX, aims to bring these scripts together: it standardizes implementations of several basic objects related to phylogenetic networks, such as the networks themselves, the labelling of the nodes, and rearrangement moves. It currently implements a limited but important set of basic functions: I/O for networks (e.g. lists of edges and extended newick format), network generation for test sets, comparing networks resulting from reconstruction methods, and computing several well-used network properties such as the reticulation number, the level, and the number of cherries.

## Related packages

As mentioned above, there are currently no Python packages that enable a complete workflow for phylogenetic networks. There are, however, a few Python packages that seem to fit that bill to a certain extent. We will argue that, despite these packages being available, there is still a need, or at least a great benefit, of using PhyloX.

## PhyloNetwork

Like PhyloX, [PhyloNetwork](#) is a Python package based on NetworkX. It has a richer implementation for phylogenetic trees than PhyloX. For example, it includes more tree-specific rearrangement moves, the calculation of node properties such as the latest common ancestor (LCA), and some presets for drawing networks.

However, it has very few methods for phylogenetic networks and most of those methods are also included in PhyloX. Besides the network methods implemented in PhyloNetwork, PhyloX also includes isomorphism checking, rearrangement methods and distances for networks, more random network and tree generators, some functionality to combine trees or networks, and calculation of network properties. Another advantage of using PhyloX over PhyloNetwork is the inclusion of explicit random seeds. This is an important factor for the reproducibility of research.

Note that code from PhyloNetwork and PhyloX may be easy to combine, as both use NetworkX to implement the phylogenetic network class.

## Biopython - Phylo

This phylogenetics part, [Phylo](#) ([Talevich et al., 2012](#)), of the Biopython package ([Cock et al., 2009](#)) is built for phylogenetic analyses in Python. However, it is set up for phylogenetic trees only. The encoding of trees as sets of [clades](#) does not easily allow extension to networks, which makes it unsuitable to use for these phylogenetic networks methods.

## DendroPy

Like Biopython's phylogenetics package, the [DendroPy](#) package focuses on phylogenetic trees ([Sukumaran & Holder, 2010](#)). Unlike Biopython, the implementation of the trees in DendroPy does seem to be graph based, making it more feasible to implement phylogenetic networks in DendroPy. This could still require large changes, as some properties of trees are built into the code on a fairly fundamental level, such as each node having (at most one) [parent node](#).

## Availability

The code of PhyloX is available as open source project on [GitHub](#) under the BSD-3-Clause license. The package is also available via [PyPI](#), so it can be installed via pip (or pip in conda), and updates to the release branch are automatically converted into new versions of the package. The releases are recorded in [Zenodo](#) so persistent identifiers can be used to cite specific releases of the software. When citing this software, please make sure to also cite the original source of the code, which is mentioned in the [documentation](#) of each method or class.

## Acknowledgements

Most of the code has been written in the form of separate scripts during the author's PhD project, which was conducted under Leo van Iersel's [Vidi](#) grant: 639.072.6

Anyone willing to contribute is very welcome to do so via pull requests and issues on github!

## References

- Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & Hoon, M. J. L. de. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11), 1422–1423. <https://doi.org/10.1093/bioinformatics/btp163>
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15).
- Janssen, R. (2021). *Rearranging phylogenetic networks* [PhD thesis]. Ph. D. thesis, Delft University of Technology.

- 79 Janssen, Remie, Jones, M., & Murakami, Y. (2020). Combining networks using cherry picking  
80 sequences. *International Conference on Algorithms for Computational Biology*, 77–92.
- 81 Janssen, Remie, & Murakami, Y. (2020). Linear time algorithm for tree-child network  
82 containment. *International Conference on Algorithms for Computational Biology*, 93–107.
- 83 Pons, J. C., Scornavacca, C., & Cardona, G. (2019). Generation of level- $k$  LGT networks.  
84 *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1), 158–164.
- 85 Sukumaran, J., & Holder, M. T. (2010). DendroPy: A python library for phylogenetic  
86 computing. *Bioinformatics*, 26(12), 1569–1571.
- 87 Talevich, E., Invergo, B. M., Cock, P. J., & Chapman, B. A. (2012). Bio. Phylo: A unified  
88 toolkit for processing, analyzing and visualizing phylogenetic trees in biopython. *BMC*  
89 *Bioinformatics*, 13, 1–9.
- 90 Zhang, C., Ogilvie, H. A., Drummond, A. J., & Stadler, T. (2018). Bayesian inference of  
91 species networks from multilocus sequence data. *Molecular Biology and Evolution*, 35(2),  
92 504–517.

DRAFT