




LinguiPhyR: A Package for Linguistic Phylogenetic Analysis in R

Marc E. Canby ¹

¹ University of Illinois at Urbana-Champaign, USA

DOI: [10.21105/joss.06201](https://doi.org/10.21105/joss.06201)

Software

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

Editor: [Andrew Stewart](#)  

Reviewers:

- [@fauxneticien](#)
- [@SimonGreenhill](#)
- [@SietzeN](#)

Submitted: 29 November 2023

Published: 22 October 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Introduction

Phylogenetic methods have become commonplace in historical linguistics research. However, much of the work is highly technical and not easily accessible to the typical classically-trained historical linguist. This paper aims to bridge the gap between linguistic and statistical research by introducing LinguiPhyR, an R package that provides a graphical user interface (GUI) to aid in the phylogenetic analysis of linguistic data. As such, very little computational background is required by the user. A linguist may simply upload a dataset, select optimization criteria, and visualize the phylogenies found by the search algorithm. Alternatively, one may upload trees of interest to be analyzed based on the dataset. Several tools for tree analysis are provided: users may examine what characters are responsible for particular splits in the tree, see the characters that are incompatible on the tree, annotate internal nodes of the tree with reconstructed states, and even see a relative chronology of state changes.

We note that, at present, our software focuses on parsimony-based tree estimation and analyses. We make this choice because such an approach is easily interpretable: the best tree is simply the tree that minimizes the number of state changes. This makes it easy for linguists to see the effect of each character in the dataset on tree search. However, one limitation of parsimony-based methods is that they are limited to searching for and analyzing tree topology: studies seeking to explore ancestral node dating (glottochronology) or branch lengths are better suited to using likelihood or Bayesian approaches. Future work will include the incorporation of other search algorithms and analytical methods into LinguiPhyR.

Statement of need

Given the recent explosion of linguistic phylogenetic datasets ([Heggarty et al., 2023](#); [Herce & Cathcart, 2024](#); [Jäger, 2018](#); [Tresoldi, 2023](#)), new tools for their analyses are called for. Many linguists want to perform parsimony analyses of their dataset, and our software makes it easy to do so with little effort. In this work, we provide an easy-to-use tool for phylogenetic analysis that emphasizes *interpretability*, allowing linguists to understand why trees are returned for a particular dataset *or* what evidence a new dataset has for existing trees suggested by the community. Currently, our software only allows parsimony-based analyses, a decision made because the go-to methods for phylogenetic analysis (Bayesian inference) requires reasonable mathematical maturity to understand, despite efforts to reduce barrier to entry. Nonetheless, future work will aim to bring other methods (such as distance-based, quartet-based, and Bayesian approaches) into the package.

The primary goals of LinguiPhyR are to

1. Make phylogenetics accessible by requiring *no* coding or writing of configuration files. While these are useful skills, giving linguists the option to spend their time analyzing trees in a GUI rather than writing code will facilitate analyses of phylogenetic inferences.

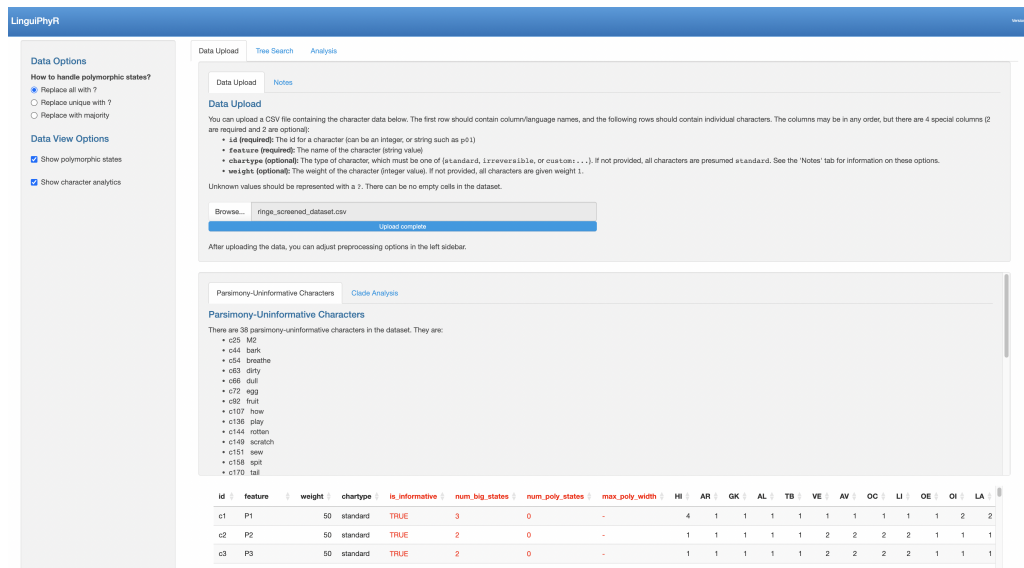


Figure 1: “Data Upload” page of LinguiPhyR.

2. Make it easy to find and visualize trees for a new linguistic dataset. One simply has to upload the dataset and select optimization criteria (or use the default settings). Trees are then displayed in the app and can be downloaded (either as images or as Nexus files) for inclusion in other work.
3. Provide a comprehensive set of (parsimony-based) analysis tools. These focus on the following questions: why are particular trees being suggested for the dataset? What evidence does a dataset contain for other trees of interest? What is the effect of particular coding decisions in the dataset on the understanding of a tree?

Our work is not the only attempt to make phylogenetic methods accessible and interpretable to linguists, nor is it the only GUI for this purpose. For example, PAUP* (Swofford, 2002) provides a GUI containing a comprehensive set of parsimony-based tools for phylogenetics, although it does require writing Nexus configuration files and is not specifically aimed at linguists. Tools specific to Bayesian linguistic phylogenetics include BEASTling (Maurits et al., 2017), which is a wrapper for BEAST (Bouckaert et al., 2014), and Traitlab (Kelly et al., 2023). A useful tutorial in R for linguistic phylogenetics is Goldstein (2020).

LinguPhyR: Linguistic Phylogenetic Analysis in R

The following sections describe each page of the app: Data Upload, Tree Search, and Analysis. Throughout the subsequent discussion, many terms familiar to historical linguists are used (e.g. *clade*, *cognate*, and *regular sound change*); we suggest Ringe & Eska (2013) for further reading. Similarly, we recommend Warnow (2017) for terms common in the phylogenetics literature, such as *character*, *polymorphism*, and *parsimony*.

Data Upload

The first page of the app is shown in Figure 1. The user first uploads a dataset of linguistic characters, which encode certain properties about languages that are likely to be relevant to the branching structure of the underlying tree. The characters should be uploaded as a CSV file. An example of the data format is shown below¹:

¹We provide the screened version of the Indo-European dataset of Ringe et al. (2002) in the correct format at the path `data/ringe_screened_dataset.csv` in the LinguPhyR Github repository.

Table 1: Example dataset specification, excerpted from the Indo-European dataset of Ringe et al. (2002).

id	feature	weight	char-type	HI	AR	GK	AL	TB	VE	AV	OC	LI	...
c1	P1	50	standard	4	1	1	1	1	1	1	1	1	...
c26	M3	50	standard	1	2	2	3	2	2	2	2	4	...
c50	bird	1	standard	1	2	3	4	5	6	6	7	8	...

Each row represents a character. The first four columns specify special character information: a unique character ID, the character name (“feature”), the weight of the character (optional, to be used in parsimony analyses), and the character type (which can be *standard*, *irreversible*, or *custom*, explained below). The remaining columns contain the character states for each attested language (i.e. the leaves of the tree).

Two languages should be given the same state for a character *if and only if* the languages’ realization of that character could be from a common genetic source (and not, for example, from borrowing). For lexical data, characters typically represent particular semantic slots (such as “bird” in the table above), and languages should share a state if their words for that meaning are cognate — that is, the words are derived from a common ancestor via regular sound change. However, if two languages share the same cognate due to detected borrowing or some other known non-genetic source, then the languages should be given different states for that character. Such cognate judgements are thus critically important to phylogenetic analysis; an abundance of phylogenetics literature discusses good methodology for doing designing characters (Heggarty, 2021; Nichols & Warnow, 2008; Ringe et al., 2002). Classical historical linguistics references are also helpful (Campbell, 2020; Ringe & Eska, 2013). Finally, our coding scheme is applicable to phonological, morphological, and structural/typological characters.

Each character may be declared “standard”, “irreversible”, or “custom”. Standard characters permit any change of state (e.g. from 0 to 1 or from 1 to 2) with uniform cost, and is generally appropriate for lexical characters. Irreversible characters are binary characters that may transition from 0 to 1 but not from 1 to 0; this is useful for representing phonological mergers, which are generally considered irreversible. Finally, custom characters allow the user to declare which state transitions are allowed and the cost of each.

Finally, our data format supports *multi-state* and *polymorphic* characters. Multi-state characters can take more than 2 states. While it is common to convert multi-state characters into a set of independent binary traits, we recommend leaving such characters in their underlying multi-state form unless the estimation algorithm explicitly requires binary characters; for further discussion, see Rexová et al. (2003), Nichols & Warnow (2008), and Heggarty (2021). Datasets may also contain polymorphic characters, which are those for which a language exhibits more than one state; for example, a language may manifest two cognate classes for the same semantic slot (Canby et al., 2024). Such examples are denoted by separating the states with a / (e.g. 1/2).

After uploading the data, the app presents some statistics:

- **Parsimony Uninformative Characters:** The characters that are not *parsimony informative* are displayed. These characters will have no effect on parsimony-based tree estimation because they can be fit equally well to any tree (see Warnow (2017) for a discussion). This is especially helpful to a linguist, who may not be thinking about the consequences of character codings to the parsimony algorithm when coding individual characters.
- **Character-level Statistics:** Various information about each character is displayed, such as the number of languages having polymorphic states for that character and whether or not the character is parsimony-informative. The dataset may be sorted by these metrics.

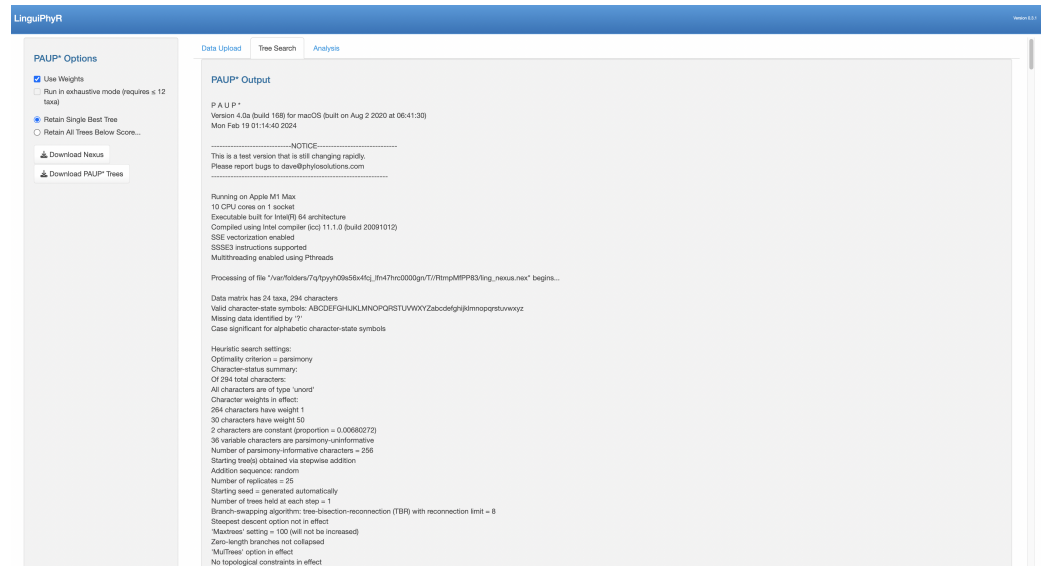


Figure 2: “Tree Search” page of LinguiPhyR.

- **Clade Analysis:** The user may select a subset of languages and analyze what characters provide support for such a clade (a clade is a subset of languages separated from all other languages by an edge in the tree). This is computed in the strictest sense: a character only supports a hypothetical clade if the languages in the clade all share the same state, and all other languages share a different state.²

Tree Search

On the second page of the app (shown in Figure 2), the user can conduct a search for the optimal tree(s) for the dataset. We use PAUP* (Swofford, 2002) to perform tree search, a well-established package in the biological community for running parsimony and other phylogenetic analyses. The user may specify various optimization criteria in the app without writing configuration files by hand, which is a big barrier to entry for many linguists. Nonetheless, users may download these configuration files from the app and modify them as needed.

Analysis

Finally, one may use the dataset to analyze trees, depicted in Figure 3. These trees can be either the result of a PAUP* tree search, or specific trees may be uploaded by the user. This latter option is especially helpful for determining the support that a dataset exhibits for various trees accepted by the community. Strict and majority consensus trees for the trees returned by PAUP* are displayed as well. The following analyses may be performed on each tree:

1. **Tree Score:** Each tree is scored using various metrics, including *parsimony*, *compatibility*, *total edge support*, and *minimum edge support*³. Hence, the trees can be ranked according to these options.

²Note that a clade on a particular tree may be supported by more than the characters that meet this condition. For example, if the dominant cognate class in a clade is lost by just one language in the clade, the character will still support the grouping if the removal of the edge separating the clade from all other languages would produce a less parsimonious tree. This can be examined in the “Analysis” page.

³The compatibility score is the total number of characters that evolve on the tree without homoplasy (see Warnow (2017) for further detail). To calculate total edge support and minimum edge support, we first calculate the number of characters that enforce, or support, each edge, based on whether or not the collapse of that edge would increase the parsimony score. Total edge support is the sum of these support values across all edges, and minimum edge support is the minimum of these values.

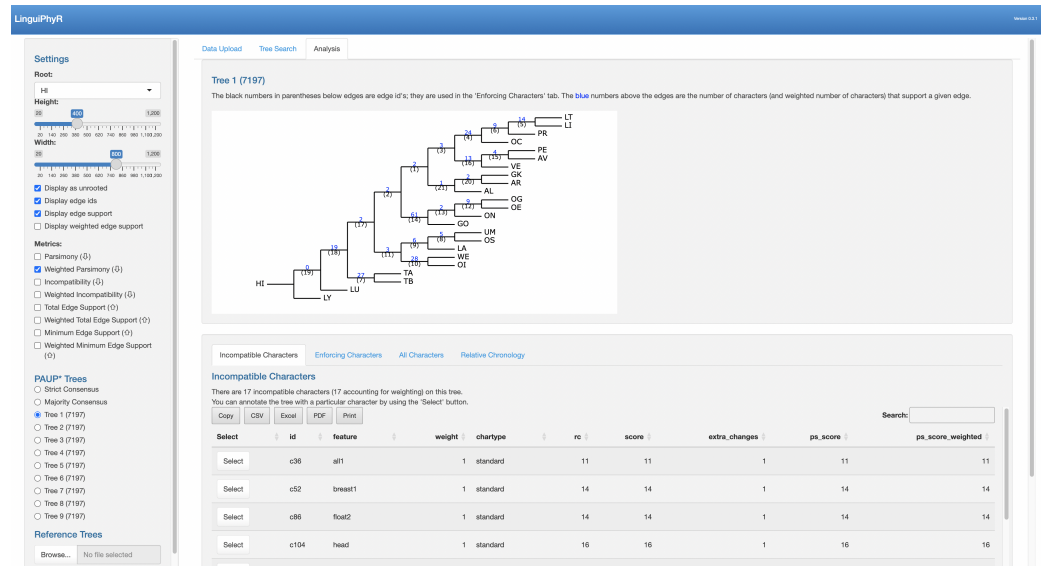


Figure 3: Analysis page of LinguiPhyR.

- Character annotations:** The most parsimonious annotation(s) of a character's states are displayed on the tree (including inferred states at internal nodes). This is convenient for studying a character's behavior and interpreting the consequences of particular character codings on phylogenetic estimation.
- Incompatible characters:** The characters not compatible on the tree are displayed. This is useful for considering the plausibility of various trees: if the set of incompatible characters seems unrealistic, a linguist may wish to discard the tree in favor of other options.
- Enforcing characters:** This reports the characters that enforce, or support, each edge. A character is deemed to support an edge if and only if the edge's collapse increases the tree's parsimony score. One can thus analyze evidence for and against various clades.
- Relative chronology:** This displays a relative chronology of state changes across characters, based on the ordering of the most parsimonious state transitions for each character on the edges from the root of the tree to a specified clade. This type of relative chronology may seem unusual to the typical historical linguist, but its results can be illuminating.

Conclusions

We present LinguiPhyR, a tool for analyzing phylogenetic datasets and trees via a graphical user interface. Even for experienced programmers, LinguiPhyR can quickly enable analysis on a new linguistic dataset. We especially emphasize (parsimony-based) interpretability by providing visualizations and tools to see the impact of certain coding decisions on tree estimation. Future work will incorporate other methods, such as likelihood-based or quartet approaches.

Acknowledgements

The author would like to acknowledge Thomas Olander, Matthew Scarborough, Simon Poulsen, Anders Jørgensen, Stefanos Baziotis, and Tandy Warnow, who have all provided invaluable feedback throughout the project.

This research was supported by the research project Connecting the Dots: Reconfiguring the Indo-European Family Tree (2019–2024), financed by the Independent Research Fund Denmark (project number 9037-00086B).

References

- Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.-H., Xie, D., Suchard, M. A., Rambaut, A., & Drummond, A. J. (2014). BEAST 2: A software platform for bayesian evolutionary analysis. *PLoS Computational Biology*, 10(4), e1003537. <https://doi.org/10.1371/journal.pcbi.1003537>
- Campbell, L. (2020). *Historical linguistics: An introduction*. Edinburgh University Press. <https://doi.org/10.1515/9781474463133>
- Canby, M. E., Evans, S. N., Ringe, D., & Warnow, T. (2024). Addressing polymorphism in linguistic phylogenetics. *Transactions of the Philological Society*, 122(2), 191–222. <https://doi.org/10.1111/1467-968X.12289>
- Goldstein, D. (2020). Indo-European phylogenetics with R: A tutorial introduction. *Indo-European Linguistics*, 8(1), 110–180. <https://doi.org/10.1163/22125892-20201000>
- Heggarty, P. (2021). Cognacy databases and phylogenetic research on Indo-European. *Annual Review of Linguistics*, 7, 371–394. <https://doi.org/10.1146/annurev-linguistics-011619-030507>
- Heggarty, P., Anderson, C., Scarborough, M., King, B., Bouckaert, R., Jocz, L., Kümmel, M. J., Jügel, T., Irlinger, B., Pooth, R., Liljegren, H., Strand, R. F., Haig, G., Macák, M., Kim, R. I., Anonby, E., Pronk, T., Belyaev, O., Dewey-Findell, T. K., ... Gray, R. D. (2023). Language trees with sampled ancestors support a hybrid model for the origin of Indo-European languages. *Science*, 381(6656), eabg0818. <https://doi.org/10.1126/science.abg0818>
- Herce, B., & Cathcart, C. A. (2024). Short vs long stem alternations in romance verbal inflection: The s-morpheme. *Transactions of the Philological Society*, n/a(n/a). <https://doi.org/10.1111/1467-968X.12271>
- Jäger, G. (2018). Global-scale phylogenetic linguistic inference from lexical resources. *Scientific Data*, 5(1), 1–16. <https://doi.org/10.1038/sdata.2018.189>
- Kelly, L. J., Nicholls, G. K., Ryder, R. J., & Welch, D. (2023). TraitLab: a Matlab package for fitting and simulating binary tree-like data. *arXiv Preprint arXiv:2308.09060*. <https://doi.org/10.48550/arXiv.2308.09060>
- Maurits, L., Forkel, R., Kaiping, G. A., & Atkinson, Q. D. (2017). BEASTling: A software tool for linguistic phylogenetics using BEAST 2. *PLoS One*, 12(8), e0180908. <https://doi.org/10.1371/journal.pone.0180908>
- Nichols, J., & Warnow, T. (2008). Tutorial on computational linguistic phylogeny. *Language and Linguistics Compass*, 2(5), 760–820. <https://doi.org/10.1111/j.1749-818X.2008.00082.x>
- Rexová, K., Frynta, D., & Zrzavý, J. (2003). Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics*, 19(2), 120–127. <https://doi.org/10.1111/j.1096-0031.2003.tb00299.x>
- Ringe, D., & Eska, J. (2013). *Historical Linguistics: Toward a Twenty-First Century Reintegration*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511980183>
- Ringe, D., Warnow, T., & Taylor, A. (2002). Indo-European and computational cladistics. *Transactions of the Philological Society*, 100(1), 59–129. <https://doi.org/10.1111/1467-968X.00091>
- Swofford, D. L. (2002). *Phylogenetic analysis using parsimony (PAUP*) 4.0*. Sinauer Associates: Sunderland, MA, USA. <https://doi.org/10.1111/j.0014-3820.2002.tb00191.x>
- Tresoldi, T. (2023). A Global Lexical Database (GLED) for Computational Historical Linguistics. *Journal of Open Humanities Data*. <https://doi.org/10.5334/johd.96>
- Warnow, T. (2017). *Computational phylogenetics: An introduction to designing meth-*

ods for phylogeny estimation. Cambridge University Press. <https://doi.org/10.1017/9781316882313>