The Journal of Open Source Software

# LinguiPhyR: A Package for Linguistic Phylogenetic Analysis in R

**Marc E. Canby**[1]

**1** University of Illinois at Urbana-Champaign, USA

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Andrew Stewart

**Reviewers:**

- @fauxneticien
- @SimonGreenhill

**Submitted:** 29 November 2023
**Published:** unpublished

## Introduction

Phylogenetic methods have become commonplace in historical linguistics research. However, many current research activities in the area are undertaken by statisticians rather than linguists, which is largely (and understandably) due to the highly mathematical and computational nature of the work. 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 or statistical expertise 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 given the dataset. Several tools for tree analysis are provided: a user 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 by linguists: the best tree is simply the tree that minimizes the number of state changes. Many popular methods for phylogenetic estimation, such as maximum likelihood and Bayesian inference, are less easily interpretable, as they are highly parametric and rely on a likelihood function that may be unrealistic or obscure to linguists. Parsimony analyses, on the other hand, make it easy to see the effect of each character in the dataset on tree search. Other concerns about fully parametric approaches have been raised as well, such as the suggestion that non-parametric methods like parsimony are more accurate (Barbançon et al., 2013; Holmes, 2003; Nichols & Warnow, 2008). Nonetheless, future work will include the incorporation of other search algorithms and analytical methods into LinguiPhyR.

## Statement of need

Given the recent explosion of new linguistic phylogenetic datasets (Heggarty et al., 2023; Herce Calleja & Cathcart, 2023; 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.

The primary goals of LinguiPhyR are to

1. Make phylogenetics accessible to linguists by requiring *no* coding or writing of configuration files. While these are useful skills, we believe phylogenetics can only be useful to historical linguistics if considerable analysis is given to the results of phylogenetic

algorithms by linguists. Over-emphasis on technical ability often hinders this work.

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 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 proposed by the community? 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).

# LinguiPhyR: 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 further discussion on language common in the phylogenetics literature, such as *character*, *polymorphism*, and *parsimony*.

## Data Upload

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 specified in a spreadsheet and uploaded as a CSV file. An example of the data format is shown below:

**Table 1:** Example dataset specification, excerpted from the screened 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*). 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 a linguist can demonstrate that two languages share the same cognate due to borrowing or some other non-genetic source, then the languages should be given different states for that character.

Such cognate judgements are critically important to the results of phylogenetic estimation. A haphazard or automated data representation will not yield meaningful trees; hence, it is important to have well-trained linguists judge relevant material and select characters that actually represent potentially shared innovations. An abundance of literature discusses good methodology for doing this (Nichols & Warnow, 2008; Ringe et al., 2002).

Our data format also supports *polymorphic* character states: these are instances where a language exhibits more than one state for a character. In the context of lexical data, this would mean that a language manifests two cognate classes for the same semantic slot. Such examples are denoted by separating the states with a / (e.g. *1/2*) in the dataset.

Finally, we note that our software permits *multi-state* characters, not just binary traits. Binary traits are particularly common in likelihood-based phylogenetic estimation, because most likelihood models require a pre-specified state space (e.g. $0$ and $1$). Non-parametric methods like parsimony do not have this assumption, and, in fact, it is not advisable to treat multi-state characters as a set of binary traits because the estimation algorithms consider the traits independent (when they are not) (Nichols & Warnow, 2008; Rexová et al., 2003; Warnow, 2017). For example, a lexical character denoting "bird" may have states $1$, $2$, and $3$, each representing a different cognate class observed in attested languages. Treating this as binary would create three traits, referring to whether or not the languages exhibit each of these cognate classes in the "bird" meaning. Unless there is a reason to make this binary conversion (e.g. because it is necessary to run likelihood algorithms), we suggest to leave the data in the underlying multi-state form.

The app then presents some statistics about the dataset, and one can perform some simple analyses:

- **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. This thus allows a linguist to carefully consider coding choices.
- **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 (among others). The dataset may be sorted by these metrics.
- **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.

## Tree Search

Then, the user may proceed to the second page of the app, which conducts a search for the optimal tree(s) given 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 having to write configuration files by hand, which is a big barrier to entry for many linguists.

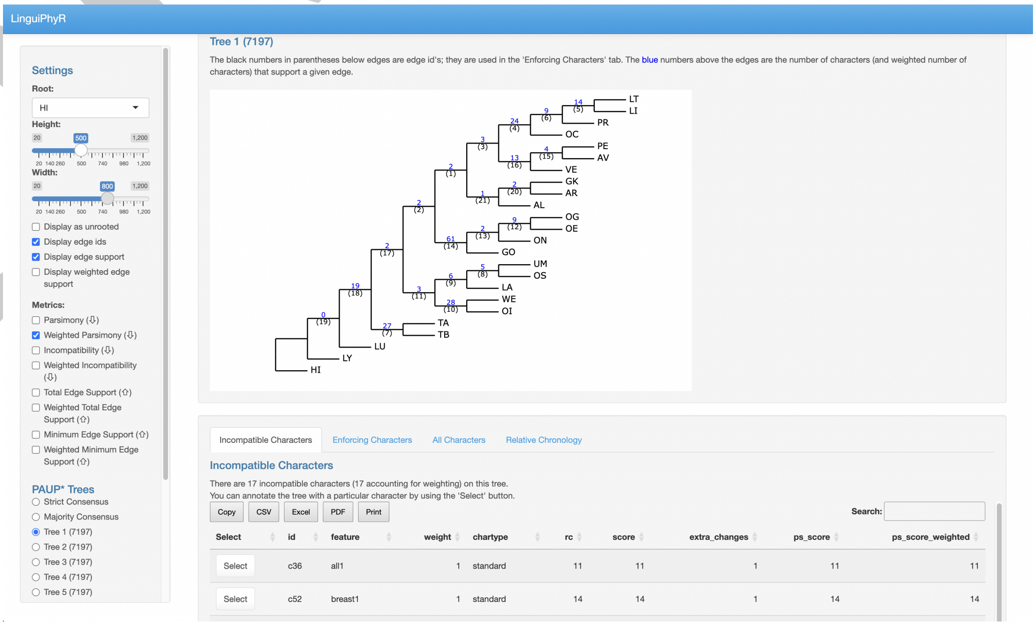
Nonetheless, users may download these configuration files from the app and modify them as needed.

## Analysis

Finally, one may use the dataset to analyze trees. These trees can be either the result of a PAUP* tree search, or specific trees of interest 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 primary analyses that can be performed on a tree are the following:

1. **Tree Score:** Each tree is scored using various metrics, including *parsimony*, *compatibility*, *total edge support*, and *minimum edge support*. Hence, the trees can be ranked according to these options.
2. **Character annotations:** The user may select any character and see the most parsimonious annotation(s) of that character's states across the tree (including reconstructed states at internal nodes). This is convenient for studying a character's behavior, and can help a linguist interpret the consequences of particular character codings on phylogeny estimation.
3. **Incompatible characters:** This reports the characters that are not compatible on a tree. This is useful for considering how plausible various trees are: if the set of characters that a tree is not compatible on seems unrealistic, a linguist may wish to discard the tree in favor of other options.
4. **Enforcing characters:** This reports the characters that enforce, or support, each edge in the tree. Thus, one may analyze evidence for and against various clades.
5. **Relative chronology:** This reports a relative chronology of state changes *across* characters. This is calculated by first determining the most parsimonious state transitions for each character, and then ordering these transitions based on the edges they occur on 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.

Figure 1 depicts an example tree analysis in LinguiPhyR.



LinguiPhyR

Settings

Root: HI

Height: 500 (20 – 1,200)

Width: 800 (20 – 1,200)

☐ Display as unrooted
☑ Display edge ids
☑ Display edge support
☐ Display weighted edge support

Metrics:

☐ Parsimony (⇩)
☑ Weighted Parsimony (⇩)
☐ Incompatibility (⇩)
☐ Weighted Incompatibility (⇩)
☐ Total Edge Support (⇧)
☐ Weighted Total Edge Support (⇧)
☐ Minimum Edge Support (⇧)
☐ Weighted Minimum Edge Support (⇧)

PAUP* Trees

○ Strict Consensus
○ Majority Consensus
● Tree 1 (7197)
○ Tree 2 (7197)
○ Tree 3 (7197)
○ Tree 4 (7197)
○ Tree 5 (7197)

Tree 1 (7197)

The black numbers in parentheses below edges are edge id's; they are used in the 'Enforcing Characters' tab. The blue numbers above the edges are the number of characters (and weighted number of characters) that support a given edge.

Incompatible Characters

There are 17 incompatible characters (17 accounting for weighting) on this tree.
You can annotate the tree with a particular character by using the 'Select' button.

| Select | id | feature | weight | chartype | rc | score | extra_changes | ps_score | ps_score_weighted |
|---|---|---|---|---|---|---|---|---|---|
| Select | c36 | all1 | 1 | standard | 11 | 11 | 1 | 11 | 11 |
| Select | c52 | breast1 | 1 | standard | 14 | 14 | 1 | 14 | 14 |

**Figure 1:** Analysis page of LinguiphyR.

## Conclusions

We present LinguiPhyR, a useful tool for analyzing phylogenetic datasets and trees without the need to code. Even for experienced programmers, LinguiPhyR can quickly enable analysis on a new linguistic dataset or provide a starting place for finding new trees. In our app, we especially emphasize parsimony-based interpretability by providing useful visualizations and tools to see the impact of certain coding decisions on tree estimation. Future work will include the incorporation of other inference methods (such as distance-based and quartet approaches), as well as more advanced analytical tools, such as bootstrap analysis.

## 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

Barbançon, F., Evans, S. N., Nakhleh, L., Ringe, D., & Warnow, T. (2013). An experimental study comparing linguistic phylogenetic reconstruction methods. *Diachronica*, *30*(2), 143–170.

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.

Goldstein, D. (2020). Indo-european phylogenetics with r: A tutorial introduction. *Indo-European Linguistics*, *8*(1), 110–180. https://doi.org/https://doi.org/10.1163/22125892-20201000

Heggarty, P., Anderson, C., Scarborough, M., King, B., Bouckaert, R., Jocz, L., Kümmel, M. J., Jügel, T., Irslinger, 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 Calleja, B., & Cathcart, C. (2023). Short vs long stem alternations in romance verbal inflection: The s-morphome. *Transactions of the Philological Society*, Epub–ahead.

Holmes, S. (2003). Statistics for phylogenetic trees. *Theoretical Population Biology*, *63*(1), 17–32.

Jäger, G. (2018). Global-scale phylogenetic linguistic inference from lexical resources. *Scientific Data*, *5*(1), 1–16.

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*.

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.

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., & Zrzavỳ, J. (2003). Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics*, *19*(2), 120–127.

Ringe, D., & Eska, J. (2013). *Historical linguistics: Toward a twenty-first century reintegration*. Cambridge University Press.

Ringe, D., Warnow, T., & Taylor, A. (2002). Indo-European and computational cladistics. *Transactions of the Philological Society*, *100*(1), 59–129.

Swofford, D. L. (2002). *Phylogenetic analysis using parsimony (PAUP*) 4.0*. Sinauer Associates: Sunderland, MA, USA.

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 methods for phylogeny estimation*. Cambridge University Press.