The Journal of Open Source Software

JPhyloRef: a tool for testing and resolving phyloreferences

Gaurav Vaidya[1, 2], **Nico Cellinese**[2], **and Hilmar Lapp**[3]

1 Renaissance Computing Institute, University of North Carolina, Chapel Hill, NC, USA **2** Florida Museum of Natural History, University of Florida, Gainesville, FL, USA **3** Center for Genomic and Computational Biology, Duke University, Durham, NC, USA

DOI: 10.21105/joss.03374

Software

- Review
- Repository
- Archive

Editor: Kelly Rowland
Reviewers:

- @iimog
- @7omasz

Submitted: 21 April 2021
Published: 15 June 2021

Summary

JPhyloRef is a command line tool as well as a web service for reasoning with ontologies containing logical definitions of phylogenetic clade definitions, called phyloreferences, and their accompanying reference phylogenetic trees. It has two main goals. The primary one is to facilitate automated testing that the semantics of the logical definitions imply ("resolve to") the correct nodes in the reference tree as clade ancestors. This is key in supporting quality control for the digitization of phylogenetic clade definitions from natural language text to a structured machine-interpretable representation. It also verifies that one of the theoretical foundational premises of phyloreferences, computational reproducibility, holds in practice. The secondary goal is to enable integration with external tools that need to obtain the clade ancestor node(s) resulting from a given ontology of phyloreferences and reference tree(s). When run as part of an automated testing workflow, JPhyloRef reports test results in the cross-platform Test Anything Protocol (TAP) format. When used to find clade ancestor nodes implied by logical clade definitions, results are returned as a JSON object. JPhyloRef uses the OWL API reference library for reading Web Ontology Language (OWL) ontologies, and for the actual ontology reasoning step it uses an external and configurable OWL reasoner.

Background and Overview

In evolutionary biology, groups of organisms consisting of an ancestor and all of its descendants ("clades") are a fundamental unit for understanding evolution and describing biodiversity (Queiroz, 2007). Phylogenetic clade definitions define clades based on shared ancestry, providing the theoretical foundation for the semantics of taxon concepts to be defined and reproducibly resolved within a hypothesis of evolutionary relationships, i.e., a phylogeny (Queiroz, 1992; Queiroz & Gauthier, 1992, 1990). We have proposed a mechanism, called Phyloreferencing, for representing phylogenetic clade definitions as structured data with fully machine-processable semantics, using the Web Ontology Language (OWL) (W3C OWL Working Group, 2012). We refer to such machine-interpretable clade definitions as "phyloreferences" (Cellinese et al., 2021).

Because phyloreferences have an OWL ontology representation in the form of logically defined classes, combining them with an OWL representation of a phylogenetic tree allows an OWL reasoner to infer, based on their logical definitions, which nodes in the phylogeny, if any, instantiate each phyloreference. We refer to this logical inference as "resolving a phyloreference." Phylogenetic clade definitions are normally created and published in the form of natural language text, and in the context of a phylogenetic hypothesis, the so-called reference phylogeny. When digitized, usually through manual curation, to a structured, machine-interpretable representation, which we call a phyloreference, a key question in quality control of both the digitization product and the ontologies and logical expression algorithm it uses arises from one of the founding premises of phyloreferences: given that phyloreferences make the semantics of their represented clade machine-interpretable, can automated computational testing check whether a phyloreference resolves, and only resolves to the node in the reference phylogeny which the original authors of the published phylogenetic clade definition designate as the expected node.

To achieve this, we built JPhyloRef, a command-line tool in Java. When run with the `test` command with an OWL ontology containing phyloreferences as well as reference phylogenies as input, JPhyloRef identifies all the phyloreferences within this ontology, compares for each one the inferred phylogeny node(s) to the expected one, and determines success (inferred node identical to expected) or failure (no inferred node, or inferred node different from expected). To better facilitate use in automated continuous integration testing, JPhyloRef also allows for phyloreferences to be marked as draft (resulting in an "incomplete" test status), or as to be skipped (by no nodes having been annotated as an expected resolution of a particular phyloreference). These test results are reported in the Test Anything Protocol (TAP) format (https://testanything.org/), a cross-platform format for reporting test results. JPhyloRef will also return an exit code indicating success (0), failure (the number of failed phyloreferences as a positive integer) or an ontology containing no phyloreferences (-1, interpreted as 255 by POSIX-compatible operating systems).

Internally, JPhyloRef relies on an external OWL reasoner (which can be configured) to perform the actual OWL inferences, with which it communicates using the OWL API 4.2.7 (Horridge & Bechhofer, 2011). We strongly recommend to use ELK (Kazakov et al., 2014), a reasoner for the OWL-EL profile, which is sufficient for the inferences necessary for phyloreference resolution. In our experience, reasoners for more expressive profiles, such as OWL-DL reasoners, do not perform efficiently enough even for modestly sized ontologies of phyloreferences.

To support integrating Phyloreferencing with other tools, JPhyloRef includes two additional commands: `resolve` and `webserver`. Using the `resolve` command with an input of an OWL ontology in any of the supported formats will result in a JSON object whose keys are the IRIs of all the phyloreferences in the ontology that resolved to any nodes, and whose values are lists of the IRIs of the phylogeny nodes that they resolved to. Using the `webserver` command starts an HTTP server on the hostname and TCP port specified in the command line arguments. This webserver provides a POST endpoint at `/reason` that can be used to submit OWL ontologies in JSON-LD format for reasoning, which returns a JSON object in the same format as produced by the `resolve` command. Neither of these commands return test statuses for phyloreferences.

JPhyloRef is currently in active use for integrating the digitization of phylogenetic clade definitions in the form of phyloreferences into an automated testing framework, as well as for resolving phyloreferences on the command line. For example, we use it in the continuous integration testing infrastructure of the phyx.js JavaScript library (https://github.com/phyloref/phyx.js/actions), where the output from the `test` command is interpreted by a JavaScript-based TAP library to ensure that phyloreferences in example OWL ontology files resolve as expected. We also use the `webserver` command of JPhyloRef as a backend for the Klados phyloreference curation software (https://github.com/phyloref/klados), allowing the curator to submit phyloreferences created from user input or a publication, along with associated phylogenies, to a backend server running JPhyloRef for resolution. The source code for JPhyloRef has been archived to Zenodo (Vaidya & Lapp, 2021).

Acknowledgements

This work was funded by the US National Science Foundation through collaborative grants DBI-1458484 and DBI-1458604 to Hilmar Lapp (Duke University) and Nico Cellinese (University of Florida), respectively.

References

Cellinese, N., Conix, S., & Lapp, H. (2021). Phyloreferences: Tree-Native, reproducible, and Machine-Interpretable taxon concepts. *EcoEvoRxiv*. https://doi.org/10.32942/osf.io/57yjs

Horridge, M., & Bechhofer, S. (2011). The OWL API: A Java API for OWL ontologies. *Semant. Web*, *2*(1), 11–21. https://doi.org/10.3233/SW-2011-0025

Kazakov, Y., Krötzsch, M., & Simančík, F. (2014). The incredible ELK: From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reasoning*, *53*(1), 1–61. https://doi.org/10.1007/s10817-013-9296-3

Queiroz, K. de. (2007). Toward an integrated system of clade names. *Syst. Biol.*, *56*(6), 956–974. https://doi.org/10.1080/10635150701656378

Queiroz, K. de. (1992). Phylogenetic definitions and taxonomic philosophy. *Biol. Philos.*, *7*(3), 295–313. https://doi.org/10.1007/BF00129972

Queiroz, K. de, & Gauthier, J. (1992). Phylogenetic taxonomy. *Annu. Rev. Ecol. Syst.*, *23*(1), 449–480. https://doi.org/10.1146/annurev.es.23.110192.002313

Queiroz, K. de, & Gauthier, J. (1990). Phylogeny as a central principle in taxonomy: Phylogenetic definitions of taxon names. *Syst. Zool.*, *39*(4), 307. https://doi.org/10.2307/2992353

Vaidya, G., & Lapp, H. (2021). *Phyloref/jphyloref: JPhyloRef v1.0.0* (Version v1.0.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4697965

W3C OWL Working Group. (2012). *OWL 2 web ontology language document overview*. World Wide Web Consortium (W3C). https://www.w3.org/TR/owl2-overview/