

AN R PACKAGE TO INTERACT WITH THE OPEN TREE OF LIFE

rotl an R package to interact with the Open Tree of Life Data

François Michonneau^{1,2}, Joseph Brown³, David Winter⁴

¹ *Whitney Laboratory for Marine Sciences, University of Florida, St. Augustine, FL, USA*

² *Florida Museum of Natural History, University of Florida, Gainesville, FL, USA*

³ *University of Michigan, Ann Arbor, MI, USA*

⁴ *Arizona State University, Tempe, AZ, USA*

Corresponding author: François Michonneau, Division of Invertebrate Zoology, Florida Museum

of Natural History, Gainesville, FL 32611-7800, USA; E-mail: francois.michonneau@gmail.com

Abstract.—

1. While phylogenies have been getting easier to build, it has been difficult to re-use, combine, and synthesize the information they provide because published trees are often only available as image files, and taxonomic information is not standardized across studies.
2. The Open Tree of Life (OTL) project addresses these issues by providing a digital tree that encompasses all organisms, built by combining taxonomic information and published phylogenies. The project also provides tools and services to query and download parts of this synthetic tree, as well as the source data used to build it. Here, we present `rotl`, an R package to search and download data from the Open Tree of Life directly in R.

- 20 3. `rotl` uses common data structures allowing researchers to take advantage of the rich set of tools and methods that are available in R to manipulate, analyze, and visualize phylogenies. Here, and in the vignettes accompanying the package, we demonstrate how `rotl` can be used with other R packages to analyze biodiversity data.
- 25 4. As phylogenies are being used in a growing number of applications, `rotl` facilitates access to phylogenetic data, and allows their integration with statistical methods and data sources available in R.

Advances in sequencing and computing technologies have lead to a revolution in systematic biology. The ability to routinely generate molecular datasets from any extant organism has allowed researchers to resolve long-standing taxonomic disputes and estimate phylogenies for previously

30 understudied groups. In parallel, the ease with which phylogenies can be estimated has spurred the development of new phylogenetic comparative methods. These methods allow researchers to explore fundamental questions about the origin of biodiversity including the evolution of morphological and ecological traits, the spatio-temporal variation in speciation rates, or both (O'Meara 2012; Pennell and Harmon 2013).

35 Ideally, the ever increasing number of published phylogenies would contribute to a synthesis of phylogenetic knowledge, ultimately leading to a better understanding of the history of life while at the same time providing high-quality phylogenetic information for use in comparative analyses. However, in practice, synthesizing phylogenetic data is a difficult task. Phylogenetic information is largely scattered, often only available as image files within publications, and the lack of

40 standardization to store and represent phylogenetic data makes it difficult for researchers to access, synthesize, and integrate this information into their own research (Stoltzfus et al. 2012; Drew et al.

2013; Magee et al. 2014; but see Cranston et al. 2014 for suggestions of best practices).

The Open Tree of Life (OTL) project aims at assembling and synthesizing our current understanding of phylogenetic relationships across all organisms on Earth while providing tools and services that facilitate access to this information (Hinchliff et al. 2015). OTL combines taxonomic information that serves as the backbone for the phylogenetic relationships, and published phylogenies to elucidate relationships among taxa. This combination of information is used to structure the comprehensive synthetic tree. Studies can be contributed to the synthetic tree through a curator interface (<https://tree.opentreeoflife.org/curator>), allowing the synthetic tree to be continuously updated as relationships are elucidated or reevaluated. The current draft of the OTL synthetic tree contains 2.3 million tips. Beyond obvious applications across the life sciences to explore questions in evolution, biodiversity, and conservation, the resources OTL provides are useful for education and outreach (e.g., illustrating course material, or developing outreach activities to explore relationships among species).

The R programming language is a popular tool for phylogenetics and comparative analysis. The R packages *ape* (Paradis et al. 2004), *phyllobase* (Bolker et al. 2015), *phangorn* (Schliep 2011) and *RNeXML* (Boettiger et al. 2015b) each provide functions to import and manipulate trees within R and save the results in standard data formats. Additional packages including *phytools* (Revell 2012), *geiger* (Pennell et al. 2014), and *ggtree* (Yu et al. 2015), allow users to analyze and visualize data in a phylogenetic context (see <https://cran.r-project.org/web/views/Phylogenetics.html> for a comprehensive list of phylogenetics packages in R). In addition to packages for phylogenetic and comparative analyses, a growing number of R packages allows users to query and access data from the web (e.g. *rFISHBASE* (Boettiger et al. 2012), *rAvis* (Varela et al. 2014), and *paleobioDB* (Varela et al. 2015)), such that data associated with taxa in a given phylogeny can be obtained directly in R.

In ecology, the development of the field of community phylogenetics (Webb et al. 2002) has created a need for researchers to have access to the evolutionary relationships of species making up communities. The relative contributions of the role of the environment (e.g., habitat filtering), and

of biotic interactions (e.g., competitive exclusion), are inferred from the distribution of taxa on a
70 phylogeny composed from species occurring at larger spatial scale. R packages and other software
have been developed to generate phylogenies from species lists using taxonomic information or
DNA sequences (e.g., Webb et al. 2008; Pearse and Purvis 2013). These phylogenies can then be used
for community phylogenetics analyses (e.g., Pearse et al. 2015; Kembel et al. 2010), but they are often
incomplete or not resolved enough. As OTL becomes more comprehensive, and its taxonomic
75 resolution increases, it could become a valuable resource for ecologists seeking to use phylogenetic
information in their research.

These packages, combined with the language's support for literate programming (Knuth
1984; Xie 2015), make R a comprehensive platform for reproducible research in phylogenetics and
comparative biology, as they allow a complete record of the steps taken in gathering, processing, and
80 analyzing a given data set to be produced.

Here we present `rotl`, an R package that allows users to download phylogenetic and
taxonomic data from the OTL directly in R. `rotl` takes advantage of OTL's Application
Programming Interfaces (APIs) to access sub-trees from the synthetic Open Tree, as well as the
published source trees that contribute to the synthesis. By providing direct access to high quality
85 phylogenetic data in R, `rotl` fills a key gap in typical comparative analysis workflows, and extends
the degree to which R supports reproducible research in phylogenetics and comparative biology.

API services provided by OTL

The OTL project provides four resources that serve data to users through the APIs:

1. The *taxonomy* used as the backbone of the tree, the Open Tree Taxonomy (OTT);
- 90 2. The *studies* and their associated trees, some of which are chosen by curators to assemble the
synthetic tree;
3. A *taxonomic name resolution service* (TNRS) used to match taxon names to the Open Tree
Taxonomy identifiers;

4. The *synthetic tree* itself, the ‘Open Tree’.

95 `rotl` gives users access to the endpoints provided by version 2 of the APIs, and other
versions of the APIs can be selected by the user as they become available (version 3 is scheduled to be
released at the end of March 2016). Two other end-points are available in version 2 of the API: *graph*
of life and *collections*. Data from the *graph of life* endpoint are not intended for typical users, and will
be removed in future releases. The *collection* endpoint is not fully functional, and no functions are
100 available in the current release of `rotl`. Future versions will make them available when the API
implements them.

Phylogenetic trees served by the API can be imported directly into R’s memory and are
represented using the `ape` (Paradis et al. 2004) tree structure (objects of class `phylo`), or can be
written to files in the Newick, NEXUS (Maddison et al. 1997), or NeXML (Vos et al. 2012) file
105 formats. This allows researchers to use these trees either directly with other R packages, or to be
imported in other programs that make use of phylogenetic tree files.

Currently, the synthetic tree does not have any branch lengths associated with it, therefore
parametric comparative methods cannot be used directly on the subtrees returned by OTL (although
the OTL treestore contains the raw published source phylogenies, complete with branch lengths and
110 node annotations; see below). However, resources and methods are being developed to add branch
lengths to these topological subtrees (e.g., Ksepka et al. 2015) or use topological trees to identify
phylogenetically equivalent species to increase overlap between chronograms and species trait data
(Pennell et al. 2015). Without branch lengths, these subtrees are nonetheless useful to illustrate
relationships among species, or to map traits on a phylogeny.

Technical information about ROTL

115 Phylogenetic information retrieved from OTL is converted into `phylo` objects by `rotl`
using the NEXUS Class Library (NCL, Lewis 2003) as implemented in the `rncl` package
(<https://cran.r-project.org/package=rncl>). Using NCL provides robust and efficient

parsing of large trees that may contain singleton nodes labeled with taxonomic information (i.e., a
120 monotypic taxon). Singleton nodes are collapsed after the tree has been parsed, making the resulting
phylo object compatible with all functions from the ape package.

The package is well-documented, and includes three package vignettes (documents that
demonstrate the use of the package and contain executable R code). There is also an extensive
test-suite that covers both the internal functions that rotl uses to connect to OTL, and public
125 functions that users apply to access and process data.

Demonstrations

Getting relationships from a list of taxa

Before a researcher can use the Open Tree to retrieve relationships among a set of taxa, they
first need to match the taxon names in their dataset with records in the Open Tree Taxonomy
130 (OTT). OTL's taxonomic names resolution service (TNRS) combines information from multiple
services (e.g., National Center for Biotechnology Information [NCBI], World Register for Marine
Species [WoRMS], Global Biodiversity Information Facility [GBIF]) and allows users to search for
taxon names, and retrieve identifiers for each matching taxon. We demonstrate the use of the TNRS
within rotl by searching for taxonomic records associated with several model organisms.

```
taxa <- tnrs_match_names(names = c("Escherichia coli",
```

```
"Chlamydomonas reinhardtii",  
"Drosophila melanogaster",  
"Arabidopsis thaliana",  
"Rattus norvegicus",  
"Mus musculus",  
"Cavia porcellus",  
"Xenopus laevis",  
"Saccharomyces cerevisiae",  
"Danio rerio"))
```

135 The function `tnrs_match_names` returns a data frame that lists the Open Tree identifiers as well as other information to help users ensure that the taxa matched are the correct ones. Here, there is no ambiguity in the taxa matched; however, as OTT includes taxa from bacteria, plants, and animals that are regulated by different nomenclatural codes (ICNP, ICN, and ICZN, respectively), both OTL and `rot1` provide tools to deal with names that may represent valid taxa in more than one
140 code. The argument `context_name` can be used to limit potential matches to a taxonomic group such as “Animals” (see the function `tnrs_contexts` for a complete list of possible options). When this strategy cannot be used (as in the present example, where the tree encompasses multiple domains), the function `inspect` lists alternative matches for a taxon name, and `update` replaces it in the results. An example of this approach is provided in the vignette “How to use `rot1`?” that
145 accompanies the package.

By default, approximate matching is enabled when attempting to match taxonomic names to their OTT identifiers. Additionally, taxonomic synonyms are included in OTT, allowing researchers to match correct identifiers for taxon names that might include misspellings or synonyms. These features will facilitate the tedious data cleaning process often needed when mapping taxon names. In
150 the example provided, both *Escherichia coli* and *Saccharomyces cerevisiae* are misspelled, but the OTL TNRS finds the correct match for these taxa.

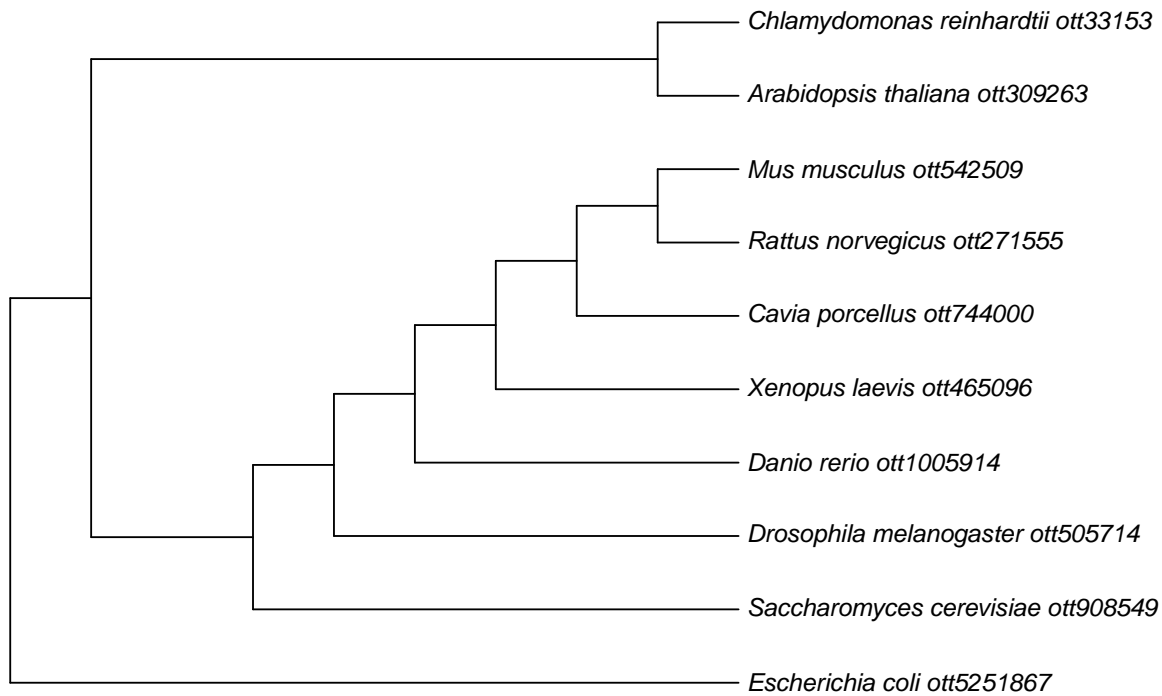


Figure 1: The phylogenetic tree returned by OTL for the list of model species used as an example.

Now that the taxon names are matched to the Open Tree identifiers, we can pass them to the function `tol_induced_subtree` to retrieve the relationships among these taxa. In turn, the tree can be plotted directly as it is returned as phylo object (Figure 1).

```
tree <- tol_induced_subtree(ott_ids = taxa[["ott_id"]])
plot(tree, cex = .8, label.offset = .1, no.margin = TRUE)
```

Getting trees from studies

`rotl` can also be used to retrieve trees accompanying studies that have been submitted through the curator interface, and identify the trees that contribute to the synthetic tree. As of March 2016, the Open Tree of Life project stores 7755 trees from 3399 studies (each having between 0 and 61 trees), and 477 of these trees are used to assemble the synthetic tree. These trees constitute a useful resource to reproduce or expand on a previously published analysis, or to explore how the elucidation of relationships within a clade has changed through time.

Criteria that can be used to search for studies or their associated trees are available through the output of the function `studies_properties`. The meaning of these properties is described at: <https://github.com/OpenTreeOfLife/phylesystem-api/wiki/NexSON>. Typically, users will want to search for studies or trees based on taxon names (or their OTT identifiers), but other criteria such as the title of the publication can be used. Here we demonstrate how to look for and retrieve trees for studies focusing on the family Felidae (Figure 2).

```
cat_studies <- studies_find_studies(property = "ot:focalCladeOTTTaxonName",
                                   value = "Felidae", exact = TRUE)

cat_studies

##   study_ids n_trees tree_ids candidate study_year
## 1  pg_1981      1 tree4052  tree4052      2006
##
##                                     title
## 1 The late Miocene radiation of modern Felidae: a genetic assessment
##
##                               study_doi
## 1 http://dx.doi.org/10.1126/science.1122277
```

Currently only one study focused on this family is available from OTL, and a single tree is associated with it. We can then retrieve the study and tree identifiers, and pass them to the function `get_study_tree` to have the tree in memory:

```
cat_tree <- get_study_tree(study_id = cat_studies[["study_ids"]][1],
                          tree_id = cat_studies[["tree_ids"]][1])

cat_tree

##
```

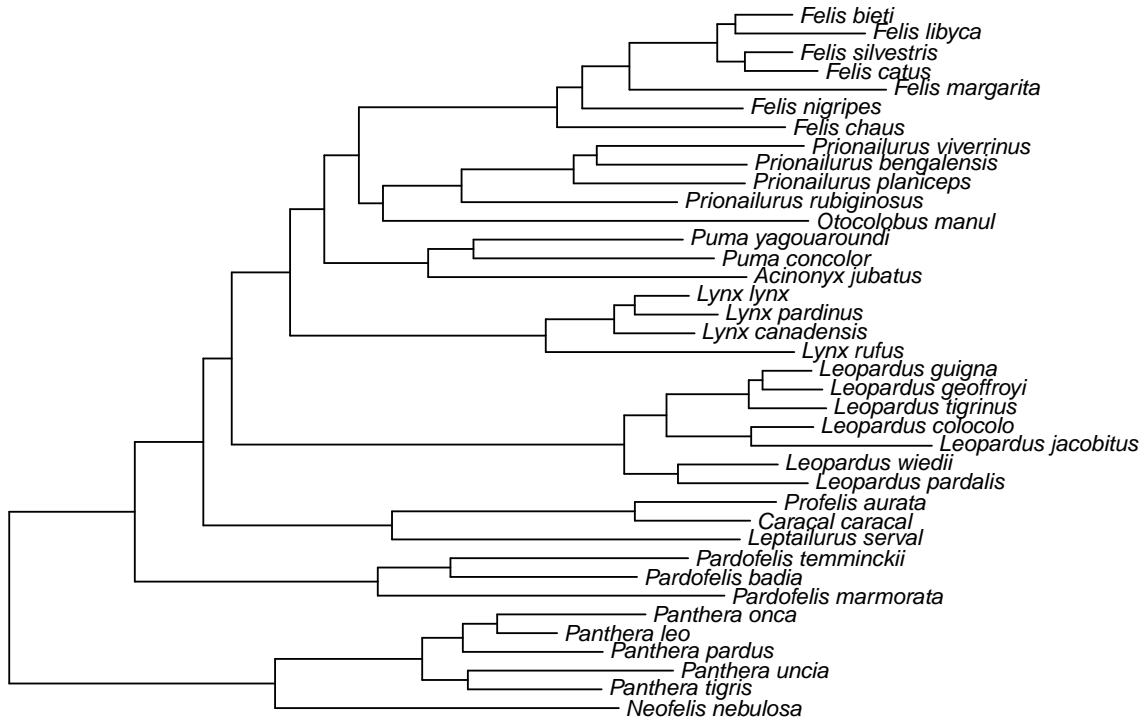


Figure 2: Phylogeny of the Felidae published in Johnson et al. 2006 and retrieved from OTL using rotl.

```
## Phylogenetic tree with 38 tips and 37 internal nodes.
##
## Tip labels:
##  Neofelis_nebulosa, Panthera_tigris, Panthera_uncia, Panthera_pardus, ...
##
## Rooted; includes branch lengths.
```

When more than one tree is available for a given study, the function `list_trees` returns a list containing the tree identifiers for each study. Alternatively, the function `get_study` returns all the trees (by default as phylo objects) associated with a particular study. Metadata about the study (e.g., citation information, information about the curator for the study, and other technical

175 information regarding the import of this study) can be obtained using the function `get_study_meta`.

How does `rotl` fit into the R package ecosystem?

In recent years, R has become an essential part of the toolbox of many researchers in evolutionary biology and ecology. R greatly facilitates the analysis of large datasets, and allows
180 researchers to combine methods in novel ways because many methods for comparative analyses are implemented, and because it is a relatively easy to use programming language. Additionally, as more data are made available online and accessible using APIs, several packages have been developed to interact and download these datasets directly in R, thereby enabling direct and reproducible analyses. Notably, the organization rOpenSci (<https://ropensci.org>) has fostered a community of
185 researchers who develop tools and methods to facilitate the use of open data as well broaden the adoption of open science practices in general (Boettiger et al. 2015a). For instance, the rOpenSci-developed package TreeBase (Boettiger and Temple Lang 2012) allows users to access phylogenies stored in treeBASE (www.treebase.org). `rotl` contributes to this initiative, and greatly extends the number of taxa for which phylogenetic data can be retrieved within R, while
190 allowing the data from OTL to be combined with other sources easily.

Here we show how we can obtain a map of the occurrences for a subset of the cat species that were included in the phylogeny retrieved from the Felidae study above (genus *Lynx*). We extract the species names from the phylogeny, and use them to ask for the records for these species found in GBIF (Figure 3). We include the code to reproduce this figure in Appendix.

195 As trait databases are becoming increasingly more common, and interfaces to the data they contains are being developed as R packages (e.g., the package `traits` Chamberlain et al. 2016), `rotl` provides a way to easily retrieve phylogenetic information for species trait data that are available.

In addition of an introduction on how to use the package, `rotl` includes two vignettes that demonstrate how to integrate a phylogeny and data associated with the taxa it represents.

200 Specifically, the “Data mashups” vignette provides an example of how to retrieve a phylogeny for species a researcher may have data for, and visualize both the phylogeny and data associated with the species at the tips. The other vignette titled “Meta-analysis” demonstrates how a complete comparative method analysis, including the gathering of data and a phylogeny, can be performed in a

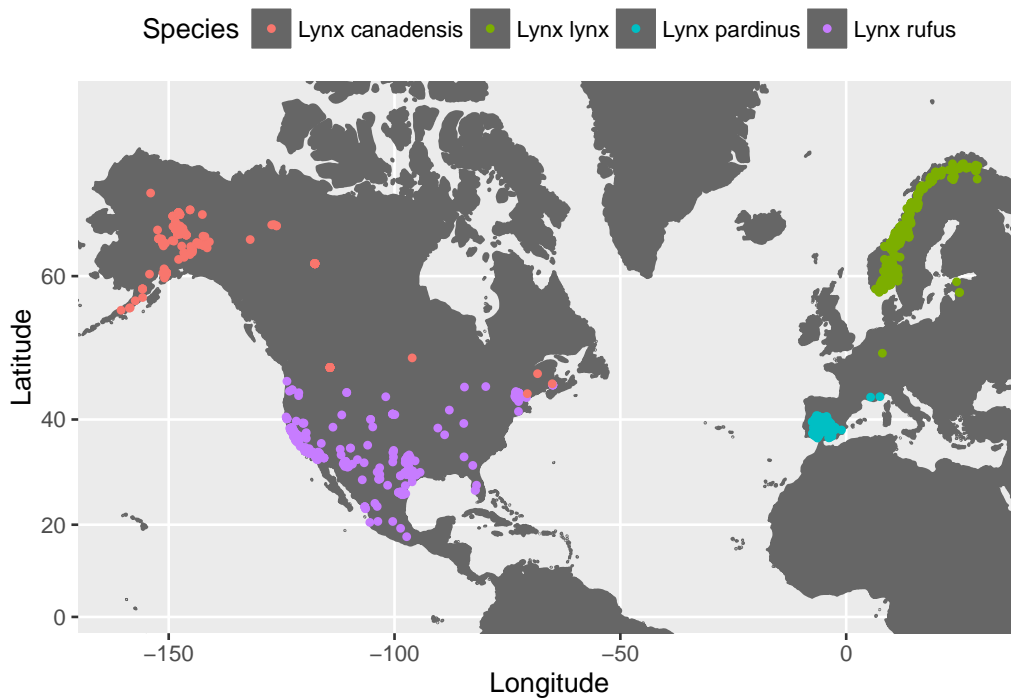


Figure 3: GBIF records for the species in *Lynx* included in the phylogeny associated with the study by Johnson et al. 2006.

single R session. We reproduce a published meta-analysis testing for differential investment in male
 205 and female offsprings among 51 species of birds. As new versions of the OTL API and rot1 are
 released, these vignettes will be kept up-to-date.

Concluding remarks

The recognition of the importance of phylogenies to account for the statistical
 non-independence of species in comparative methods, the recent development of methods to
 210 explore trait evolution or changes in diversification rates, and attempts to incorporate the
 evolutionary history of species forming ecological communities, have driven the need for accurate
 phylogenies. However, there is often a discrepancy between taxa targeted by studies wanting to use
 phylogenetic information, and taxa for which phylogenies are available. Typically, the latter result
 from focused studies of taxonomic groups, while the former encompass species found in a given

geographical location or ecosystem. We believe that by providing an easy-to-use interface to obtain phylogenies for an arbitrary set of taxa directly in R, `rotl` will be useful in a wide variety of contexts.

The accuracy and usefulness of the data provided by OTL relies on the community to make generated phylogenies (and their metadata) digitally available as tree files (i.e., Newick, NEXUS or NeXML). We strongly encourage researchers to submit their published phylogenies to OTL using the curator interface (<https://tree.opentreeoflife.org/curator>). By facilitating the discovery and re-use of published trees and of the synthetic Open Tree, we hope `rotl` will contribute to the wider adoption of best practices to make phylogenetic information available and re-usable.

Availability

`rotl` is free, open source, and released under a Simplified BSD license. Stable versions are available from the CRAN repository (<https://cran.r-project.org/package=rotl>), and development versions are available from GitHub (<https://github.com/ropensci/rotl>). This manuscript was built using version 0.5.0.901-`rotl`-ms.2 (<https://github.com/ropensci/rotl/tree/v0.5.0.901-rotl-ms.2>). The package is under active development, and authors welcome bug reports or feature requests via the GitHub repository. The source for this manuscript is available on GitHub (<https://github.com/fmichonneau/rotl-ms>).

Python (<https://github.com/OpenTreeOfLife/pyopentree>) and Ruby (<https://github.com/SpeciesFileGroup/bark>) libraries to interact with the OTL APIs are also available.

Acknowledgments

We would like to thank the organizers of the OpenTree of Life APIs hackathon that was held at the University of Michigan, Ann Arbor, 15-19 September, 2014, where the development of `rotl` was started. We would also like to thank Scott Chamberlain (rOpenSci) for providing a thorough

code review, Scott Chamberlain and Ross Mounce for commenting on the pre-print version of this
240 manuscript, Shinichi Nakagawa and Alistair Senior for their help in developing the package's
meta-analysis vignette, and Rich FitzJohn (Associate Editor) and an anonymous reviewer for
comments on the manuscript. DJW was supported by NIH Grant R01-GM101352. FM was
supported by iDigBio, and therefore this material is based upon work supported by the National
Science Foundation's Advancing Digitization of Biodiversity Collections Program (Cooperative
245 Agreement EF-1115210).

*

References

Boettiger, C., S. Chamberlain, E. Hart, and K. Ram. 2015a. Building Software, Building Community:
Lessons from the rOpenSci Project. *Journal of Open Research Software* 3.

250 Boettiger, C., S. Chamberlain, R. Vos, and H. Lapp. 2015b. RNeXML: a package for reading and
writing richly annotated phylogenetic, character, and trait data in R. *Methods in Ecology and
Evolution* Pages n/a–n/a.

Boettiger, C., D. T. Lang, and P. C. Wainwright. 2012. Rfishbase: Exploring, manipulating and
visualizing FishBase data from R. *Journal of Fish Biology* 81:2030–2039.

255 Boettiger, C. and D. Temple Lang. 2012. Treebase: an R package for discovery, access and
manipulation of online phylogenies. *Methods in Ecology and Evolution* 3:1060–1066.

Bolker, B., M. Butler, P. Cowan, D. de Vienne, D. Eddelbuettel, M. Holder, T. Jombart, S. Kembel,
F. Michonneau, D. Orme, B. O'Meara, E. Paradis, J. Regetz, and D. Zwickl. 2015. phylobase: Base
Package for Phylogenetic Structures and Comparative Data.

260 Chamberlain, S., Z. Foster, I. Bartomeus, D. LeBauer, and D. Harris. 2016. traits: Species Trait Data
from Around the Web.

Cranston, K., L. J. Harmon, M. A. O’Leary, and C. Lisle. 2014. Best practices for data sharing in phylogenetic research. *PLoS currents* 6:1–8.

Drew, B. T., R. Gazis, P. Cabezas, K. S. Swithers, J. Deng, R. Rodriguez, L. A. Katz, K. A. Crandall,
265 D. S. Hibbett, and D. E. Soltis. 2013. Lost Branches on the Tree of Life. *PLoS Biology* 11.

Hinchliff, C. E., S. A. Smith, J. F. Allman, J. G. Burleigh, R. Chaudhary, L. M. Coghill, K. A. Crandall,
J. Deng, B. T. Drew, R. Gazis, K. Gude, D. S. Hibbett, L. A. Katz, H. D. Laughinghouse, E. J.
McTavish, P. E. Midford, C. L. Owen, R. H. Ree, J. A. Rees, D. E. Soltis, T. Williams, and K. A.
Cranston. 2015. Synthesis of phylogeny and taxonomy into a comprehensive tree of life.
270 *Proceedings of the National Academy of Sciences* 112:12764–12769.

Johnson, W. E., E. Eizirik, J. Pecon-Slaterry, W. J. Murphy, A. Antunes, E. Teeling, and S. J. O’Brien.
2006. The Late Miocene Radiation of Modern Felidae: A Genetic Assessment. *Science* 311:73–77.

Kembel, S. W., P. D. Cowan, M. R. Helmus, W. K. Cornwell, H. Morlon, D. D. Ackerly, S. P.
Blomberg, and C. O. Webb. 2010. Picante: R tools for integrating phylogenies and ecology.
275 *Bioinformatics* 26:1463–1464.

Knuth, D. E. 1984. Literate programming. *The Computer Journal* (British Computer Society)
27:97–111.

Ksepka, D. T., J. F. Parham, J. F. Allman, M. J. Benton, M. T. Carrano, K. A. Cranston, P. C. J.
Donoghue, J. J. Head, E. J. Hermesen, R. B. Irmis, W. G. Joyce, M. Kohli, K. D. Lamm, D. Leeher,
280 J. L. Patané, P. D. Polly, M. J. Phillips, N. A. Smith, N. D. Smith, M. Van Tuinen, J. L. Ware, and
R. C. M. Warnock. 2015. The Fossil Calibration Database—A New Resource for Divergence
Dating. *Systematic Biology* 64:853–859.

Lewis, P. O. 2003. NCL: A C++ class library for interpreting data files in NEXUS format.
Bioinformatics 19:2330–2331.

- 285 Maddison, D. R., D. L. Swofford, and W. P. Maddison. 1997. NEXUS: an extensible file format for
systematic information. *Systematic biology* 46:590–621.
- Magee, A. F., M. R. May, and B. R. Moore. 2014. The dawn of open access to phylogenetic data.
- O'Meara, B. C. 2012. Evolutionary Inferences from Phylogenies: A Review of Methods. *Annual
Review of Ecology, Evolution, and Systematics* 43:267–285.
- 290 Paradis, E., J. Claude, and K. Strimmer. 2004. APE: Analyses of Phylogenetics and Evolution in R
language. *Bioinformatics* 20:289–290.
- Pearse, W. D., M. W. Cadotte, J. Cavender-Bares, A. R. Ives, C. M. Tucker, S. C. Walker, and M. R.
Helmus. 2015. Pez: Phylogenetics for the environmental sciences. *Bioinformatics* 31:2888–2890.
- Pearse, W. D. and A. Purvis. 2013. phyloGenerator: An automated phylogeny generation tool for
295 ecologists. *Methods in Ecology and Evolution* 4:692–698.
- Pennell, M. W., J. M. Eastman, G. J. Slater, J. W. Brown, J. C. Uyeda, R. G. FitzJohn, M. E. Alfaro,
and L. J. Harmon. 2014. Geiger V2.0: an Expanded Suite of Methods for Fitting
Macroevolutionary Models To Phylogenetic Trees. *Bioinformatics* 30:2216–2218.
- Pennell, M. W., R. G. FitzJohn, and W. K. Cornell. 2015. A simple approach for maximizing the
300 overlap of phylogenetic and comparative data. *Methods in Ecology and Evolution* .
- Pennell, M. W. and L. J. Harmon. 2013. An integrative view of phylogenetic comparative methods:
connections to population genetics, community ecology, and paleobiology. *Annals of the New
York Academy of Sciences* 1289:90–105.
- Revell, L. J. 2012. phytools: an R package for phylogenetic comparative biology (and other things).
305 *Methods in Ecology and Evolution* 3:217–223.
- Schliep, K. P. 2011. phangorn: Phylogenetic analysis in R. *Bioinformatics* 27:592–593.

Stoltzfus, A., B. O'Meara, J. Whitacre, R. Mounce, E. L. Gillespie, S. Kumar, D. F. Rosauer, and R. a. Vos. 2012. Sharing and re-use of phylogenetic trees (and associated data) to facilitate synthesis. BMC Research Notes 5:574.

310 Varela, S., J. González-Hernández, E. Casabella, and R. Barrientos. 2014. rAvis: An R-package for downloading information stored in proyecto AVIS, a citizen science bird project. PLoS ONE 9.

Varela, S., J. Gonzalez-Hernandez, L. F. Sgarbi, C. Marshall, M. D. Uhen, S. Peters, and M. McClennen. 2015. paleobioDB: An R package for downloading, visualizing and processing data from the Paleobiology Database. Ecography 38:419–425.

315 Vos, R. A., J. P. Balhoff, J. A. Caravas, M. T. Holder, H. Lapp, W. P. Maddison, P. E. Midford, A. Priyam, J. Sukumaran, X. Xia, and A. Stoltzfus. 2012. NeXML: Rich, extensible, and verifiable representation of comparative data and metadata. Systematic Biology 61:675–689.

Webb, C. O., D. D. Ackerly, and S. W. Kembel. 2008. Phylocom: Software for the analysis of phylogenetic community structure and trait evolution. Bioinformatics 24:2098–2100.

320 Webb, C. O., D. D. Ackerly, M. A. McPeck, and M. J. Donoghue. 2002. Phylogenies and Community Ecology. Annual Review of Ecology and Systematics 33:475–505.

Xie, Y. 2015. Dynamic Documents with R and knitr. 2nd ed. Chapman and Hall/CRC, Boca Raton, Florida.

Yu, G., D. Smith, H. Zhu, Y. Guan, and T. T.-Y. Lam. 2015. ggtree: an R package for visualization
325 and annotation of phylogenetic tree with different types of meta-data. Methods in Ecology and Evolution submitted.

Appendix

Source code used to generate Figure 3.

```
## load required packages. Versions used for the manuscript listed below.
library(rotl)      # v0.5.0.901
library(rgbif)     # v0.9.2
library(ggplot2)   # v2.1.0
library(maps)      # v3.1.0

## Import the felid tree using study and tree IDs discovered with
## studies_find_studies() in the manuscript
cat_tree <- get_study_tree(study_id = "pg_1981",
                          tree_id = "tree4052")

## Find the species of Lynx in the phylogeny
cat_species <- cat_tree$tip.label
lynx_species <- grep("^Lynx", cat_tree$tip.label, value = TRUE)

## Match the Lynx species to the GBIF identifiers
gbif_keys <- sapply(lynx_species,
                   function(x) name_backbone(name = x)$speciesKey,
                   USE.NAMES = FALSE)

## Search for the GBIF records for these species
lynx_loc <- occ_search(taxonKey = gbif_keys, limit = 500,
                      return = "data", fields = "minimal",
                      hasCoordinate = TRUE)

## Make a data frame of the results
lynx_loc <- do.call("rbind", lynx_loc)
names(lynx_loc)[1] <- "Species"

## Clean up the data with missing locality data
lynx_loc[["decimalLatitude"]] <- as.numeric(lynx_loc[["decimalLatitude"]])
lynx_loc[["decimalLongitude"]] <- as.numeric(lynx_loc[["decimalLongitude"]])
lynx_loc[lynx_loc[["decimalLatitude"]] == 0 &
         lynx_loc[["decimalLongitude"]] == 0,
         c("decimalLatitude", "decimalLongitude")] <- c(NA, NA)
lynx_loc <- lynx_loc[complete.cases(lynx_loc), ]

## Draw the map
world <- map_data("world")

ggplot(lynx_loc) +
  annotation_map(world, fill="gray40", color="gray40") +
  geom_point(aes(y = decimalLatitude, x = decimalLongitude, color = Species),
            size = 1) +
  coord_map(projection = "mercator", orientation = c(90, 0, 0)) +
  xlab("Longitude") + ylab("Latitude") +
  theme(legend.position="top", legend.key = element_rect(fill = "gray40")) +
  ylim(c(0,72))
```