Journal Title
XX(X):1–9
©The Author(s) 0000
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Phylotastic: improving access to tree-of-life knowledge with flexible, on-the-fly delivery of trees

Van D. Nguyen[1*] , Thanh H. Nguyen[1†] , Abu Saleh Md. Tayeen[1‡] , H. Dail Laughinghouse IV[2,3§] , Luna L. Sánchez-Reyes[4¶] , Jodie Wiggins[4‖] , Enrico Pontelli[1] , Dmitry Mozzherin[5], Brian O'Meara[4††] and Arlin Stoltzfus[6,2‡‡ §§]**

[1]*Department of Computer Science, New Mexico State University, Box 30001, MSC CS, Las Cruces, 88003, NM, USA*
[2]*Institute for Bioscience and Biotechnology Research, 9600 Gudelsky Drive, Rockville, 20850, MD, USA*
[3]*Ft. Lauderdale Research and Education Center, University of Florida/IFAS, 3205 College Avenue, Davie, 33314, FL, USA*
[4]*Department of Ecology and Evolutionary Biology, University of Tennessee, 569 Dabney Hall, Knoxville, 37996, TN, USA*
[5]*Illinois Natural History Survey, Species File Group, University of Illinois, 1816 South Oak St., Champaign, 61820, IL, USA*
[6]*Office of Data and Informatics, NIST, 100 Bureau Drive, Gaithersburg, 20899, MD, USA*

Abstract

(1) A comprehensive phylogeny of species, i.e., a tree of life, has potential uses in a variety of contexts in research and education. This potential is limited because accessing the tree of life requires special knowledge, complex software, or long periods of training.

(2) To mitigate these barriers, the Phylotastic project provides access to expert phylogenetic knowledge through web-servicess, with the aim to make it as easy to get a

phylogeny of species as it is to get online driving directions. In prior work, we presented a design for an open system to validate and manage species names, find phylogeny resources, extract subtrees matching a user's species list, scale them to time, and integrate other information and resources (e.g., images) from online sources.

(3) Here we report the implementation of a robust, publicly accessible system for on-the-fly delivery of phylogenetic knowledge, developed with user feedback on what types of functionality are considered useful by researchers and educators. The implementation currently consists of a web portal to execute a general workflow to obtain species phylogenies (scaled by geologic time and decorated with thumbnail images); more than 30 underlying web services accessible via a common registry; and code toolkits in R and Python so that others can create applications that leverage these services. These resources cover most of the use-cases identified in our analysis of user needs.

(4) The Phylotastic system, accessible via http://www.phylotastic.org, provides a unique resource to access the current state of phylogenetic knowledge, useful for a variety of cases in which a tree extracted quickly from online resources (as distinct from a tree custom-made from character data) is sufficient, as it is for many casual uses of trees identified here.

Introduction

A phylogeny broadly covering the diversity of known species would provide "a comparative and predictive framework for all fundamental and applied biology" [**?**]. Furthermore, a tree of life is not only a tool for researchers: like a detailed map of world geography, or a periodic table of the chemical elements, a tree of life is a fundamental guide to the living world, something to be consulted by policy-makers, taught by educators at all levels, and explored by curious members of the public. Therefore, it is important for current knowledge of the tree of life to be discoverable and accessible.

Yet, phylogenetics is traditionally a technical discipline with inaccessible products. Phylogenies are difficult to generate from raw data following best practices, which are constantly updated to reflect new methodologies and expanding sources of data. Phylogeny experts typically do not employ practices that make their phylogenies discoverable, accessible, and re-useable [**? ? ?**].

This situation is beginning to change, and the potential has emerged recently to make expert phylogenetic knowledge accessible for a broad array of uses. First, a supertree constructed from available taxonomies and published phylogenies is periodically updated by the OpenTree project [**? ?**]. The current "synthetic tree" has over 2 million species and is constructed from 987 source trees. Second, the practical value of this tree of life is greatly enhanced by the availability of systematic data from other resources, including images and basic taxon information (size, habitat, etc) from the Encyclopedia of Life (EOL) [**?**], taxonomic name mappings [**? ?**], and occurrence records from GBIF (Global Biodiversity Information Facility) [**?**] or iNaturalist. Third, there has been a common movement among all of the above (and other) resource-providers to support programmatic access to content via web services. A web service provides access to data or operations over the world wide web, using standard machine-to-machine communication protocols. The simplest web-service queries can be typed manually into a web browser, but web services are designed to be interoperable, so that they can be invoked automatically by other software, and chained together to create complex workflows to the advantage of users needs.

The combination of these developments makes possible the kind of "Phylotastic" system envisioned previously [**?**]. The Phylotastic design concept leverages an open system of web services to support various workflows to discover, modify, and add value to phylogenetic trees. The aim is to make it as easy to get a tree online as it is to get driving directions.

Here we describe the first full implementation of this design. In order to identify the most common scenarios for getting a quick tree (use-cases), possibly combined with other data, we consulted a diverse set of potential users, emphasizing non-expert users (thus complementing the analysis in [**?**]). Based on the list of use-cases that resulted from the consultation, we designed and implemented more than 30 web services, a web-service registry, libraries in R and Python, and a web portal. The Phylotastic web portal is an interactive web application that illustrates the capabilities of the system, supporting various workflows for obtaining trees beginning with a list of taxa, which may be assembled in different ways, e.g., scraping scientific names from electronic documents or web sites. The resulting trees, which have thumbnail images and links to catalog entries, may be saved as images or data for independent processing. The underlying web services that provide this functionality also may be accessed using custom software, with the aid of the R and Python libraries, that allow generating workflows for extended use-cases.

Description

Analysis and design

The concept for a Phylotastic system [**?**] grew from the research interests of scientists with expertise in phylogenetics and informatics. To ensure broader value to the community, we developed a prototype Phylotastic web portal and used it to obtain feedback (via correspondence as well as in-person interviews) from a broader range of potential users, including researchers and educators at multiple levels who were not experts in informatics or phylogenetics. Based on this information, we prioritized the development of tools and workflows to support the following use-cases.

1. **Generate a tree from a specified list of taxa.** Provide a tree for a user-supplied list of species or higher taxa.

2. **Generate a tree of N species sampled from a named taxon.** Given a named taxon and a number N, provide a tree with N species chosen in some way, e.g., at random, by popularity, or by maximal diversity (taxonomic or phylogenetic).

3. **Generate a phylo-guide from an electronic resource.** Create a tree with images and links to species information from a resource that includes taxonomic names, such

as a scientific paper, a web site about ants, or a document listing the species found in a park or zoo.

4. Contextualize phylogenetic relationships. Place a given list of species in a larger tree that shows phylogenetic relationships in a broader context. Or, given a small set of taxa, generate a tree using representative species that illustrate the relationship, possibly including species from other (unspecified) taxa for context.

5. Integrate data or metadata with phylogeny. Given the set of species implicated by any method described above, return a tree and an associated data table integrating information or resources of interest, including images, links to information (e.g., EOL or wikipedia), or data on features such as toxicity, pathogenicity, availability of fossil data, medicinal value, conservation status, size, biogeography, or habitat.

Currently, the system we implemented covers each use case partially or wholly (see Examples and Discussion). Some types of data requested by potential users (e.g., medicinal value, pathogenicity), and some types of operations, are not yet implemented in available algorithms (e.g., choosing a set of species based on both popularity and diversity).

Implementations

The types of operations identified previously for the implementation of a Phylotastic system [**?**] include (1) taxonomic name resolution, i.e., rectifying possible misspellings in scientific names by matching them to authoritative taxonomic data bases; (2) tree retrieval, i.e., finding available trees with coverage of user-identified taxa and extracting subtrees; (3) tree scaling, i.e., assigning branch lengths to subtrees; (4) tree comparison, i.e., comparing subtrees; (5) taxon information and images, i.e., getting and adding data or metadata from species or higher taxa in the subtree; (6) rendering a tree graphically. The expanded set of use cases identified above implicates a slightly larger set of operations: (7) scraping names, i.e., extracting taxonomic names embedded in text and media; (8) taxon sampling, i.e., choosing species from a taxonomic group by some criteria; (9) converting common names to scientific names;and (10) list management, i.e., creating, reading, publishing, updating, and removing a list of names associated with a managed user account.

Web services: The above operations were translated to actual tools (services) that can be accessed and manipulated by users, and that are implemented and executed via web services written in Python. In general, Phylotastic web services are designed to operate synchronously. This means that workflows are carried out in real time. One exception is the set of services to manage persistent lists, so that a list created by a user in a session may be accessed in a later session, or by a different user, enhancing the potential for reusability.

Currently we have made available more than 30 web services described in Table 1. Some of these services are thin wrappers around external services, while others were developed for this project. Documentation for phylotastic web services can be found at the portal and the source code is available at GitHub project.

Code toolkits: We developed R [?] and python packages to allow users to access the Phylotastic system with their own software and computers, using functions and methods written in the native language. Both toolkits provide access to nearly all of the categories of services described in Table 1. The rphylotastic package wraps Phylotastic web services using the R packages jsonlite [?] and httr [?], designed for working with URLs and HTTP calls, with roxygen2 [?] for documentation and testthat [?] for automated tests.

The phylotastic_py package has a main module, *phylotastic_services* composed of sub-modules implementing the different Phylotastic services. The package documentation was generated with Sphinx [?]. To test the functional correctness of sub-modules, a set of unit tests were implemented using Python Unit Testing Framework and deployed in Travis-CI (https://travis-ci.org/).

Web portal: The Phylotastic portal (http://portal.phylotastic.org) provides a graphical interface to create or select a list of taxa, manage the list, and retrieve a tree, which is displayed using an embedded viewer. A list of taxa may be designated by

- choosing a public list, e.g., "Finding Nemo" list available directly on the web portal.
- uploading a list, as a text file with one name per line, or as a Darwin Core Archive (DwC-A) file
- scraping names from an electronic resource (PDF, doc, txt, xls), including image files (processed via optical character recognition), either uploaded or identified via URL
- from a named taxon, choosing:
  - all or a random sample of N species
  - species with known genomes (via NCBI services)
  - species with occurrence records in a given location (via iNaturalist)

The portal manages sessions and accounts, so that users can maintain persistent lists and trees. Lists can be downloaded in a simple format with one name per line. Trees can be downloaded as Newick or as a graphical rendering in png or SVG format. These features allow the portal to support a wide range of use-cases, as explained below (Examples).

The web portal is written in Ruby using Rails, a model-view-controller architecture for rapid design and development of robust web applications. The portal takes advantage of PostgreSQL for database management; Paperclip for managing file attachments; TwitterBootstrap, JQuery, and FontAwesome for front-end development; Devise for authentication management; Wicked PDF for PDF generation; Capybara and Minitest for automated testing; Docker and Kubernetes for containerization and deployment. The test suite covers model tests, controller tests, and interactive tests (simulated in Poltergeist, which mimics user interactions).

Web services registry: The portal and library software described above depend on concrete workflows instantiated by specific known services. However, if web services are described abstractly (e.g., in terms of input types, options, and output types) in a web-services registry using a machine-readable language, then it is possible for an intelligent system to query the registry, discover useful services, and construct a workflow on the fly. Indeed, the Phylotastic project was designed with the aim of supporting this kind of automatic workflow construction, and all the web services above are described abstractly in the registry. The use and importance of the Phylotastic web services registry in automated workflow composition, and in fault-tolerant execution (e.g., re-computing a workflow to avoid an unresponsive service), has been thoroughly described elsewhere [**?** **?**].

Examples

Each of the use-cases described above is supported, partly or fully, by the Phylotastic portal. The first 3 use-cases (user-supplied list, sample from taxon, generate phylo-guide), are straightforward applications of the basic Portal workflow of designating a list, managing the list, and retrieving a tree for visualization. Because lists can be downloaded in a simple format with one name per line, it is straightforward to combine or edit lists manually. This makes it possible to embed a list of focal species in a sample from a higher taxon (use-case 4), or to construct the example used in Fig 2 (aquatic mammals), explained in more detail below.

The portal integrates a limited set of useful data and metadata (use-case 5), specifically EOL links and thumbnail images. One way to integrate other data by combining the portal and external graphical tools is to (1) use the portal to scrape names from a data table (in text or Excel format), then obtain and download (as a Newick file) a phylogeny for the implicated taxa, then (2) combine the original data table and the downloaded tree using a web tool designed for this purpose, such as EvolView [**?**] or IToL [**?**].

For users who are able to write code in Python or R, much more flexibility is possible using the toolkits described above. Each of the use-cases above is supported, at least partly, by the toolkits. An example of contextualizing a list of species within a larger taxon (use-case 4) is given in detail below, along with an example of integrating external data with a phylogeny (use-case 5).

Birds from Yellowstone

In this use case, a visitor to Yellowstone National Park composes a list of birds seen in the park, then uses Phylotastic tools to: (1) translate the common names to scientific names; (2) get all scientific names of Yellowstone birds from the U.S. National Park Service website (https://www.nps.gov/), (3) get a dated phylogeny of all species and plot them with observed species highlighted, as shown in Fig. 1. The scaled tree is constructed from curated trees via DateLife, without any need for the user to conduct phylogenetic inference or calibration. The format of the Park Service list does not matter, because names are scraped using intelligent algorithms. This example can be conducted as follows using the DateLife and R-Phylotastic packages:

```
library("rphylotastic", "datelife")
birds_I_saw <- taxa_common_to_scientific(c("Osprey",
  "House sparrow", "Mallard duck", "American Robin",
  "Song Sparrow", "Mourning Dove", "House Wren"))
yellowstone_birds <- url_get_scientific_names(
    URL="https://www.nps.gov/yell/learn/nature/upload/
    BirdChecklist2014.pdf")
yellowstone_bird_tree <- datelife::datelife_search(
    taxa_get_otol_tree(yellowstone_birds), summary_format
    = "phylo_median")
```

Aquatic mammals

The portal screencast (www.youtube.com/watch?v=8Q5m0ldaGIg) illustrates the ability to upload a PDF, extract names, and generate a tree with images and links. In this case, the source is a scientific publication about the origin of Cetacea (whales and dolphins) by Tsagkogeorga, et al. [**?**]. The portal extracts 39 taxon names from the PDF, including 26 species names. Retrieving a tree yields a tree with 26 tips including whales, dolphins and various mammalian outgroups. The screencast explains how to expand this into a lesson about repeated evolution, by including representatives of two other aquatic groups, with relatives of each. The list of species scraped from the PDF is downloaded, some names are removed to simplify the presentation, then a small list of names is added for the Pinnipedia (the seals and sea lions) with their carnivore relatives, and the Sirenia (the dugongs and manatees) with the elephant as a close relative. To obtain Fig. 2, a time-scaled tree from this list is decorated with thumbnail images, and the 3 groups are highlighted using the clade-highlighting feature of the portal's tree-viewer.

Use case 5, integrate other data than images

Discussion

Comparison with other resources

Phylotastic resources can be considered broadly as a way of making tree-of-life knowledge accessible conveniently to non-specialists, and as a way of supporting automated workflows to get phylogenetic knowledge and mash it up with other information. The system is designed for automation, integration, and convenience. As such, it is unique. Ordinary approaches of phylogenetic inference are out of reach for most users, even online systems that, from the perspective of experts, provide convenient interfaces (e.g., CIPRES, Phylogeny.fr). ToLWeb [**?**], OneZoom [**?**], IToL [**?**] and the OpenTree web portal all allow interactive browsing of a tree of life, and OpenTree and TREEBase both provide some web services. However, the Phylotastic project, via either the portal, or the suite of web services (accessed directly or via toolboxes), provides a unique functionality. For instance, combining automated tree retrieval with a diverse set of powerful methods to generate lists of interest to users is a combination that makes the Phylotastic project unique, and this combination only partly captures the capacity of the portal or the suite of web services.

The phylotastic system is unique in that it encompasses all functionality from these alternative resources in a dynamic, and responsive fault-tolerant way, through both a set of tools to support the development of software and a multi-purpose interactive tool, the Phylotastic web portal.

Development priorities

Some of the features repeatedly requested by users are (1) supporting the use of common names, (2) integrating character data, (3) integrating species data that are of broad interest (e.g., medicinal value, pathogenicity, endangered status), and (4) sampling species from a taxon by popularity. The portal currently provides initial (limited) support for common names and for sampling by popularity (based on functionality provided by OneZoom).

Availability

Code, applications and services may be discovered and accessed via the project's web home at https://www.phylotastic.org. The web portal is accessible via https://portal.phylotastic.org, and the registry is at https://registry.phylotastic.org. Source code is available under an open-source license from the GitHub Phylotastic organization (https://www.github.com/phylotastic). At present, all resources are accessible without restriction; usage restrictions on web services may be imposed in the future if necessary to ensure that the services are broadly usable. The stable version of rphylotastic can be installed from R directly from the CRAN repository (https://cran.r-project.org/package=rphylotastic) using *install.packages(pkgs = "rphylotastic")*. Development versions are available from GitHub repository (https://github.com/phylotastic/rphylotastic) and can be installed using *devtools::install_github("phylotastic/rphylotastic")*. The Python library can be installed following the instructions at https://github.com/phylotastic/phylotastic_py.

Author contributions

VDN and ASMD implemented the portal, and contributed to design and testing, along with AS and HDL; AS and HDL analyzed requirements for the portal and web services; LLSR and BO designed, implemented and tested the DateLife web portal, DateLife web services, and the rphylotastic package; ASMD and THN implemented web services, and contributed to design and testing along with AS; ASMD designed and implemented the phylotastic_py package; DM designed and implemented improvements to taxonomic name resolution services; AS, BO and EP conceived of the project and oversaw all aspects of design and testing; AS drafted the manuscript, which was completed with the help of the other authors.

Acknowledgements

This work was supported by funding from the US National Science Foundation (award 1458572, "Collaborative Research: ABI Development: An open infrastructure to disseminate phylogenetic knowledge"). The identification of any specific commercial products is for the purpose of specifying a protocol, and does not imply a recommendation or endorsement by the National Institute of Standards and Technology. We thank Yan Wong, James Rosindell, Karen Cranston, Jonathan Rees, Jaime Huerta-Cepas, Brian Sidlauskas, Jim Allman, Ramona Walls, Mark Holder, Daisie Huang, Cyndy Parr, Katja Schulz, and Son Tran, for useful comments, discussions, and technical advice. We thank many others who contributed indirectly to this project through their participation in two NESCent hackathons with a Phylotastic theme.

TABLE 1
Main Phylotastic services description

Web Service	Description
Common Names to Scientific Names	Get the scientific name of a species from its common name
`NCBI_common_name`	following the NCBI database
`EBI_common_name`	following EBI services
`ITIS_common_name`	following ITIS services
`TROPICOS_common_name`	following TROPICOS services
`EOL_common_name`	following EOL services
Scientific Name Extraction	Scrape scientific names from a URL, text or any type of file
`GNRD_wrapper_URL;`	using Global Names Recognition and Discovery (GNRD) services
`GNRD_wrapper_text;`	
`GNRD_wrapper_file`	
`TaxonFinder_wrapper_URL;`	using Taxon Finder
`TaxonFinder_wrapper_text`	
Taxonomic Name Resolution	Match scientific names to authoritative taxonomies and resolve mismatches
`OToL_TNRS_wrapper`	using the Open Tree of Life taxonomy
`GNR_TNRS_wrapper`	using the Global Names Resolver tool (several taxonomies)
`iPlant_TNRS_wrapper`	using iPlant collaborative services
Taxon Sampling	Get all scientific names of species within a given higher-taxon name
`Taxon_all_species`	
`Taxon_country_species`	and are found in a given country (using iNaturalist database),
`Taxon_genome_species`	or have a genome sequence (deposited in NCBI),
`Taxon_popular_species`	or match the most popular species within the taxon using OneZoom tool
Taxon Information and Images	Get various information of a species such as
`Image_url_species`	image urls and corresponding license information using EOL
`Info_url_species`	information urls from EOL
`ECOS_Conservation`	conservation status from ECO services
Tree Retrieval	Get phylogenetic trees from a list of taxa
`OToL_wrapper_Tree;`	from Open Tree of Life synthetic tree
`OToL_supported_studies`	and all supporting studies
`Phylomatic_wrapper_Tree`	from Phylomatic
`Treebase_Tree`	from TreeBase
`Supersmart_wrapper_Tree`	using supersmart
Tree Scaling	Scale branch lengths of a tree to geologic time
`Datelife_scale_tree`	using the DateLife service
`OToL_scale_tree`	using OToLs unofficial scaling service
Tree Comparison	Compare two phylogenetic trees symmetrically
`Compare_trees`	
List Management	Save, publish, access, remove or update lists of names.
`Add_new_list; Get_list;`	
`Replace_species_list;`	
`Update_metadata_list;`	
`Remove_list;`	

yellowstone_bird_tree_plot2_grayscale-A.pdf

FIGURE 1. Dated phylogenetic tree of birds from Yellowstone National Park obtained as described in the text. Species from the user's observation list are shown in red. Bird families are delimited by gray arcs. This figure was generated with functions from rphylotastic, datelife, and ape [?] R packages. Code has been made fully reproducible by implementing a plan with the R package drake [?].

aquatic_mammals.jpg

FIGURE 2. Three separate groups of aquatic mammals (Cetacea, Pinnipedia, and Sirenia). This figure was generated using the Phylotastic portal, as described in the text.