The Journal of Open Source Software

Physcraper: A Python package for continually updated gene trees

Luna L. Sanchez Reyes[1], **Martha Kandziora**[1], **and Emily Jane McTavish**[1]

1 University of California, Merced

DOI:

Software

- Review
- Repository
- Archive

Submitted:
Published:

Abstract

1. Phylogenies are a key part of research in all areas of biology. Tools that automatize some parts of the process of phylogenetic reconstruction (mainly character matrix construction) have been developed for the advantage of both specialists in the field of phylogenetics and nonspecialists. However, interpretation of results, comparison with previously available phylogenetic hypotheses, and choosing of one phylogeny for downstream analyses and discussion still impose difficulties to one that is not a specialist either on phylogenetic methods or on a particular group of study.

2. Physcraper is an open-source, command-line Python program that automatizes the update of published phylogenies by making use of public DNA sequence data and taxonomic information, providing a framework for comparison of phylogenies.

3. Physcraper can be used by the nonspecialist, as a tool to generate phylogenetic hypothesis based on already available expert phylogenetic knowledge. Phylogeneticists and group specialists will find it useful as a tool to facilitate comparison of alternative phylogenetic hypotheses (topologies). ***Is physcraper intended for the nonspecialist?? We have two types of nonspecialists: the ones that do not know about phylogenetic methods and the ones that might know about phylogenetic methods but do not know much about a certain biological group.***

4. Physcraper implements node by node comparison of the the original and the updated trees using the conflict API of OToL.

5. We hope the physcraper workflow demonstrates the benefits of opening results in phylogenetics and encourages researchers to strive for better data sharing practices.

6. Physcraper can be used with any OS. Detailed instructions for installation and use are available at https://github.com/McTavishLab/physcraper.

Introduction

Phylogenies are important. Generating phylogenies is not easy. The process of phylogenetic reconstruction implies many steps that can be generalized to the following: 1. Obtention of molecular or morphological character data – get DNA from some organisms and sequence it, or get it from an online repository, such as GenBank. 1. Assemble a hypothesis of homology – Create a matrix of your character data, by aligning the sequences, in the case of molecular data. 1. Analyse this hypothesis of homology to infer phylogenetic relationships among the organisms you are studying – Use different available programs to infer molecular evolution, trees and times of divergence. 1. Discuss the inferred relationships in the context of previous hypothesis, the biology and biogeography of the organisms, etc. – Answer the question, *is this phylogenetic solution fair/reasonable?*

Each of these steps require different types of specialized training: in the field, in the lab, in front of a computer, discussions with experts in the methods, and/or in the biological group of study. All of these steps also require considerable amounts of time for training and implementation.

In the past decade, various studies have developed solutions to automatize the first and second steps, by creating pipelines that mine already available molecular data from the GenBank repository, to obtain homologous characters that can be used for phylogenetic reconstruction. These tools have been presented as aid for the nonspecialist to decrease some of the difficulties in the generation of phylogenetic knowledge. However, they are not that often used as so, suggesting that there are still difficulties for the nonspecialist. The phylogenetic community has some reserves towards these tools, too. Mainly because they sometimes act as a black box. However, automatizing the assembly of the character data set is a crucial step towards reproducibility for a task that was otherwise primarily artisanal and hence largely non-reproducible.

Even if it is hard to obtain phylogenies, we invest copious amounts of time and energy in generating them. They are crucial to solve problems such as food security, global warming, global health. There is a lot of phylogenetic knowledge already available in published peer-reviewed studies. In this sense, the non-specialists (and also the specialist) face a new problem: how do I choose the best phylogeny.

Public phylogenies can be updated with the ever increasing amount of genetic data that is available on GenBank.

A way to automatize the comparison of phylogenetic hypotheses and to allow reproducibility of the last step of the process.

A key aspect of the standard phylogenetic workflow is comparison with already existing phylogenetic hypotheses and with phylogenies that are considered "best" by experts not only in phylogenetics, but also experts on the focal group of study.

It is well known that GenBank holds enormous amounts of genetic data, and it continues to grow. A lot of this genetic data has the potential to be used to reconstruct the phylogenetic history of various organisms (Sanderson, Boss, Chen, Cranston, & Wehe, 2008). Pipelines that harness this potential have been available for over a decade now, such as the Phylota browser, and PHLAWD. New ones keep on being developed, such as SUPERSMART and the upgraded version of PHLAWD, PyPHLAWD. Notably large phylogenies have been constructed using some of these tools, Some other have not been used that much. So, how well accepted is this approach in the community?

Concerns with these tools: Errors in identification of sequences Little control along the process Too much of a black box?

Most of these phylogenies are being constructed by people learning about the methods, so they want to know what is going on.

The pipelines are so powerful and they will give you an answer, but there is no way to assess if it is better than previous answers, it just assumes it is better because it used more data.

All these pipelines start tree construction from zero?

The goal of Physcraper is to build upon previous phylogenetic knowledge, allowing a direct comparison of existing phylogenies to phylogenies that are constructed using new genetic data from GenBank

To achieve this, Physcraper uses the Open Tree of Life phylesystem and connects it to the TreeBase database, to (1) get the original DNA data set matrices (alignments) that produced a phylogeny that was published and then made available in the OToL database, (2) use this DNA alignments as a starting point to get new genetic data belonging to the focal group of study, to (3) finally update the phylogenetic relationships in the group.

A less automated workflow is one in which the alignments that generated the published phylogeny are stored in other public database (such as DRYAD) or elsewhere (the users computer), and are provided by the users.

The original tree is by default used as starting tree for the phylogenetic searches, but it can also be set as a full topological constraint or not used at all, depending on the goals of the user.

Physcraper implements node by node comparison of the the original and the updated trees, using the conflict API of OToL.

How does Physcraper work?

The input: a study tree and an alignment

- The phylogenetic tree has to be in the Open Tree of Life store (McTavish et al., 2015). You can choose from a variety of published trees supporting any node of the Tree of Life. If the tree you are interested in is not in Open Tree of Life, you can easily upload it via the curator tool.
- The alignment should be a gene alignment that was used to generate the tree. The alignments are usually stored in a public repository such as TreeBase (Piel et al., 2009; Vos et al., 2012) or DRYAD (http://datadryad.org/). If the alignment is stored in TreeBase, `physcraper` can download it directly either from the TreeBASE website (https://treebase.org/) or through its GitHub repositiry, SuperTreeBASE (https://github.com/TreeBASE/supertreebase). If the alignment is on another repository, a copy of it has to be downloaded by the user, and it's location has to be provided as an argument.
- A taxon name matching step is performed to verify that all taxon names on the tips of the tree are in the character matrix and vice versa.
- A ".csv" file with the summary of taxon name matching is produced for the user.
- Unmatched taxon names are dropped from the tree and alignment. ***What is the minimum amount of tips necessary for a physcraper run??*** Just one!
- A ".tre" and ".aln" files are generated and saved for a `physcraper` run.

DNA sequence search and cleaning

- The first step is to identify the MRCA of all taxa in the tree that is also a named clade in the NCBI taxonomy. This "taxonomic MRCA" can be different from the phylogenetic MRCA when the latter is an unnamed clade. ***Does physcraper take into account the focal clade from Open Tree in any way???*** Not really, it is a search MRCA more than a taxonomic, the default is to get the ingroup from Otol, which is not the focal group.
- The taxonomic MRCA is used to constrain the DNA sequence search within that taxonomic group on the GenBank database.
- The DNA sequence search can be done on a local database that is easily setup by the user, or remotely.
- The BLAST algorithm is used to identify all DNA sequences in the database within the taxonomic MRCA that are similar to each sequence in the alignment ***How does the blast is setup? Is it an all-blast-all??*** The rmote BLAST uses bioPython and the local BLAST uses BLASTn
- Reverse complement sequences are identified and translated.
- Sequences found for each

- A fasta file containing all sequences resulting from the search is generated for the user.
- ***Add cycles of blast searches with new sequences***
- How many sequences per species? Was hardcoded to 5, but it will be an argument now, default to 5. ***How does pyphlawd, or phylota does it?***

DNA sequence alignment

- The software MUSCLE (Edgar, 2004) is implemented for profile alignment, in which the original alignment is used as a template to align all new sequences.

Tree reconstruction

- A gene tree is reconstructed for each alignment provided, using RAxML with bootstrap replicates.
- The final result is a gene tree couples to the conlict info.

Tree comparison

- Conflict information can only be generated in the context of the whole Open Tree of Life. Otherwise, it is not really possible to get conflict data. ***- One way to compare two independent phylogenetic trees is to compare them both to the synthetic OToL and then measure how well they do against each other***

Use case/ example

Imagine you are starting to work on a new biological group X. You have not much of an idea about its phylogenetic relationships, you are a newly established researcher, and the group is not anything any of your collaborators have worked on before. A good idea is to start an intensive literature review on the phylogenetics of the group. Rapidly, you find out there are 5 different phylogenies, that used different markers, and that the papers, published at different times, do not discuss which phylogeny is the one accepted by the expert community on X. You might need to go to the annual conference of X, and even then, you might only find different and contrasting opinions. Somewhere along these months or even years doing this task, you looked into the the OToL database. You found in there some or all the published trees of X, along with a tree that has been deemed the best tree by curators and ideally experts on X?

Ascomycota Example

Let's be more specific now about our X group and say it is the Ascomycota. The best tree currently available in OToL was published by Schoch et al. (2009). The first step, is to get the Open Tree of Life study id. There are some options to do this: - You can go to the Open Tree of Life website and browse until you find it, or - you can get the study id using R tools: - By using the TreeBase ID of the study (which is not fully exposed on the TreeBase website home page of the study, so you have to really look it up manually):

```
rotl::studies_find_studies(property = "treebaseId", value = "S2137")
##   study_ids n_trees         tree_ids candidate study_year title
## 1    pg_238       2 tree109, tree110                 2009
##                                    study_doi
## 1 http://dx.doi.org/10.1093/sysbio/syp020
```

- By using the name of the focal clade of study (but this behaved very differently):

```
rotl::studies_find_studies(property="ot:focalCladeOTTTaxonName", value="Ascomycota
```

Once we have the study id, we can gather the trees published on that study:

```
rotl::get_tree_ids(rotl::get_study_meta("pg_238"))
## [1] "tree109" "tree110"
rotl::candidate_for_synth(rotl::get_study_meta("pg_238"))
## NULL
my_trees <- rotl::get_study("pg_238")
```

Both trees from this study have 434 tips.

Let's check what one of the trees looks like:

1. Download the alignment from TreeBase If you are on the TreeBase home page of the study, you can navigate to the matrix tab, and manually download the alignments that were used to reconstruct the trees reported on the study that were also uploaded to TreeBase and to the Open Tree of Life repository. To make this task easier, you can use a command to download everything into your working folder:

```
physcraper_run.py -s pg_238 -t tree109 -o ../physcraper_example/pg_238
```

In this example, all alignments posted on TreeBase were used to reconstruct both trees.

1. With the study id and the alignment files saved locally, we can do a physcraper run with the command:

```
physcraper_run.py -s pg_238 -t tree109 -a treebase_alns/pg_238tree109.aln -as "nex
```

Testudines example

Phylogeny of the Testudines 6 tips from Crawford et al. (2012) There is just one tree in OToL. There is just one alignment on treebase with all the 1 145 loci.

```
physcraper_run.py -s pg_2573 -t tree5959 -tb -db ~/branchinecta/local_blast_db/ -o
```

Tools on a similar track:

Tools that do similar things: pyPhlawd (Smith & Walker, 2019) SUPERSMART (Antonelli et al., 2017)

Acknowledgements

We acknowledge contributions from

References

Antonelli, A., Hettling, H., Condamine, F. L., Vos, K., Nilsson, R. H., Sanderson, M. J., Sauquet, H., et al. (2017). Toward a self-updating platform for estimating rates of speciation and migration, ages, and relationships of taxa. *Systematic Biology*, *66*(2), 152–166.

Crawford, N. G., Faircloth, B. C., McCormack, J. E., Brumfield, R. T., Winker, K., & Glenn, T. C. (2012). More than 1000 ultraconserved elements provide evidence that turtles are the sister group of archosaurs. *Biology letters*, *8*(5), 783–786. doi:10.1098/rsbl.2012.0331

Edgar, R. C. (2004). MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, *32*(5), 1792–1797. doi:10.1093/nar/gkh340

McTavish, E. J., Hinchliff, C. E., Allman, J. F., Brown, J. W., Cranston, K. A., Holder, M. T., Rees, J. A., et al. (2015). Phylesystem: A git-based data store for community-curated phylogenetic estimates. *Bioinformatics*, *31*(17), 2794–2800. doi:10.1093/bioinformatics/btv276

Piel, W., Chan, L., Dominus, M., Ruan, J., Vos, R., & Tannen, V. (2009). Treebase v. 2: A database of phylogenetic knowledge. E-biosphere. London.

Sanderson, M. J., Boss, D., Chen, D., Cranston, K. A., & Wehe, A. (2008). The PhyLoTA Browser: Processing GenBank for Molecular Phylogenetics Research. *Systematic Biology*, *57*(3), 335–346. doi:10.1080/10635150802158688

Schoch, C. L., Sung, G.-H., López-Giráldez, F., Townsend, J. P., Miadlikowska, J., Hofstetter, V., Robbertse, B., et al. (2009). The ascomycota tree of life: A phylum-wide phylogeny clarifies the origin and evolution of fundamental reproductive and ecological traits. *Systematic biology*, *58*(2), 224–239.

Smith, S. A., & Walker, J. F. (2019). PyPHLAWD: A python tool for phylogenetic dataset construction. *Methods in Ecology and Evolution*, *10*(1), 104–108.

Vos, R. A., Balhoff, J. P., Caravas, J. A., Holder, M. T., Lapp, H., Maddison, W. P., Midford, P. E., et al. (2012). NeXML: Rich, extensible, and verifiable representation of comparative data and metadata. *Systematic biology*, *61*(4), 675–689. doi:10.1093/sysbio/sys025