SOFTWARE TOOL ARTICLE

Phylommand - a command line software package for phylogenetics [version 1; peer review: 2 approved with reservations]

Martin Ryberg

Department of Organismal Biology, Uppsala University, Uppsala, Sweden

V1 First published: 22 Dec 2016, 5:2903
<https://doi.org/10.12688/f1000research.10446.1>
Latest published: 22 Dec 2016, 5:2903
<https://doi.org/10.12688/f1000research.10446.1>

Abstract

Phylogenetics is an intrinsic part of many analyses in evolutionary biology and ecology, and as the amount of data available for these analyses is increasing rapidly the need for automated pipelines to deal with the data also increases. Phylommand is a package of four programs to create, manipulate, and/or analyze phylogenetic trees or pairwise alignments. It is built to be easily implemented in software workflows, both directly on the command prompt, and executed using scripts. Inputs can be taken from standard input or a file, and the behavior of the programs can be changed through switches. By using standard file formats for phylogenetic analyses, such as newick, nexus, phylip, and fasta, phylommand is widely compatible with other software.

Keywords

Phylogeny , Work-flow , Pipeline , Supermatrix , Supertree

This article is included in the **Phylogenetics** collection.

Open Peer Review

Approval Status ? ?

	1	2
version 1	?	?
22 Dec 2016	view	view

1. **Thomas S. B. Schmidt** , European Molecular Biology Laboratory, Heidelberg, Germany
2. **Siavash Mirarab**, University of California, San Diego, San Diego, USA
Niema Moshiri , University of California, San Diego, San Diego, USA

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Martin Ryberg (martin.ryberg@ebc.uu.se)

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2016 Ryberg M. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Ryberg M. **Phylommand - a command line software package for phylogenetics [version 1; peer review: 2 approved with reservations]** F1000Research 2016, 5:2903 <https://doi.org/10.12688/f1000research.10446.1>

First published: 22 Dec 2016, 5:2903 <https://doi.org/10.12688/f1000research.10446.1>

Introduction

The improvement of high throughput sequencing methods, and the ability to produce more and/or longer reads at a cheaper price have allowed the use of these data in new areas of research (Ellison *et al.*, 2011; Jumpponen & Jones, 2009; Lemmon *et al.*, 2012). More or less, automated software pipelines are often an intrinsic part in the development of ways to process and analyze data for new types of research questions.

Phylogenetic analyses are an integral part of many biological studies. Phylommand is a package of four software - treebender, treeator, contree, and pairalign - with capabilities in manipulation and analyses of phylogenetic trees. The functions include rooting, splitting, and comparing trees, calculating parsimony and likelihood scores, as well as performing parsimonious stepwise addition of taxa, nearest neighbour interchange branch swapping, and construction of neighbour joining trees. In addition there are functions, such as calculating decisiveness (Sanderson *et al.*, 2010), MAD scores (Smith *et al.*, 2009), and matrix representation of trees, for use in supermatrix and supertree pipelines. Similar to the Newick utilities software suit (Junier & Zdobnov, 2010), that also act on phylogenetic trees, the phylommand programs are designed to be used on the command line, without the overhead from a graphical interface. Phylommand also accepts inputs from file or standard input, outputs the results to standard output (i.e. usually screen if not redirected), and works without any configuration files or user input (after execution). It is thus made to work in software pipelines.

Both phylommand and Newick utilities can use the newick file format and are therefore compatible, and they complement each other as most of the functions in phylommand are not included in the Newick utilities. Even if phylommand is made for pipelines, each software work independent of any other program (given appropriate input).

Methods

Implementation

Phylommand is written in the C++ programming language and is primarily distributed as source code. Its core utilities only depend on standard libraries to facilitate ease of compilation and use on multiple platforms. However, in addition to standard libraries, treeator may be compiled with the option to optimize model parameters under the maximum likelihood criterion using the NLOpt library (Steven G. Johnson, The NLOpt nonlinear-optimization package, <http://ab-initio.mit.edu/nlopt>); pairalign can be compiled in a parallelized version using pthreads, which may be useful to decrease runtime when doing pairwise alignment. As an addition to the four core programs rudisvg, a rudimentary svg viewer, that depends on the X11 library and a X11 server is included with phylommand (rudisvg).

Operation

The core programs of phylommand, in their basic version, have been successfully compiled and used on OS X 10.10 and Ubuntu Linux 16.04 (including Ubuntu 14.04 on Windows [anniversary edition of Windows 10]) using GNU g++ and make, and on Windows 10 using the same tools in MinGW. The svg viewer rudisvg has been successfully compiled and used on OS X 10.10 and Ubuntu 16.04

(Gnu/Linux; including Ubuntu on Windows). The behavior of the programs in phylommand are controlled by switches, and all programs have an extensive help documentation that can be accessed by the switch --help. Phylogenies can be given in either newick or nexus format. Pairalign reads DNA sequences in fasta format, and treeator reads character matrices/sequences in fasta, phylip, or nexus format (as outputted from for example AliView 1.19; Larsson, 2014) or work on distance matrices. The output formats for phylogenies are newick and nexus, but treebender can also output trees as standard vector graphics (svg). The svg output from treebender can be displayed by rudisvg. Treebender is faster at rooting trees than nw_reroot from Newick utilities (Table 1), but the difference is small compared to how much faster nw_reroot is for the same task compared with packages in interpreted languages such as BioPerl (Perl), APE (R), and ETE (python; Junier & Zdobnov, 2010).

Use cases

In the examples, alignment_file.fst can be replaced by any fasta formatted alignment file, and tree_file.tree (tree_file.trees if including more than one tree) can be replaced by any newick or nexus formatted file. Treebender can easily be used to create a svg image that can be piped into a file:

```
treebender --output svg tree_file.tree > tree_file.svg
```

Treebender can also be used to create monophyletic operational taxonomic units in a tree based on branch lengths (c.f. virtual taxa sunsu Öpik *et al.*, 2009):

```
treebender --cluster branch_length \
--cut_off 0.03 tree_file.tree
```

To view a neighbour joining tree representation of a set of DNA sequences it is possible to use:

```
pairalign -j -n -m -v alignment_file.fst \
treeator -n | treebender --output svg | rudisvg
```

The parsimony score and trees of all nearest neighbour interchange swaps from a topology can be given by:

Table 1. Time to root 10 trees, using one tip as outgroup.

Number of tips	Treebender ¹	nw_reroot ¹
1000	0.051 (0.0012)	0.058 (0.00060)
5000	0.19 (0.0029)	0.2178 (0.0082)
10000	0.33 (0.0041)	0.40 (0.0019)
20000	0.57 (0.0068)	0.96 (0.017)

¹Mean and standard deviation in seconds based on 10 separate runs on a workstation with a Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 24Gb RAM, running Ubuntu 16.04.1, with Linux kernel 4.4.0-34.

```
treebender --nni all tree_file.tree \
-f alignment_file.fst -p
```

Pairalign can be used to get which taxa can be aligned confidently according to MAD scores:

```
pairalign --group alignment_groups \
-v alignment_file_with_taxon_string.fst
```

Contree can be used to draw the support values (e.g. bootstrap support) from a set of trees on a given topology:

```
contree -d tree_file.trees -a tree_file.tree
```

Contree can also be used to get supported conflicts between trees. The output can be either text or svg formatted as html (Figure 1):

```
contree -c 70 --html tree_file.trees
```

Figure 1. Figure of conflict. Part of output from `contree -c 70 --html` (re-formatted so trees are put next to each other and text removed). Tips in clade with more than 70 in support in tree to the left that is in conflict with tree to the right with more than 70 in support are colored green. The tips that cause the conflict in tree to the right are colored red.

These and further examples, and an example of a bash script to do a search for the most parsimonious tree and a Perl script to find groups that are alignable according to MAD scores without a pre-defined taxonomy, are distributed with the source code.

Summary

Phylommand offers an efficient way of manipulating and analyzing phylogenetic trees without the overhead of a graphical interface or specialized command line interpreter. It can be used in both automated (through scripts) or manual work-flows. Since it is made to be compilable with minimum reliance of non standard libraries it is possible to use it on most operating systems including UNIX like systems as OS X and Linux, and Windows. This increases its utility for pipelines that will need to work on different platforms.

Software availability

1. Latest source code available at: <https://github.com/mr-y/phylomand>
2. Archive source code as at the time of publication: <http://doi.org/10.5281/zenodo.200397> (Ryberg, 2016)
3. License: GPL3

Author contributions

The author coded the software and wrote the manuscript.

Competing interests

No competing interests were disclosed.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Acknowledgments

I am grateful to Ding He, Anders Larsson, John Pettersson, and Marisol Sanchez-Garcia for testing phylommand and giving comments on earlier versions of this manuscript, and Anders Larsson for contributions to phylommand's documentation.

References

- Ellison CE, Hall C, Kowbel D, *et al.*: Population genomics and local adaptation in wild isolates of a model microbial eukaryote. *Proc Natl Acad Sci U S A*. 2011; 108(7): 2831–2836.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Jumpponen A, Jones KL: Massively parallel 454 sequencing indicates hyperdiverse fungal communities in temperate *Quercus macrocarpa* phyllosphere. *New Phytol*. 2009; 184(2): 438–448.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Junier T, Zdobnov EM: The Newick utilities: high-throughput phylogenetic tree processing in the UNIX shell. *Bioinformatics*. 2010; 26(13): 1669–1670.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Larsson A: Aliview: a fast and lightweight alignment viewer and editor for large datasets. *Bioinformatics*. 2014; 30(22): 3276–3278.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Lemmon AR, Emme SA, Lemmon EM: Anchored hybrid enrichment for massively high-throughput phylogenomics. *Syst Biol*. 2012; 61(5): 727–744.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Öpik M, Metsis M, Daniell TJ, *et al.*: Large-scale parallel 454 sequencing reveals host ecological group specificity of arbuscular mycorrhizal fungi in a boreonemoral forest. *New Phytol*. 2009; 184(2): 424–437.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Ryberg M: Phylommand - a command line software package for phylogenetics. *Zenodo*. 2016.
[Data Source](#)
- Sanderson MJ, McMahon MM, Steel M: Phylogenomics with incomplete taxon coverage: the limits to inference. *BMC Evol Biol*. 2010; 10: 155.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Smith SA, Beaulieu JM, Donoghue MJ: Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. *BMC Evol Biol*. 2009; 9: 37.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 19 April 2017

<https://doi.org/10.5256/f1000research.11256.r21590>

© 2017 Mirarab S et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Siavash Mirarab

Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, CA, USA

Niema Moshiri

Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, San Diego, CA, USA

The article describes the author's open-source toolkit, "Phylommand," which was designed to perform basic operations on phylogenetic trees from the command line. Phylommand was designed with an emphasis on portability/compatibility (by using standard libraries that enable easy cross-platform compilation), efficiency (by writing in C++ as opposed to higher-level scripting languages), and usability (by providing clean and extensive documentation, both online as well as in the tools themselves).

The article provides a nice summary of what the tools in Phylommand do. However, the reader is referred to the manuals to learn about the exact features implemented in the toolkit; in the paper, only general descriptions of the functions included in each tool are included. Moreover, some of the operations (e.g. mid-point rooting) require non-trivial algorithms, and how the tool approaches these steps is unclear.

Appropriately, the authors make multiple references to their main competitor toolkit, namely Newick Utilities. There is one study comparing the two tools in terms of runtime, but this comparison is limited to a single task (rooting trees). Further, even though Newick Utilities is (to our knowledge) the only command line alternative, there are solutions to many of these problems outside of Newick Utilities in scripting languages (e.g. Dendropy for Python and APE in R). It can be argued that a command line tool has a different type of utility and is perhaps more usable. Nevertheless, since scripting languages are also relatively easy to use, comparisons to these platforms in terms of usability and speed would be informative. At a minimum, a mention would be needed.

The code is in C++ using standard libraries, making it simple to compile and run. We were able to

compile it out-of-the-box with the “make” command on a Mac OS X laptop as well as on a CentOS and an Ubuntu server.

Regarding the algorithmic aspects of the toolkit, we have concerns about efficiency, mainly driven by our examination of the code related to the midpoint rooting (note that other potentially non-trivial functions may also have issues, but we only explored midpoint rooting). As mentioned before, the algorithmic details are not provided in the paper. After digging a bit into the code, we were able to understand the midpoint rooting function (tree.cpp lines 528 through 577), and it appears a heuristic is performed: the algorithm starts with the initial rooting of the tree, and it then iteratively shifts the root to the left or right child to minimize the imbalance between the maximum distance to a left descendant leaf vs. a right descendant leaf. Once the imbalance does not improve beyond a hardcoded threshold (average of the maximum left tip distance and the maximum right tip distance, divided by 10,000), the iterative search terminates and the left and right branches of the “optimal” root are adjusted.

The main problem is the following. This “heuristic” solution is designed for a problem that has a trivial exact linear-time solution. Thus, the heuristic is unnecessary. With one bottom-up traversal and one top-down traversal of the tree, one can find the exact correct midpoint root in linear time. A linear time implementation of the midpoint rooting algorithm can be found here: <https://github.com/uym2/MinVar-Rooting> (runs in under a minute for 200,000 leaves).

The midpoint rooting code raises some questions. First, why is the number 10,000 hardcoded here? There is no restriction in the unit of distance used by trees that can be passed to this tool, so what if the input tree used a unit that resulted in huge integers for branch lengths? It seems as though hard-coding 10,000 could potentially cause issues. Second, this heuristic solution is quadratic-time in the worst case, but as mentioned earlier, an exact solution can be trivially obtained in linear time. Third, when given a tree rooted at a leaf, the algorithm fails (and the tool segfaults); why? This is a valid edge case. Below is the tree we tested:
(B:2000000000,(C:3000000000,D:4000000000)E:5000000000)F:1000000000)A;

We did not investigate any other functions but given our reservations about the correctness of the midpoint rooting function, we have reservations about the correctness of other non-trivial functions performed by Phylommand. Explanations of the algorithms behind the non-trivial functions seem essential to the article.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Phylogenetics

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Reviewer Report 06 February 2017

<https://doi.org/10.5256/f1000research.11256.r19805>

© 2017 Schmidt T. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Thomas S. B. Schmidt

European Molecular Biology Laboratory, Heidelberg, Germany

The present article is a short application note on an open-source phylogenetic software tool, phylomand. The article is very concise and can serve well as a reference for a tool under ongoing development. However, I believe that in its present form, it does not contain all necessary information to (i) fully describe the presented software and to therefore (ii) allow an informed evaluation of the tool.

From the article itself, it remains relatively unclear what the individual programs in phylommand actually do. The pointer to the GitHub page, and the documentation included therein, are quite helpful. As a standalone, however, the article is quite generic and does not provide sufficient answers to the basic questions which (I think) should be addressed in a tool description:

- (i) what does the tool do?
- (ii) how does it do it (implementation details)?
- (iii) why is this important?
- (iv) how does it compare to other solutions?

With regard to the first point, the second paragraph of the article does contain a list of things that phylommand can do, but it does not provide context. I believe that even including a freeze of the GitHub wiki and tool documentation as a Supplement would help the reader to make more sense of the software's functionalities.

Implementation details are provided; although they are quite generic, the necessary dependencies for compilation and additional functionalities (parallelization and and SVG viewer) are pointed out.

The article does not really provide information on why phylommand is an important addition to the list of existing tools, or how it fills an (important) gap in the present toolbox, other than stating that it can be integrated into software pipelines. In particular, it remains unclear why the implementation of exactly these functions (out of the wealth of possible operations on phylogenies) provides an added value over existing solutions. Clearly, phylommand is not a full-fledged standalone phylogenetic software suite, but it provides (efficient) solutions to specific problems; this could be carved out more clearly in the text.

Relating to this, the article in general does not compare phylommand to existing tools, neither conceptually, nor in terms of performance. Newick utilities are mentioned several times, but there is such a wealth of phylogenetic software available, and it remains unclear how phylommand positions itself in that list. Perl-based, R-based and Python-based solutions are mentioned in a half-sentence, but from the article alone, it remains unclear what the "niche" of phylommand among the many existing tools can be. Likewise, I believe that a full assessment of the tool's performance requires more benchmarks than a speed test of one individual function (tree rooting) relative to one competitor. For the core implemented functions, it would be great to get at least some speed and memory benchmarks (relative to a limited set of competing tools).

I was able to download and run phylommand on my OS X system. In general, I commend the author for developing and providing a potentially very useful tool for the community. However, as detailed above, I believe that the present article needs more information to act as a stand-alone reference.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research