PhyloToL version 4.1

Table of Contents

About

PhyloToL is a phylogenomic pipeline aimed at exploring evolutionary hypotheses at ancient (i.e. >100 million year) time scales. The scripts and logic developed over numerous years and conversations within Laura Katz's lab at Smith College. The first versions of the pipeline were written by Jessica Grant [1], and the third version was nearly completely revised by Mario Cerón-Romero [2]. For the current version (v4.0), minor bugs were corrected and the pipeline was translated to python 3. PhyloToL was originally designed for analyses of the eukaryotic tree of life, the flexibility of PhyloToL allows users to add their taxa of interest and to explore hypotheses at varying taxonomic levels.

PhyloToL consists of four major components: Component 1) Gene family (GF) assessment per taxon (i.e. adding taxa to the database); Component 2) Refinement of homologs and gene tree reconstruction, Component 3) Tree-based contamination removal; and Component 4) building of a supermatrix for species tree reconstruction. These components can be executed independently. PhyloToL is written primarily in the Python 3 programming language but it also incorporates Perl, Ruby and Bash custom scripts. PhyloToL only runs through the command line (i.e. there is no a GUI), therefore a minimum knowledge of UNIX is required. PhyloToL was designed to run on a cluster, but can also easily be run on desktop computers with single or multiple threads.

1 Grant, J. R. & Katz, L. A. Building a phylogenomic pipeline for the eukaryotic tree of life - addressing deep phylogenies with genome-scale data. *PLoS currents* **6**, doi:10.1371/currents.tol.c24b6054aebf3602748ac042ccc8f2e9 (2014).

2 Cerón-Romero, M. A., Maurer-Alcalá, X. X., Grattepanche, J. D., Fonseca, M. M. & Katz, L. A. PhyloToL: A taxon and gene rich phylogenomic pipeline. (Submitted).

Back to Table of Contents

Download and dependencies

Distribution: PhyloToL can be obtained by clone or download of the whole package from https://github.com/Katzlab/PhyloTOL

Dependencies: PhyLoToL relies on numerous dependencies, the links for each are provided here. As with any software package, users will have to download all dependencies and install them on their platforms before using PhyloToL.

- Biopython (https://biopython.org/)
- DendroPy (https://dendropy.org/)
- P4 (http://p4.nhm.ac.uk/)
- Bioperl (https://bioperl.org/)
- MAFFT (v7; https://mafft.cbrc.jp/alignment/software/)
- USEARCH (any version; https://www.drive5.com/usearch/)
- Guidance (v2.02; http://guidance.tau.ac.il/overview.html)
- trimAl (v1.3; http://trimal.cgenomics.org/)
- RAxML (v8; https://cme.h-its.org/exelixis/web/software/raxml/index.html)

For Dendropy and P4 be sure of doing the installations for python 3.

Back to Table of Contents

Databases

There are two parts in the databases: 1) the seed dataset and 2) the added taxa dataset. The seed dataset is a group of protein-coding GFs in fasta format that represent the phylogenetic markers of PhyloToL. When the user has high-throughput sequencing data of certain taxa that are not represented in the seed dataset and processed them through PhyloToL, PhyloToL would find in those data the sequences are homologs to the GFs of the seed dataset. Those new homologs from new taxa would constitute the added taxa dataset. For more details on building the adding taxa dataset see adding taxa section.

The databases should follow the folder structure outlined below:

```
DataFiles/
        allOG5Files/
                OG5_126595
        ncbiFiles/
                Am_my_Dpur_XX_Dictyostelium_purpureum.fasta
        BlastFiles/
                Am_my_Dpur_XX_Dictyostelium_purpureum.fasta_1e-
                10keepall_BlastOutall.oneHit
```

The folder “allOG5Files” should contain the seed dataset. Each GF should be represented by a text file named with a unique code and containing the sequences for this GF. In our laboratory we use OrthoMCL data for building this dataset and use the OrthoMCL codes as unique gene family identifiers (e.g. OG5_126595 for actin). The user is free to pick a different system to identify GFs but we recommend keeping the prefix “OG5_” or modifying the scripts accordingly.

The folders “ncbiFiles” and “BlastFiles” represent the added taxa dataset. While the folder “ncbiFiles” contains the actual sequences (e.g., transcriptome, genome or protein sequences), the folder “BlastFiles” contains the results of Blasting the sequences of the new taxa against the seed dataset (see adding taxa section). There should be one file per taxon in both folders.

Back to Table of Contents

Taxa names

We use a 10-digit code system for naming the taxa. The 10-digit code is intended to represent taxonomic placement. For instance, the code for *Plasmodium falciparum* is Sr_ap_Pfal. Here, the Sr_ represents the eukaryotic major clade SAR and Sr_ap_ represents the “minor” clade Apicomplexa.

Formatting taxon names:
MC_mc_Gsp1_*anything*
MC = major clade
mc = minor clade
Gsp1 = 4 letter code for genus, species, ID

See section on ‘formatting taxon names’ for a list of minor clade assignments that we recommend.

Back to Table of Contents

Run PhyloToL – component 1: adding taxa

In order to add new taxa to the database, the user can run the first of the four major components of PhyloToL. This component takes high throughput sequencing data and conducts the following steps: identification and removal of sample bleeding from an Illumina lane, removal of prokaryotic and rDNA sequences, assigning of sequences to GFs and translation of sequences using informed genetic codes. Finally, the output is represented in four files, a fasta file with nucleotide sequences, a fasta file with amino acids (AA file), a BLAST report in tap separated format, and the same report in XML format. The AA file should be placed in the "ncbiFiles" folder and the XML formatted report file should be put in the "BlastFiles" folder.

Dedicated documentation for running this component of PhyoToL is available here

Back to Table of Contents

Run PhyloToL – components 2 & 3: Homology assessment, alignment and tree building, tree-based contamination removal

Standard run

Make sure you have this folder/file structure (Bold: input files)

```
PhyloToL/
        DataFiles/
                allOG5Files/
                ncbiFiles/
                BlastFiles/
                taxaDBpipeline4
                Taxa_test.txt
                GFs_test.txt

        Scripts/
                *PhyloToL scripts
                **PhyloToL package directories
                pipeline_parameter_file.txt
```

*PhyloToL scripts: phylotol.py, phylotol-concleaner.py, phylotol-resumer.py
**PhyloToL package directories: Gene, Taxon and Pipeline.

taxaDBpipeline4: This is a csv file with three columns. The first columns specifies the dataset: om = seed dataset (folder allOG5Files), wg = whole genome (folder ncbiFiles), and nw = no whole genome (also from folder ncbiFiles). The second column specifies if the taxon is an eukaryte (euk) or prokaryote (pro). Finally, the third column contains the 10-digit code. For instance:

```
nw,euk,Am_ar_Enut
wg,pro,Za_em_Mman
om,euk,Op_me_drer
```

Taxa_test.txt: This is a text file that contains the list of taxa that the user wants to run through PhyloToL. For instance:

```
EE_is_Tmar
EE_ka_Rtru
EE_ap_Asig
Ex_ma_Mjak
Am_ar_Enut
Am_ar_Mbal
Am_di_Acas
```

The user can also use text match in the Taxa_test.txt file. For instance, to include all eukaryotes set file as:

```
wg,euks,
nw,euks,
```

GFs_test.txt: This is a text file that contains the list of GFs that the user wants to run through PhyloToL. For instance:

```
OG5_133844
OG5_133879
OG5_128106
```

Set parameters in the file pipeline_parameter_file.txt; navigate to PhyloToL/Scripts/ and type…

```
$ python phylotol.py
```

Running with contamination removal

Make sure you have this folder/file structure (Bold: input files)

PhyloToL/
- DataFiles/
  - allOG5Files/
  - ncbiFiles/
  - BlastFiles/
  - taxaDBpipeline4
  - **GFs_test.txt**
  - **Taxa_test.txt**
  - **Rules.txt**
- Scripts/
  - PhyloToL scripts
  - PhyloToL package directories
  - pipeline_parameter_file.txt

Rules.txt: This is a text file that contains a set of rules for contamination removal that PhyloToL will use for categorizing a sequence as contamination or not. These rules are user-defined with previous knowledge by either manual inspection of a sample of trees, literature or any other methods. A rule can be set as:

```
Sr_rh_Lvor	Op_me		Pl_
```

This will tell PhyloToL to consider as contamination a case in which the taxon Sr_rh_Lvor is nested among either Op_me or Pl_.

There are three modes for contamination removal: mode c1 – from the seed dataset (allOG5Files folder), mode c2 – from the added taxa (ncbiFiles folder), and mode c3 – from both seed dataset and added taxa. For instance, for running contamination cleaning in mode 1, once the rules are set, navigate to PhyloToL/Scripts/ and type…

```
$ python phylotol-concleaner.py ../DataFiles/rules.txt c1
```

Running partially

Depending on the type of study, you might want to run PhyloToL just for collecting candidate gene families (e.g., using it as a tool to test for homology) or for collecting homologs but not trees (e.g., if you want to try another tree inference tool than RAxML). The user can run PhyloToL in two different modes: “ng” and “nr”, respectively.

```
$ python  phylotol.py ng (Runs PhyloToL until Guidance)
$ python  phylotol.py nr (Runs PhyloToL until RAxML)
```

Restarting

It is also possible to run PhyloToL up to guidance (mode ng) and later resume the run and either produce post-guidance files and trees or only post-guidance files (with option "nr" - no raxml)

```
$ python phylotol-resumer.py path_to_working_directory (see below)
$ python phylotol-resumer.py path_to_working_directory nr (no tree)
```

If only post-guidance files were produced but not trees, then the user can resume the run and produce trees in this way:

```
$ python phylotol-resumer.py path_to_working_directory
```

In order to run partially and re-start, PhyloToL requires a working directory. If the user ran PhyloToL up to guidance and wants to resume, the working directory will be a folder (e.g., out) with all pre-guidance files inside. The folder/file structure will be as follows (bold: working directory):

PhyloToL/
- DataFiles/
  - allOG5Files/
  - ncbiFiles/
  - BlastFiles/
  - taxaDBpipeline4
  - GFs_test.txt
  - Taxa_test.txt
- **out/**
  - **pre-guidance files**
- Scripts/
  - PhyloToL scripts
  - PhyloToL package directories
  - pipeline_parameter_file.txt

If the user ran PhyloToL up to RAxML and wants to resume (to produce trees), then, the structure has to be kept as follows (bold: working directory):

PhyloToL/

```
DataFiles/
        allOG5Files/
        ncbiFiles/
        BlastFiles/
        taxaDBpipeline4
        GFs_test.txt
        Taxa_test.txt
out/
        out_resume/
                GFs_test_results2keep/
                        Pre-guidance files
                        Post-guidance files
Scripts/
        PhyloToL scripts
        PhyloToL package directories
        pipeline_parameter_file.txt
```

out/, **out_resume**, **GFs_test_results2keep/**, **Pre-guidance files**, **Post-guidance files**

Back to Table of Contents

Running PhyloToL – fourth component (supermatrix)

This component can be run for choosing orthologous sequences and produce alignments for concatenation. In order to do this, the option “concatAlignment = y” should be set in the file “pipeline_parameter_file.txt”. Then PhyloToL is run as shown under “Quick start” (see above). Once PhyloToL has finished type:

```
python concatenateFastas.py path_to_alignments_for_concatenation
```

This will produce a supermatrix that can be used for species tree building. Alternatively, users can export alignments here and use third party tools to concatenate.

Back to Table of Contents

Cleaning intermediary files

PhyloToL has by default a mechanism to clean all intermediary files and re-structure the folders once the process is done. If the users want to explore the intermediary files, they can comment (#) the line “Utilities.cleaner(testPipelineList, PathtoFiles, PathtoOutput)” from the script phylotol.py. If later the users want to clean those intermediary files and re-structure the folders, they can do it with the command…

```
python phylotol.py cl
```

This command can also be used after stopping a process due to error or after any type of circumstances that require a re-structure of the original files/folders.

NOTE: Don’t comment the line mentioned above when performing contamination cleaning. This process is a loop that requires the intermediary files removal method in each iteration.

Back to Table of Contents

Formatting taxon names: Recommended codes for major and minor clades of eukaryotes:

Though users can easily alter and add codes, we provide a key for codes as we developed them. Again, our key recommendation is that users do not alter structure of codes:

MC_mc_Gsp1_*anything*
MC = major clade
mc = minor clade
Gsp1 = 4 letter code for genus, species, ID

Code	**Major Clade**	**minor clade**
Am_ar	Amoebozoa	Archamoebae
Am_di	Amoebozoa	Discosea
Am_my	Amoebozoa	Mycetozoa
Am_hi	Amoebozoa	Himatismenida
Am_is	Amoebozoa	incertaesedis
Am_th	Amoebozoa	Thecamoebida
Am_tu	Amoebozoa	Tubulinea
Am_va	Amoebozoa	Vannellidae
EE_ap	Orphans	Apusozoa
EE_br	Orphans	Breviatea
EE_cr	Orphans	Cryptophyta
EE_ha	Orphans	Haptophyceae
EE_is	Orphans	incertaesedis
EE_ka	Orphans	Katablepharidophyta
Ex_eu	Excavata	Euglenozoa
Ex_fo	Excavata	Fornicata
Ex_he	Excavata	Heterolobosea
Ex_is	Excavata	incertae sedis
Ex_ja	Excavata	Jakobida
Ex_ma	Excavata	Malawimonadidae
Ex_ox	Excavata	Oxymonadida
Ex_pa	Excavata	Parabasalia
Op_ch	Opisthokonta	Choanoflagellida
Op_fu	Opisthokonta	Fungi
Op_ic	Opisthokonta	Ichthyosporea
Op_is	Opisthokonta	incertae sedis
Op_me	Opisthokonta	Metazoa
Op_nu	Opisthokonta	Nucleariidae and Fonticula
Pl_gl	Plantae	Glaucophytes
Pl_gr	Plantae	Green algae
Pl_rh	Plantae	Red algae

Sr_ap	SAR	Apicomplexa
Sr_ch	SAR	Chromerida
Sr_ci	SAR	Ciliates
Sr_di	SAR	Dinoflagellates
Sr_is	SAR	incertae sedis
Sr_pe	SAR	Perkinsea
Sr_rh	SAR	Rhizaria
Sr_st	SAR	Stramenopiles

Back to Table of Contents

Frequently asked questions

1. Is there any graphical interface available?

No, PhyloToL is a command-line tool.

2. What type of computer can run PhyloToL?

The most basic computers in which we have tried PhyloToL is a Core i7 MacBook pro (late 2013) with 4 cores and 16 GB of RAM and a Core i5 iMac (late 2015) with 4 cores and 8 GB of RAM. The most powerful computers in which we have tried PhyloToL have been an Intel Xeon server E5-2699 with more 30 cores and 500 GB of RAM an Intel Xeon iMac Pro (2017) with 18 cores and 128 GB of RAM. In theory any computer.

3. What operator system is needed to run PhyloToL?

We have tested PhyloToL in both MacOS and Linux (Ubuntu and centOS). PhyloToL may also work in other operator systems such as Windows if the user has access to the command line. Also, the user needs to check the system requirements of the third-party tools that PhyloToL needs (see “Download and dependencies” in the manual).

4. What level of experience with the command-line is needed?

The user needs to be able to install software by command line and use some commands to access files and move through folders. For taking advantage of the high-flexibility of PhyloToL for different type of studies, the user would need programming experience.

5. In which programming language was PhyloToL written?

PhyloToL v4.1 is mostly based on Python 3 (about 90%). Some scripts are written in Perl, Bash, Ruby. The current version can potentially still run with Python 2.7 but we recommend to always use Python 3.

6. How long does a PhyloToL run take?

It depends in many factors like number of threads, RAM, type of processor, number of taxa included and complexity of the genes analyzed. For instance, a dataset composed of 50 eukaryotic taxa and 20 genes can be processed through PhyloToL in an average time of 3-4 days in iMac with 4 cores and 8 GB of RAM.

Some dependencies in PhyloToL don’t paralyze efficiently (e.g., some processes in Guidance 2.02 don’t parallelize and the free version of Usearch limits the number of threads that can be used). Therefore, PhyloToL is set to use only one thread. The user can modify easily the scripts Utilities.py in order to use more threads. Using more threads in a single run will increase the efficiency of PhyloToL, but based on our experience a more efficient way of running PhyloToL in many threads is setting many instances and use only one thread per instance.

7. What taxa can be used and what kind of genes?

PhyloToL was designed for exploring the evolutionary history of eukaryotes, therefore, we recommend datasets that include either taxa from the whole tree of life or only eukaryotes. PhyloToL can also be used in datasets that only include Bacteria and/or Archaea, but other phylogenic pipelines designed specifically for these taxa can be more efficient. Likewise, datasets that include shallow levels of diversity of widely known taxa (e.g., just primates, just angiosperms) can be more efficiently analyzed in other phylogenomic pipelines.

The genes that can be analyzed in PhyloToL are protein coding genes.

8. Is PhyloToL only used for species tree inference?

No, PhyloToL is highly flexible and can be used for exploring the evolution of gene families, metagenomic studies, checking the quality of high throughput sequencing data and removing contaminant sequences, among many other types of experiments. The manual contains detailed information of how to run the different components of PhyloToL.

9. What input formats are accepted by PhyloToL?

PhyloToL have multiple entry points. These are the different formats needed by every PhyloToL component.

a) First component (adding taxa):
   - Transcriptomes: Assembly file (.fasta)
   - Genomes: Annotated GenBank file (.gff)
   - Protein/EST: Sequences (.fasta)

b) Second and third components (Homology assessment, alignment and tree building and tree-based contamination removal):
   - Database of gene families: Sequences (.fasta)
   - For added taxa with the first component of PhyloToL: The files containing the sequences (.fasta) and the Usearch report (.xml) generated with the first component of PhyloToL.
   - List of genes (.txt)
   - List of taxa to be included in the analysis (.txt)

Back to Table of Contents