

Phylo* File Processing and Automation Using PirANHA



Quick Guide for the Impatient

PirANHA provides a set of tools for automating file processing and analysis in (phylo*=) phylogenetics, phylogenomics, and phylogeography.

NOTE: By convention, in all content we refer to this software package as "PirANHA" and we write the name of the main script in the package as `piranha`.

Downloading, installing, and upgrading PirANHA (on macOS or Linux) is easy thanks to [Homebrew](#) distribution.

Homebrew install

[See here](#).

Two Homebrew Install Options:

Recommended:

- Development (cutting edge) Homebrew install 'by-hand':

```
# Install:
brew tap justincbagley/homebrew-tap ;
brew update ;
brew install --HEAD piranha ;
piranha -i ;
```

- Regular Homebrew install 'by-hand' (stable, but may not be up to date):

```
# Install:
brew tap justincbagley/homebrew-tap ;
brew update ;
brew install piranha ;
```

Updating PirANHA

Recommended:

- Upgrading development (cutting edge) version:

```
# Uninstall:
brew uninstall piranha && brew untap justincbagley/homebrew-tap ;

# Upgrade:
brew tap justincbagley/homebrew-tap ;
brew update ;
brew install --HEAD piranha ;
piranha -i ;
```

Help

Get the help text and a list of functions available in PirANHA like so (notice `<TAB>` for dynamic completion):

```
piranha -h
piranha -f <TAB>
piranha -f list
piranha -s|--shortlist
```

Get the help text for a particular function:

```
piranha -f <function> -h
```

Example usage

Convert between DNA alignment formats like so:

```
# Convert FASTA to PHYLIP:

piranha -f FASTA2PHYLIP -f 1 -i <inputFASTA> -k 1 -v 1 # Single PHYLIP file
piranha -f FASTA2PHYLIP -f 2 -k 1 -v 1 # Multiple PHYLIP files

# Convert FASTA to VCF:

piranha -f FASTA2VCF -i <inputFASTA> -o <output>

# Convert Mega to PHYLIP:

piranha -f Mega2PHYLIP -i <inputMega> -k 1 # Single Mega file
piranha -f Mega2PHYLIP -m 1 -k 1 # Multiple Mega files
# Convert NEXUS to PHYLIP:

piranha -f NEXUS2PHYLIP -i <inputNEXUS> -v 1 # Single NEXUS file

# Convert PHYLIP to FASTA:

piranha -f PHYLIP2FASTA -i <inputPHYLIP> -k 1 # Single FASTA file
piranha -f PHYLIP2FASTA -m 1 -k 1 -v 1 # Multiple FASTA files

# Convert PHYLIP to Mega:

piranha -f PHYLIP2Mega -i <inputPHYLIP> -k 1 # Single PHYLIP file
piranha -f PHYLIP2Mega -m 1 -k 1 # Multiple PHYLIP files

# Convert PHYLIP to NEXUS:

piranha -f PHYLIP2NEXUS -i <inputPHYLIP> -p <partitionsFile> -f NEX # Single PHYLIP file
piranha -f PHYLIP2NEXUS -m 1 # Multiple PHYLIP files
```

Concatenate DNA sequence alignments (e.g. genes) like so:

```
# Create <taxonNamesSpaces> file with getTaxonNames function (creates file '<numTips>_taxon_names_spaces.txt'):

piranha -f getTaxonNames -n <numTips>

# Concatenate PHYLIP alignments (e.g. 1 per gene):

piranha -f concatenateSeqs -t <numTips>_taxon_names_spaces.txt

# Complete (fill in missing individuals) and concatenate PHYLIP alignments:

piranha -f completeConcatSeqs -t <numTips>_taxon_names_spaces.txt
```

Trim DNA sequence alignments like so:

```
# Use trimSeqs to trim single PHYLIP alignment with default settings and PHYLIP output:

piranha -f trimSeqs -i <inputPHYLIP> -o phylip
piranha -f trimSeqs --input <inputPHYLIP> --output phylip

# Use trimSeqs to trim multiple PHYLIP alignments with default settings and PHYLIP output:

piranha -f trimSeqs -m 1 -o phylip
piranha -f trimSeqs --multi 1 --output phylip

# Use trimSeqs to trim PHYLIP alignments with custom gap handling and sequence conservation settings for trimAl:

piranha -f trimSeqs --multi 1 --output phylip --cons 60 --gt 0.1

# Use trimSeqs to trim PHYLIP alignments stringently, removing all sites with gaps:

piranha -f trimSeqs --multi 1 --output phylip --nogaps 1

# NOTE: You may also switch output formats to FASTA (--output fasta) or NEXUS (--output nexus) formats.
```

Phase consensus sequences from HTS (e.g. targeted sequence capture) using reference:

```
# Phase alleles with default settings (creates intermediate files and final, unaligned phased FASTAs):

piranha -f phaseAlleles -i <input> -o <output> -r <reference>

# Phase alleles while masking reference indels (insertions/deletions) in final, unaligned phased FASTAs:

piranha -f phaseAlleles -i <input> -o <output> -r <reference> -m 1
```

Run standard evolutionary analysis programs (run with `-h` for help text first):

```
# Run BEAST:

piranha -f BEASTRunner

# Run aaai:

piranha -f dadRunner
piranha -f dadUncertainty

# Run RAXML:

piranha -f MAGNET
piranha -f RAXMLRunner

# Run RogueNaRok:

piranha -f RogueNaRokRunner

# Run SNAPP:

piranha -f SNAPPRunner
```

Conduct post-processing of results from standard evolutionary analysis programs (run with `-h` for help text first):

```
# Process output from BEAST:

piranha -f MLEResultsProc
piranha -f BEASTPostProc

# Process output from ExaBayes:

piranha -f ExaBayesPostProc

# Process output from MrBayes:

piranha -f MrBayesPostProc
```

Use the documentation links in the sidebar at *right* to navigate this documentation and learn more about PirANHA, and [contact the author for technical support](#) or [raise an issue](#).

PirANHA Publications

We are working on a paper describing PirANHA while we develop towards major release v1.0 (hopefully later this year, in 2020). However, the alpha pre-release versions of PirANHA have been used in several of our publications, including:

- Bagley, J.C., Hickerson, M.J. and Johnson, J.B., 2018. Testing hypotheses of diversification in Panamanian frogs and freshwater fishes using hierarchical approximate Bayesian computation with model averaging. *Diversity*, 10(4), 120.
- Bagley, J.C., Mayden, R.L. and Harris, P.M., 2018. Phylogeny and divergence times of suckers (Cypriniformes: Catostomidae) inferred from Bayesian total-evidence analyses of molecules, morphology, and fossils. *PeerJ*, 6, p.e5168.
- Bagley, J.C., Uribe-Convers, S., Carlsen, M., Muchhala, N., 2020. Utility of targeted sequence capture for phylogenomics in rapid, recent angiosperm radiations: Neotropical *Burmeistera* bellflowers as a case study. *Molecular Phylogenetics and Evolution*. Available online. PubMed: <https://www.ncbi.nlm.nih.gov/pubmed/32081762>. doi: <https://doi.org/10.1016/j.ympev.2020.106769>.

Contact Info

Questions? Comments? Concerns? Bug fix requests? Feature requests for existing functions, or requests for new functions? If so, please get in touch [here](#).

License

All code within **PirANHA v0.4a3** repository is available "AS IS" under a 3-Clause BSD license. See the [LICENSE](#) file for more information.

Citation

If you use scripts from this repository as part of your published research, please cite the repository as follows (also see DOI information below):

- Bagley, J.C. 2020. PirANHA v0.4a3. GitHub repository, Available at: <http://github.com/justincbagley/PirANHA>.

Alternatively, provide the following link to this software repository in your manuscript:

- <https://github.com/justincbagley/PirANHA>

DOI

The DOI for PirANHA, via [Zenodo](#) (also indexed by [OpenAIRE](#)), is as follows: `DOI: 10.5281/zenodo.596766`. Here are some examples of citing PirANHA using the DOI:

Bagley, J.C. 2020. PirANHA v0.4a3. GitHub package, Available at: <http://doi.org/10.5281/zenodo.596766>.

Bagley, J.C. 2020. PirANHA. Zenodo, Available at: <http://doi.org/10.5281/zenodo.596766>.

Uninstall

Uninstalling Homebrew `piranha`

```
# Uninstall:
brew uninstall piranha && brew untap justincbagley/homebrew-tap ;
if [[ -f /usr/local/bin/local_piranha ]]; then rm -rf /usr/local/bin/local_piranha ; fi
if [[ -f /usr/local/bin/brew_piranha ]]; then rm -rf /usr/local/bin/brew_piranha ; fi
```

December 14, 2020 - Justin C. Bagley, Jacksonville, AL, USA

[Next \(Introduction\) >>](#)