Version dated: November 19, 2016

Phyx: Phylogenetic tools for Unix

Joseph W. Brown†, Joseph F. Walker†, and Stephen A. Smith *

[1] *Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, Michigan, 48109, USA*

Author Emails: josephwb@umich.edu, jfwalker@umich.edu and eebsmith@umich.edu.

What is Phyx?

Phyx (Pronounced however you would like) is a set of data analysis programs modeled after POSIX-style tools, to help them be easily incorporated in bioinformatic pipelines. The majority of Phyx programs focus on phylogenetic analyses, which includes a variety of programs to clean data matrices, simulate data and perform basic phylogenetic analyses. Phyx is an ever expanding library of programs and any request can be made to the authors. All programs are open source and Phyx operates under the licence: GPL https://www.gnu.org/licenses/gpl-3.0.html

Installing Phyx

Installation Instructions may be found at: https://github.com/FePhyFoFum/phyx

List of programs and features of Phyx

The following is a list of the programs that are currently available, for all options a program offers type the program then (-h) and a list of features will be displayed.

repeatability

Phyx will track the programs and commands input to a file called ”phxy.log”, this allows users to see exactly what settings they used to run a program and can help others replicate experiments.

Piping

Phyx provides the ability of programs to pipe the output of one into another, allowing for more efficient processing of data. An example of this would be to if someone wanted to perform a codon alignment, then clean all missing data and finally make a rough neighbor joining tree.

```
pxaa2cdn -a amino acid alignment -n nucleotide alignment || pxclsq -p 0.0
|| pxnj -n 3 -o output tree file
```

pxaa2cdn

Often times a coding DNA alignment does not end up with the data divided into sets of three (codons) and as a result this may introduce bias into the analysis or make positive selection tests difficult. This program allows the user to first align the amino acid alignment, then using the alignment the user inputs the corresponding nucleotide sequences and the program will give the codon aligned sequence.

```
pxaa2cdn -a "AA Alignment" -n "Unaligned nucleotides" -o "Output (Codon Alignment)"
```

pxbdfit

Diversification has becoming a rapidly expanding field and as a result tools to analyze the data are essential. This program will fit a diversification model to a tree. The model which is chosen with (-m) may be either the default a birth-death model (bd) or a yule model (yule). The program will return the model parameters (b,d,r,e), likelihood, aic and tree statistics.

```
pxbdfit -t tree_file -m "yule"
```

pxbdsim

Birth death processes are an essential part to understanding diversification and simulation gives researchers the ability to study these processes using known birth and death values. The user is allows to specify the number of extant taxa (-e) or the simulation may be run for a given amount of time (-t), this can be taken further by allowing the user to incorporate the number of extinct taxa with (-s).

```
pxbdsim -e 100 -s -b 1 -d 0.5 -o output_tree_file
```

pxboot

Using a variety of statistical methods for evaluating certainty of phylogenetic trees is essential as all methods have both positives and negatives associated with them. This program will allow the user to create datasets for two of the most commonly used methods (Bootstrap and Jackknife). The proportion of data to be incorporated in a Jackknife may be specified with (-f) and a random seed may be specified with (-x).

```
pxboot -s alignment -x random seed -f 50 -o output of 50% jackknife
```

pxbp

Analyzing similarities among phylogenetic trees has become a growing part of phylogenetics, especially in the field of phylogenomics to determine if a clade is found in a gene tree and in a species tree. This program allows the user to print out all the bipartitions that are in a phylogenetic tree.

```
pxbp -t tree_file -o bipartitions
```

pxbpsq

Would someone mind taking a look at this I think there is an error for the last bipartition

pxcat

When developing a supermatrix for an analysis concatenation of the genes is essential and manual programs that perform this for thousands of genes at once are capable of saving users a lot of time importing each gene into visualization software. This program allows the user to specify a variety of different file types to be concatenated together and can print a user specified partition file (-p).

```
An example where the sequences to be concatenated are in a variety of formats
pxcat -s *.fa *.phy *.nex  -p partition file -o output matrix
```

pxclsq

Having a large amount of missing data in a column of a supermatrix, may be due to errors in alignment or a variety of other factors. Therefore, removing highly ambiguous columns of data may help better estimate a model of evolution for a dataset. This program allows the user to specify a proportion of data that is allowed to be missing (-p), it is important to specify if Amino Acids are being used with the option (-a).

```
An example to clean an amino acid sequence down to only columns with at least 50% data.
pxclsq -s sequence -p 0.5 -a
```

pxconsq

This program will allow the user to get the consensus sequence from different file types

```
pxconsq -s sequence
```

pxcontrates

Comparing continuous characters across phylogenies provides a valuable tool for understanding the evolution of such characters. Two of the most commonly used models are Brownian and OU models, and this program can be used to estimate the rate of character evolution. The input for this is a fasta file where instead of nucleotide data there is tab delimited character states and a tree file for this to mapped onto. The program may then perform an ancestral state reconstruction (-a 0 or default) or test for model fit between OU and Brownian motion (-a 1).

```
Example model test for a set of characters across a tree
pxcontrates -c contrates_file.txt -t contrates_tree.tre -a 1
```

pxfqfilt

Filtering based on a certain quality score is essential for processing raw fastq reads from next generation sequencing data. This program allows the user to specify a mean quality score (-m) and filter based on that quality score.

```
pxfqfilt -s fastq sequence -m 10
```

pxlog

This program is an MCMC log manipulator and concatenator.

An example of usage for this is:

Resamples parameter or tree MCMC samples using some burnin and thinning across an arbitrary number of log files. NOTE: resampling parameters are in terms of number of samples, not number of generations. To determine the attributes of the log files, you can first use the -i (–info) flag:

```
pxlog -t tree_files -i
```

and then sample accordingly:

```
pxlog -t tree_files -b some_burnin -n some_thinning
```

pxlssq

Due to the high variability that is found in sequences and in data matrices it is often important to find out various aspects (eg. amount of missing data, proportion of character etc...). This program will allow provide the user with a variety of these aspects of a the data and provide an easy way to summarize sequence data and concatenated matrices.

```
pxlssq -s Alignment
```

pxlstr

Aspects of trees often provide a large amount of information regarding the behavior of the data that was used to create the tree. This program allows the user to uncover many of these aspects, such as tree depth, rtvar etc...

```
pxlstr -t Tree.tre
```

pxmrca

This program will provide the information regarding the most recent common ancestor. Specifically the user provides the species in a clade of interest using an MRCA file formatted as follows: MRCANAME = tip1 tip2

```
pxmrca -t tree -m mrcafile
```

pxmrcacut

With extremely large trees becoming more common place (species level, gene families etc...) it is useful to focus on certain clades. This program allows the user to specify tips of a clade (-m), only two are required and will remove a newick for the smallest clade that encompasses both species specified.

mrca file format: MRCANAME = tip1 tip2

```
pxmrcacut -t tree -m mrca_file
```

mrcaname

This program allows the user to label the internal nodes with clade names. The program takes in an mrca file in the same format as (pxmrca and pxmrcacut)

```
pxmrcaname -t tree -m mrca_file
```

pxnj

This program will create a basic neighbor joining tree.

```
pxnj -s Alignment
```

pxnw

This program will do pairwise alignments using the needleman-wunsch algorithm. It also allows alignment scores to be analyzed and various scoring matrices to be used (-m).

```
pxnw -s Alignment
```

pxrecode

This program will recode dna to specify only transitions/tranversions (RY coding).

```
pxrecode -s SeqFile
```

pxrevcomp

This program will provide the reverse complement of dna sequences in a file.

```
pxrevcomp -s SeqFile
```

pxrls

This program allows the user to rename taxa by giving a sequence file and specifying a list of current name (-c) and new names (-n), files must match in order of current and new.

```
pxrls -s SeqFile -c CurrentNames -n NewNames
```

pxrlt

This program provides a way to re-label the tips of trees by specifying a list of current name (-c) and new names (-n), files must match in order of current and new.

```
pxrlt -s TreeFile -c CurrentNames -n NewNames
```

pxrms

This program will remove sequences from a sequence file, either by typing them on the command line using (-n) or by inputing a file using (-f).

```
Example to remove sequences called Taxon1 and Taxon2
pxrms -s SeqFile -n Taxon1,Taxon2
```

pxrmt

This program will remove tips from a tree file, either by typing them on the command line using (-n) or by inputing a file using (-f).

```
Example to remove tips called Taxon1 and Taxon2
pxrms -t Tree -n Taxon1,Taxon2
```

pxrr

This program will re-root a tree file, based on specified the outgroups (-g), or the program can unroot an already rooted tree. By default this program re-roots the trees based off of given outgroup(s) (alternatively, trees can be un-rooted through using the -u flag). If not all the outgroups are found in the tree the program will re-root the tree based on the outgroups that are available. It provides a useful tool for re-rooting thousands of trees which can then be used for analyzing gene discordance across phylogenies.

```
Example: pxrr -t Tree -g outgroup1,outgroup2
```

pxs2fa and pxs2phy and pxs2nex

This programs are all designed in a similar vain, with the ability to convert a file from its current format to fasta, phylip or nexus respectively. You may also specify if you would like to have the output in uppercase with the option (-u).

```
Example: pxs2* -s file_of_some_format
```

pxseqgen

This is a sequence simulator that allows the user to give a tree and specify a model of evolution and sequences will be generated for that tree under the model. Some features are that it allows for the model of evolution to change at nodes along the tree using the (-m) option. The program also allows the user to specify rate variation through a value for the shape of the gamma distribution with the (-g) option and the user is able to specify the proportion of invariable sites the would like to include using the (-i) option. Other options can be found from the help menu by typing (-h) after the program.

The sequence simulator features have been thoroughly tested except the multimodel simulation which is still under active development and has not been thoroughly tested to the developers comfort!

For multimodel simulations it is easiest to print out the node labels on your tree originally using the (-p) option. Once you know the nodes that you would like the model to change at you can specify these nodes on the input using the (-m) option. An example if you wanted two models of evolution on your tree one for the tree and one where it changes at node two, you would enter the command as follows.

```
if the model you want for the tree is: (.33,.33,.33,.33,.33) where values
correspond to (A<->C,A<->G,A<->T,C<->G,C<->T,G<->T)

and the model you want to change to at node two is: (.30,.30,.20,.50,.40)
where values correspond to (A<->C,A<->G,A<->T,C<->G,C<->T,G<->T)

The command would be as follows:

pxseqgen -t tree_file -o output_alignment
-m A<->C,A<->G,A<->T,C<->G,C<->T,G<->T,Node#,A<->C,A<->G,A<->T,C<->G,C<->T,G<->T
```

```
pxseqgen -t tree_file -o output_alignment
-m .33,.33,.33,.33,.33,.33,2,.3,.3,.2,.5,.4,.2
```

pxsm0

This program is designed to calculate a selection model

pxsm2a

pxstrec

This is a program that does some ancestral state reconstruction and stochastic mapping of categorical characters. There are a number of options and the requirement for a control file. The control file can be as simple as ancstates = all which designates that you want ancestral states calculated for each node. The can then be output on a tree in a file given by an -o FILE option. If you only want to look at particular nodes, these can be designated in the control with the mrca = MRCANAME tipid1 tipid2. Then the MRCANAME can be given at the ancstates = MRCANAME. If you would like stochastic mapping with the time in the state mapped you can use the same format but instead of ancstates you would put stochtime. For stochastic number of events stochnumber or ‘stochnumber any. For the stochastic mapping, you will need to designate an MRCA or MRCAs (not all). Multiple can be separated by commas or spaces. You can output these to a file with -n for number of events, -a for the total number of events, and -m for the duration.

```
pxstrec -d test.data-t tree_file -c config -o test.tre.anc
```