Phyx

Phyx contains a large number of phylogenetics programs written in C++ designed to ease workflows for phylogenomic analyses and simulations. The program outputs can be piped immediately from one program to another or can be run individually with output to a file. This manual is designed to provide an overview of what the programs may be used for along with examples, however, it is by no means completely comprehensive and all program options can be checked by running the program name with the option (-h).

Programs

Exploratory programs

pxlssq: lists sequence attributes
pxlstr: print out tree attributes

Sequence manipulation programs

pxaa2cdn: converts amino acid alignment and unaligned nucleotide to codon alignment
pxboot: sequence alignment bootstrap or jackknife resampler.
pxcat: concatenate a large number of sequence files
pxclsq: clean up sequences for missing data
pxfqfilt: fastq filtering program
pxrecode: sequence recoder
pxrevcomp: reverse complementing program
pxrms: removes taxa from a fasta file
pxs2fa: changes file format to fasta
pxs2nex: changes file format to nexus
pxs2phy: changes file format to phylip
pxvcf2fa: convert vcf file to fasta alignment
pxtlate: translates a nucleotide file to amino acid

Tree manipulation programs

pxbp: print out bipartitions of a tree file
pxmrca: gives information about the mrca
pxmrcacut: removes based off of mrca
pxmrcaname: adds names to the mrca
pxnni: nearest neighbor interchange program
pxrmt: remove specified taxa from a tree
pxrr: reroot/unroot a phylogenetic tree
pxt2new: changes tree to newick format
pxtscale: tree rescaling

Sequence analysis programs
pxbpsq: print out bipartitions from a sequence file
pxnw: pairwise alignment using the Needleman-Wunch algorithm
pxsw: pairwise alignment using the Smith-Waterman algorithm
pxconsq: Produce consensus sequence of an alignment

Tree inference programs
pxnj: a basic neighbor joining program designed for very large datasets
pxupgma: a basic UPGMA program

Simulation programs
pxbdsim: birth-death simulator
pxseqgen: sequence simulator, with the ability to simulate under multiple models

Comparative methods programs
pxbdfit: fit a birth death model to a phylogenetic tree
pxcontrates: continuous character estimation using Brownian motion and OU
pxstrec: Ancestral state reconstruction analysis and stochastic mapping program

Bayesian
pxlog: a MCMC log manipulator/concatenator

Example of command line interface

Phyx is designed to perform most tasks through command line and as a result it can be important to act on a large number of files. An easy way to do this is to use a for loop in linux such as cleaning multiple files using pxclsq.

Example Folder: Examples/CommandLineExample

```
for x in *.aln; do pxclsq -s $x -p 0.3 -o $x.cln; done
```

Example of piping from one program to next

Phyx allows for the output of one program to be the input of the next program, allowing for a more streamlined process.

Example Folder: Examples/pxaatocdn_example/

```
pxaa2cdn -a AminoAcid.fa -n Nucleotide.fa | pxclsq -p 0.0 | pxnj -n 3 -o output_tree.tre
```

Exploratory programs

pxlssq

Prints a sequence summary of a file

Options	Function
-s (required)	input seq file, stdin otherwise
-i	output stats for individual sequences
-n	return the number of sequences
-c	return the number of characters (only if aligned) - for unaligned seqs, use with -i flag
-l	return all taxon labels (one per line)
-p	force interpret as protein (if inference fails)
-a	return whether sequences are aligned (same length)
-f	return character state frequencies
-o (default to stdout)	Output file

pxlssq -s alignment_file

pxlstr

This program will give a summary of a given tree file

Options	Function
-t (required)	input tree file, stdin otherwise
-r	return whether the tree is rooted
-a	return the height of root (must be rooted and ultrametric)
-n	return the number of terminals
-u	return whether tree is ultrametric
-b	return whether tree is binary
-l	return the length of the tree
-i	return all tip labels (one per line)
-o (default to stdout)	Output file

pxlstr -t tree.tre

Sequence manipulation

pxaa2cdn

This program will take an Amino Acid alignment and an unaligned nucleotide file, and align the nucleotide file by the amino acid sequence. It does not require the two files to have sequences in the same order and will alert you if a sequence is found in one and not the other.

Example Folder: Example/pxaa2cdn_examples/

Option	Function
-a (required)	Aligned amino acid file
-n (required)	Unaligned nucleotide file
-o (default to stdout)	Output file

pxaa2cdn -a AminoAcid.fa -n Nucleotide.fa -o Codon.fa

pxboot

This is a program alignment bootstrap or jackknife resampling.

Example Folder: Example/pxboot_examples/

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output sequence file
-p (default none)	File listing partitions: Name = Start-Stop
-f (if not used will run bootstrap)	JackKnife percentage
-x (default clock)	Random seed number

For bootstrap

pxboot -s Alignment -p parts

For Jackknife

pxboot -s Alignment -f 0.8

pxbpsq

This program is designed to give bipartitions from a sequences file. Output will be in order A | C | G | T

Option	Function
-s (required)	Sequence file, stdin otherwise
-t (required)	Tree file to go with sequence file
-o (default stdout)	Output file

pxbpsq -s sequence.fa -t tree.tre

pxcat

This is a program designed to concatenate large datasets consisting of thousands of genes.

Example Folder: Example/pxcat_examples/

Options	Function
-s (required)	List of input sequence files
-o (default stdout)	Output sequence file
-p (default none)	Partition file in RAxML form gene# = Start-Stop

pxcat -s *.phy *.fas -p partition_file -o Concatenated.fa

pxclsq

This program will clean sequence data allowing for removal of ambiguous sites

Example Folder: Example/pxclsq_examples/

Options	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-p (default = 0.5)	Percent that is allowed to be missing (given as decimal)

Example allowing up to 30% of data to be missing

pxclsq -s Alignment -p 0.3

pxconsq

Produces a consensus sequence for an alignment

Example: Examples/pxconsq_example/

Options	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file

Creates a consensus sequence from an alignment

pxconsq -s consq_test.nex

pxfqfilt

Filters fastq files by a mean value

Example Folder: Examples/pxfqfilt_examples

Options	Function
-m (default 30)	Mean value under which to filter
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file

pxfqfilt -s fqfilt_test.fastq

pxnw

Performs multiple pairwise alignments and gives scores using the needleman-wunch algorithm.

Example Folder: Examples/pxnw

Options	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output of alignment scores
-a (default off)	Outputs the alignments
-t (default dna)	Molecule type (DNA=0, AA=1)
-m (default EDNAFULL or BLOSUM62)	Scoring matrix
-n (default 2)	Number of threads
-v (default only scores)	Output is more verbose

pxnw -s drosophila.aln

pxrecode

Basic sequence recoding program, currently only switches to RY coding.

Example Folder: Examples/pxrecode_example

Options	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output sequence file

pxrecode -s Nucleotide.fa

pxrevcomp

Basic sequence reverse complementing program

Example Folder: Examples/pxrevcomp_example

Option	Function
-s (required)	Sequence file, stdin otherwise
-o (default stdout)	Output file

pxrevcomp -s Nucleotide.fa

pxrms

Program to remove taxa from a fasta file

Example Folder: Examples/pxrms_examples

Option	Function
-s (required)	Input fasta file, stdin otherwise
-r (required)	List of taxa to be removed, all on separate line
-o (default stdout)	Output file

pxrms -s Nucleotide.fa -r List.txt

pxs2fa

Changes the format of a sequence file that is in fastq, nexus or phylip to fasta format

Example Folder: Examples/pxs2fa_example

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-u (default off)	Force characters to uppercase

pxs2fa -s s2fa_test.phy

pxs2nex

Changes a sequence file that is in phylip, fasta or fastq format to a nexus file.
Example Folder: Examples/pxs2nex_example

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-u (default off)	Force characters to uppercase

pxs2nex -s s2phy_test.phy

pxs2phy

Changes a sequence file that is in nexus, fasta or fastq format to a phylip file.

Example Folder: Examples/pxs2phy_example

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-u (default off)	Force characters to uppercase

pxs2phy -s s2phy_test.nex

pxvcf2fa

Changes the format of a vcf file to fasta format

Example Folder: Examples/pxvcf2fa_example

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-u (default off)	Force characters to uppercase

pxvcf2fa -s vcf2fa_test.vcf

pxsw

Uses smith waterman algorithm to perform a pairwise alignment of all sequences in the file

Example Folder: Examples/pxsw/

Option	Function
-s (required)	Sequence file to be aligned, stdin otherwise
-o (default stdout)	output score/distance file
-a (default off)	Output aligned sequence file
-t (default DNA)	Molecule type (DNA =0, AA = 1)
-m (default DNA=EDNAFULL, AA=BLOSSUM62)	Scoring matrix to be used
-n (default 2)	Number of threads to be used
-v (default off)	Makes the output more verbose, turns off parallel

pxsw -s drosophila.fa -a aln_dros.aln

pxtlate

Program to translate from nucleotide to amino acid using basic translation table

Example Folder: Examples/pxtlate/

Option	Function
-s (required)	File to be translated, stdin otherwise
-o (default stdout)	Output file

pxtlate -s trlateTest.fa

Tree manipulation

pxbp

This program will print out all the bipartition in a given tree

Example Folder: Example/pxbp/

Options	Function
-t (required)	Input treefile, stdin otherwise
-o (default stdout)	Output file

pxbp -t Tree.tre -o bipartitions

pxmrca

This program will give information regarding the mrca, how many tips there are for it etc...

Example Folder: Example/pxmrca/

Options	Function
-t (required)	Input tree file, stdin otherwise
-m (MRCA file)	File of MRCA information is wanted about. Format: MRCA = tip1 tip2
-o (default stdout)	Output file

pxmrca -t mrca_test.tre -m mrca.txt

pxmrcacut

This program will remove parts based off of the MRCA and print out the clades that are given.

Options	Function
-t (required)	Input tree file, stdin otherwise
-m (MRCA file)	File of MRCA wanted to be cut from tree. Format: MRCA = tip1 tip2
-o (default stdout)	Output file

pxmrcacut -t mrca_test.tre -m mrca.txt

pxmrcaname

This program will add names to nodes for the MRCAs.

Options	Function
-t (required)	Input tree file, stdin otherwise
-m (MRCA file)	File with names to be applied to the MRCA's. Format: MRCA = tip1 tip2
-o (default stdout)	Output file

pxmrcaname -t mrca_test.tre -m mrca.txt

pxnni

This program will take a tree and perform a nearest neighbor interchange.

Example Folder: Examples/pxnni

Options	Function
-t (required)	Input tree, stdin otherwise
-o (default stdout)	Output file

pxnni -t nni_tree.tre

pxrmt

Program to remove specified taxa from a tree file

Example Folder: Examples/pxrmt

Option	Function
-t (required)	Tree file to remove taxa, stdin otherwise
-n (required if '-f' is not used)	Comma separated list of taxa to be removed
-f (required if '-n' is not used)	File with taxa to be removed each on a line
-o (default stdout)	Output file

pxrmt -t rmt_test -n s1,s2

pxrr

Program that will reroot a phylogenetic tree, if the outgroup is not found it can also say an error and if multiple outgroups are specified but only certain ones are found it will root based off the ones that are found and can give an error message if requested.

Example Folder: Examples/pxrr

Option	Function
-t (required)	Input tree file, stdin otherwise
-g (required)	Outgroups separated by commas
-o (default stdout)	Output file
-s (default off)	Silences all warnings

pxrr -t rr_test.tre -g s1

pxt2new

Convert a tree file to newick format

Example Folder: Examples/pxt2new

Option	Function
-t (required)	Input tree file, stdin otherwise
-o (default stdout)	Output file

pxt2new -t t2new_test.tre

pxtscale

Tree rescaling by providing either scaling factor or root height (not both); the latter requires an ultrametric tree.

Example Folder: Examples/pxtscale

Option	Function
-t (required)	Input tree file, stdin otherwise
-s (required)	Edge length scaling factor
-r (required)	Height of root (tree must be ultrametric)
-o (default stdout)	Output file

pxtscale -t ultratre -s 10.0

Tree making

pxnj

A basic neighbor joining program that uses canonical branch lengths of substitutions per site. It is designed to create rough trees for very large datasets.

Example Folder: Examples/pxnj_example/

Option	Function
-s (required)	Input sequence file, stdin otherwise
-o (default stdout)	Output file
-n (default = 1)	Number of threads

pxnj -s drosophila.aln

pxupgma

A basic UPGMA tree construction program that will also print out the distance matrix used to create the tree.

Example Folder: Examples/pxupgma_examples

Option	Function
-s (required)	Sequence alignment, stdin otherwise
-o (default stdout)	Output file

pxupgma -s drosophila.aln

Simulation programs

pxbdsim

This program is a birth death simulator that allowing the user to simulate trees under various processes.

Option	Function
-e (integer)	Number of extant taxa (otherwise time)
-t (integer)	Depth of tree (otherwise extant taxa)
-b (default 1)	Birth rate
-d (default 0)	Death rate
-n (default 1)	Number of replicates
-o (default to std out)	Output file
-s (default off)	Show dead taxa
-x (default uses clock)	Random seed number

Example code for pure birth model ending in 100 taxa.
pxbdsim -e 100

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.

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.

Option	Function
-t (required)	Input tree under which sequences will be simulated
-o (default stdout)	outfile
-l (default 1000)	Length of sequence to be simulated
-b (default equal)	Base pair frequency
-g (default no variation)	Shape of gamma distribution
-i (default 0.0)	Proportion of invariable sites
-r (default JC69)	Rate matrix
-w (default all equal)	Amino acid rate matrix
-q (default all equal)	Amino acid frequencies
-c (default off)	Run as amino acid
-n (default 1)	Number of replicates
-x (default clock)	Random number seed
-a (default off)	Print ancestral sequences
-p (default off)	Print tree with nodes labeled
-m (default off)	Use more than one model across the tree
-k (default randomly generated sequence)	Give an ancestral sequence

Example Simulation
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)

pxseqgen -t your_tree.tre -o outfile.txt -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 seqgen_test.tre -o outfile.txt -m .33,.33,.33,.33,.33,.33,2,.3,.3,.2,.5,.4,.2

Run as amino acid

pxseqgen -t your_tree.tre -c

Comparative methods

pxbdfit

This program will fit a birthdeath model to a given ultrametric phylogenetic tree

Option	Function
-t (required)	Input tree file, stdin otherwise
-m (default bd)	Diversification model
-o (default stdout)	Outfile

pxbd -t treefile.tre

pxcontrates

Estimates if a continuous character is evolving under brownian motion or the OU model.

Example Folder: /Examples/pxcontrates_example/

Option	Function
-c (required)	Character file, should be in phylip or fasta and multiple characters should be separated by a tab, space or comma.
-t (required)	Tree file
-a (default is anc)	Analysis type “0” for anc, “1” for a rates test
-o (default stdout)	outfile

pxcontrates -c contrates_file.txt -t ultra_100.tre -a 1

pxstorec

Performs ancestral state reconstruction and stochastic mapping of categorical variables

Option	Function
-d (required)	Data file
-w	Data is in wide format
-z	Data is in probability format
-t (required)	Tree file
-c	Configuration file
-o	Output for ancestral calculation
-n	Output for stochastic mapping duration
-a	output file for stochastic mapping number any
-m	Output for stochastic mapping duration
-p	Comma separated times
-l	Log file

Bayesian

pxlog

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:

Option	Function
-p	input parameter log file(s)
-t	input tree log file(s)
-o (default stdout)	output file
-b	number of samples to exclude at the beginning of a file
-n	interval of resampling
-r	number of random samples (without replacement) not yet implemented!
-i	calculate log file attributes and exit
-x	random number seed, clock otherwise
-v	make the output more verbose