

PLEX Quick Start Guide

Phylogenetics, Likelihood, Evolution, and Complexity

by A.P. Jason de Koning, Wanjun Gu, and David D. Pollock

Quick Start Guide by Corey J. Cox, Todd A. Castoe, Jason de Koning, and David D. Pollock

This guide is intended to walk a new *PLEX* user through downloading the software, running the sample datasets, and interpreting the results. Current versions and documentation can be found at <http://www.EvolutionaryGenomics.com> (mirror at <http://jasondk.org>).

Downloading the Software

Downloads available at the above URLs are the most current versions of the software and the quick start guide (this document). The current version as of August 27, 2012 is *PLEX_v0.95_distribution*. Older versions will be archived at this site as well. Mac and Linux versions have been tested and compiled, but with luck it should work on other OSs too.

Unpacking the Software

Move the archive to a convenient location for expansion. **On Mac**, run the *Archive Utility* program to expand the archive by double-clicking. *Archive Utility* is standard on OS X. **On Linux**, Run the *unzip* utility to expand the archive. *unzip* is standard on most versions of Linux.

Main Directory Structure and File Descriptions

The archive will expand to the folder *PLEX_v0.95_distribution*. On Linux there may also be a folder named *__MACOSX*, which can be ignored or deleted.

The *PLEX_v0.95_distribution* directory contains the following files and folders:

README.txt – This file provides minimal step-by-step instructions on how to compile and use *PLEX*, as well as how to use the associated analysis scripts.

examples – This directory contains example controlfiles and associated data files to run example analyses. For a detailed description of these examples please see the Examples section below.

scripts – This directory contains helpful post-processing scripts for examining the output of the analysis and validating the results. For a detailed description of these examples please see the Post-Processing Scripts section below.

src – This directory contains source code for *PLEX*. This is where the program is compiled.

Compiling PLEX

To compile *PLEX*, open a terminal window and navigate to your *PLEX_v0.95_distribution* directory. From here, enter the *src* directory and issue the *make* command to compile:

```
cd src/  
make
```

This may take a while to complete, depending on the system that you are using.

NOTE: By default *PLEX* compiles using the **-ffast-math** compiler flag turned on, as do some other phylogenetics packages. This flag can, in limited circumstances, lead to numerical instability (although we have never had problems). If you prefer, you may turn this flag off in the Makefile, by commenting out the default CFLAGS line, and uncommenting one without **-ffast-math**. The Makefile can be edited with any text editor, but edits must be saved in plaintext format. A line is commented out if it begins with an '#’.

Running PLEX

To run *PLEX*, we recommend using a working directory to contain your data, control files, and the plethora of output files that *PLEX* will produce (e.g., we include `./working` in the *PLEX* distribution for this purpose). To run examples and post-processing scripts, change to the working directory and copy the examples and scripts from the appropriate directories.

```
cd ../working
cp ../examples/* .
cp ../scripts/* .
```

Next, choose an example controlfile to run. *PLEX* expects the controlfile to be named 'controlfile'. You can therefore copy or rename the control file of your choice, or link the name *controlfile* to your chosen example file before running *PLEX*. For example,

```
ln -sf controlfile_5taxon_tester controlfile
./PLEX
```

To run other examples, use different control files in place of *controlfile_5taxon_tester*. For a description of the files produced by running *PLEX* please see the *Output Files* section below.

Post-Processing Results

PLEX comes with a few convenient scripts to examine and validate the results. These scripts use the *R* statistical programming language and *Perl*. If you do not have *R* installed on your system it can be downloaded free from <http://www.r-project.org/>. If you do not have *Perl* installed on your system it can be found at <http://www.Perl.org/>. All of the *R* scripts have *Perl* wrappers, meaning they do not have to be run directly.

The *Perl* scripts can be run directly from the working directory, after having been copied there as described above. For further detail please see the *Post-Processing Scripts* section in the *Detailed Descriptions* below.

To examine log-likelihood traces for the incomplete-data log likelihood function (the pruning-based standard likelihood), type

```
perl plotInLikelihood.pl 1 likelihoodfile
```

It should be noted that *PLEX* avoids calculating full likelihoods as much as possible, since they are expensive to compute and are not required for most update mechanisms. The likelihood appears to stay at the same value over many generations in these plots because it is only updated infrequently. To change the frequency of the incomplete-data log-likelihood calculation, see the below section on controlfile parameters.

The complete data log-likelihood is updated by the program more frequently and can be viewed using

```
perl plotInLikelihood.pl 1 likelihoodfile
```

Note: this should never be used for model comparison. If you don't know what this is, you probably don't need to look at it!

To examine a posterior rate matrix heat map summary for amino acid models, type

```
perl posteriorSummary.pl 1 matrixoutfile
```

In these examples, *likelihoodfile* and *matrixoutfile* refer to files generated by *PLEX*. On Mac OSX this will create a *pdf* file and automatically open the file for you in the program *Preview*. On Linux the script will generate an error when attempting to open *Preview*; ignore the error and open the *pdf* file manually.

Detailed Descriptions

This section provides a full listing of the files mentioned, along with additional information.

Example Files

These files are provided as examples for how you may want to run *PLEX*. These files include datasets to demonstrate different run conditions.

The first example is a small 5-taxon example to test that everything is set up properly. It includes a control file, a sequence (fasta) file, and a tree file.

```
controlfile_5taxon_tester    --Example control file for 5-taxon example
5taxon.fasta                 --fasta file data for 5-taxon example
5taxon.tree                  --tree data for 5-taxon example
```

The next example demonstrates different types of *PLEX* analyses (nucleotide general time reversible, nucleotide non-reversible, protein general time reversible, and protein non-reversible) with a 224-taxon dataset of cytochrome-*b* (cytb) sequences. There are four control files, two sequence files and two tree files.

```
controlfile_224cytb_gtr
controlfile_224cytb_prot_generalNonRev
controlfile_224cytb_nonRevGeneral
Mam_CYTB_v4.reduced.noGaps.noAmbig.fasta
Mam_CYTB_v4.reduced.noGaps.noAmbigs.fasta.prot
CYTB_nuc.tree
Mam_CYTB_v4.reduced.paml_mtMam.tree
```

When running a general non-reversible model in *PLEX* without rate variation, Conjugate Gibbs sampling will be used for both the rate matrix and branch-lengths, yielding the fastest possible analysis. Use of a reversible model or rate variation will cause slice sampling to be used, which is not as fast as Conjugate Gibbs, but is still much nicer than using Metropolis Hastings. By default, branch-length sampling is always performed using Conjugate Gibbs methods.

Post-Processing Scripts

R and *Perl* scripts used in post-run processing to visualize data. As mentioned above, the *R* scripts are called directly by the *Perl* scripts.

```
plotHeatmaps.pl
```

posteriorSummary.pl
plotLnLikelihood.pl
posteriorVariance.pl
plotLikelihood.pl

Output files

These files may be produced by PLEX as output. Below are lists of the columns, in order.

suboutfile -

A customizable listing of augmented substitutions across the data-set. The substitutions are marked based on the standard one-letter nucleotide or amino acid codes.

likelihoodfile -

Count - MCMC generation

Updates - Approximate number of updated parameters

LnLikelihood - The complete-data log-likelihood (i.e., conditioned on the current set of augmented data) - this likelihood should rarely be used by the user, but is included for reference

IntegratedLnLikelihood The incomplete-data log-likelihood (i.e., integrated over ancestral and transient states) - this likelihood should be used for most applications

matrixoutfile -

Gen - MCMC generation

Mu - the uniformization constant; see de Koning et al. (2010) for description

rateSum - scaled sum of the rates matrix (used for normalization)

treeDepth - the sum of all branchlengths

Remaining Headers - amino acid substitution using one-letter codes

siteRates -

when among-site rate variation models are used this contains the rate assignments for each site over time

siteRateParameters -

when among-site rate variation models are use, this contains the parameters describing the rate distribution

stateFrequencies -

when using reversible models, this contains the state frequency parameters

stationaryFrequencies -

when using a non-reversible model, this contains stationary frequencies implied by the rate matrix over time

timetakenfile -

contains the analysis time of the most recent run

treeoutfile -

when branch-lengths are sampled, this contains a Newick-formatted output of the current branchlengths

In addition, we have also reserved the following output files for future use.

countoutfile

RARparameters

mixtureDependencies

posteriorPredictive

seqoutfile

siteAssignments

Control Files

The file *controlfile* should be a file or a symbolic link to a file containing the appropriate parameter settings for PLEX to run your desired analysis. Example control files are available in the "*examples*" section. An example of a control file is appended with brief descriptions of the variables and their settings. The basic format is that comments are bracketed by hash marks, while parameter setting lines begin with a variable name followed by the variable setting. Note that unclosed comments may hang the program. The file ends with the word "end". The main controls are for input, output, and how to run the program (MCMC, assumptions, model, proposal updates). This example has been highlighted for readability but should work if saved as plain text.

PLEX control file

Input data

```
treefile                    5taxon.tree
    # file containing tree (or trees) to be input #
seqfile                    5taxon.fasta
    # input sequence file #
```

MCMC control parameters

```
generations                50000
    # num generations to run MCMC #
outputFrequency            100
    # how often to output chain status, etc. (smaller numbers generate more file output,
    slower) #
outputincompletelogl      1000
    # output the incomplete-data logL every ? generations #
ancstateupdatefreq        0.01
    # freq of ancestral/transient state updates #
forcefullupdater           0
    # 1 = force ancestral/transient state update to always use pruning-based sampler,
    more expensive; 0 = use local fast updater from Krishnan et al. (2004) #
updatebls                  1
    # 1 = update branchlengths; 0 = no; Conjugate Gibbs by default #
```

Assumptions

```
gapsasmissing              0
    # 0 = exclude gapped columns; 1 = treat as missing, impute during MCMC
maxbl                      0.08
    # maximum branch segment length when sampling transient states; shorter is more
    precise, longer is faster #
# liktemp                  1.0      #
    # for thermodynamic integration; commented out here #
```

Model customization

```
ratemodel                  0
    # choice of input rate model to use, if needed #
    # 0 = General model; 1 = Poisson; 2 = JTT; 3 = mtMam #
reversible                  0
    # Use of preset models that require reversibility will override this setting #
    # 0 = non-reversible; 1 = reversible #
freefreqs                  0
    # state frequencies may be estimated during run, or use empirical freqs #
    # 1 = estimate state frequencies; 0 = fixed #
empiricalFreqs             0
```

```

# 1 = fix state frequencies as observed in alignment; 0 = estimate or use stationary
freqs (depends on model) #

ASRV                                0
# ASRV stands for "among-site rate variation" #
# 1 = use gamma rate variation; 0 = no rate variation #
ASRV_numCat                        1
# number of discrete rate categories for ASRV #
ASRV_gammaShape                    1.0
# starting value for the gamma shape parameter; see literature for explanation #
ASRV_update                        0
# 1 = use Metropolis-Hastings updater for the shape parameter; 0 = fixed #

# Proposal and updater options #
sliceNumGens                       3
# For slice sampler, number of draws per sample; higher numbers = slower but less
correlation between samples #
updatewindow                       0.005
# For MH proposals; width of uniform proposal #

# What output files do you want to print? #
outtreeflag                        0
outseqsflag                        1
outmatsflag                        1
outcountflag                       1
outlikesflag                       1

# Special settings for substitution history output #
outputSubs                         0
# 1 = output it; 0 = no #
outputSubFrequency                 10
# frequency of output (print substitutions every ? generations; separate from main
output frequency) #
outputSubType                      4
# Output formats: 0=subs on tree (based on branch endpoints); 1=subs in table (based
on branch endpoints); 2=subs in table (branch endpoints + transient points); etc. #

# Other output options #
outputSiteRates                    0
# 1 = output site rate assignments; 0 = no, don't bother #

# Output file names #
treeoutfile                       treeoutfile
seqoutfile                        seqoutfile
countoutfile                      countoutfile
likelihoodoutfile                 likelihoodfile
matrixoutfile                     matrixoutfile

end

```