# PLEX Quick Start Guide

## PLEX: 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 and David D. Pollock*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This guide is intended to walk a new *PLEX* user from downloading the software through running the sample datasets and interpreting the results. Current versions and documentation can be found at www.EvolutionaryGenomics.com/ProgramsData/PLEX/

### Downloading the Software

The software available online at [www.EvolutionaryGenomics.com](http://www.EvolutionaryGenomics.com) (mirrored at [jasondk.org](http://jasondk.org)) is the most current version of the manual and quick start guide. The current version as of July 27, 2012 is the *PLEX\_v0.94\_distribution*. Older versions should be archived at this site as well. Mac and unix versions have been tested and compiled, but with luck it should work on other OSs.

### Unpacking the Software

Move the archive to a convenient location for expansion. ***On Mac***, run the *Archive Utility* program to expand the archive. *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\_v0.94\_distribution*. On Linux, there may also be a folder named \_\_MACOSX, which can be ignored or deleted.

The *PLEX\_v0.94*\_*distribution* directory contains the following files and folders:

*README.txt* – This file is a brief step-by-step description 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 will be compiled.

### Compiling PLEX

To compile *PLEX*, open a terminal window and navigate to your *PLEX\_v0.93\_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.

### Running PLEX

Current best practice for running *PLEX* is to make a working directory to store all of your data and control files for the program. To run examples and post-processing scripts, copy the needed files to the working directory by starting in the source directory and issue the following commands:

*mkdir* ../working

cp ../*PLEX* ../working

cp ../examples/\* ../working/

cp ../scripts/\* ../working/

cd ../working

Once in the working directory you will need to decide which example to run. You then need to use the correct control file. You can copy and rename the control file of your choice, or link the name *controlfile* to your chosen example control file before you run *PLEX* by typing ./*PLEX* from the working directory. For example,

ln -sf *controlfile\_224cytb\_gtr* *controlfile*

./*PLEX*

To run other examples, use different control files in place of *controlfile\_224cytb\_gtr* .

### Post-Processing Results

*PLEX* comes with a few convenient scripts to examine and validate the results. These scripts use the *R Project for Statistical Computing* and the *Perl* scripting language. If you do not have *R* installed on your system it can be downloaded free from [www.r-project.org/](http://www.r-project.org/). If you do not have *Perl* installed on your system it can be found at [www.Perl.org/](http://www.perl.org/). All the *R* scripts have *Perl* wrappers, meaning they do not have to be run directly.

To run the scripts, first copy the script files into the current directory if they're not there already. Then run the appropriate *Perl* script as described below. These scripts need to be run with parameters on the command line, and if you don't have nice paths set up, you may need to type "./*Perl*" instead of just "*Perl*". There are examples of post-processing that can be performed with the analysis scripts provided. For further detail please see the *Post-Processing Scripts* section in the *Detailed Descriptions* below.

To examine runs from either complete or incomplete-data log likelihoods over time, type

*Perl* *plotLikelihood.pl* 1 likelihoodfile

To examine a posterior rate matrix summary, type

*Perl* *posteriorSummary.pl* 1 matrixoutfile

Here *likelihoodfile* and *matrixoutfile* are output from running *PLEX* as in our examples above. On Mac OSX this will create a *pdf* file and automatically open the file for you in the program *Preview*. On Linux the script command that opens the file will generate an error; 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 analysis (nucleotide general time reversible, protein general time reversible, nucleotide non-reversible, and protein non-reversible model analyses) 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

controlfile\_224cytb\_prot\_RAR

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

**Post-Processing Scripts** – *R*, *Perl* and *Bash* scripts used in post-run processing to visualize data. The *R* scripts can be run directly, but we have included the *Perl* and *Bash* scripts for convenience as wrappers to run the R scripts with appropriate parameters.

calcESS\_rateMatrix.sh plotLikelihood.pl

effectiveSize\_ratematrix.R plotLikelihoodHist.R

plotHeatmap\_sorted.R plotLikelihoods.R

plotHeatmap\_unsorted.R plotLikelihoods2\_incomplete.R

plotHeatmap\_var\_unsorted.R posteriorSummary.pl

plotHeatmaps.pl posteriorVariance.pl

plotInLikelihood.pl

**Output files** – These files may be produced by PLEX as output.

treeoutfile

seqoutfile

countoutfile

likelihoodfile

matrixoutfile

**Control Files** – The file *controlfile* should be a file or symlink to a file containing the appropriate variable 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 lines to set parameters begin with a variable name followed by the variable setting. The file ends with the word "end". The main controls are for input, output, and how to run the program (MCMC, assumptions, model, proposal updater). This example has been highlighted for readability but should work if saved as plain text.

## # PLEX control file #

# Note: comments must have an open hash and close hash as shown in this line. #

### # 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 ouput 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 = no #

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 for B1 or B1u ; 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

# models can followed detailed reversibility, or not *#*

# 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

# How to get empirical frequencies *#*

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

ASRV 0

# ASRV stands for "across sites 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

# gamma shape parameter for initiation or fixed shape runs; 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 adjacent 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 (actually, gens between; 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**