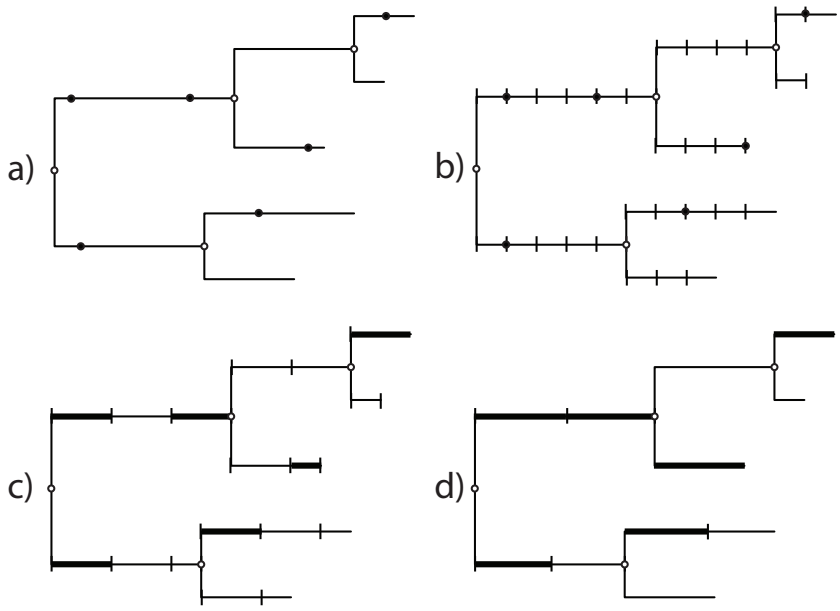
PLEX EXTENDED DOCUMENTATION

PHYLOGENETICS, LIKELIHOOD, EVOLUTION, AND COMPLEXITY

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

Beta version 0.95. Updated September 1, 2012.



For the most up-to-date version, please visit:

http://www.EvolutionaryGenomics.com/ProgramsData/PLEX/

or

http://jasondk.org/Software/PLEX/

Requests, questions, or bug reports should be directed to:

Jason de Koning (jason.de.koning@gmail.com)

or

David Pollock (David.Pollock@UCDenver.edu)

1. INTRODUCTION

PLEX is a multi-purpose Bayesian software package that performs Markov Chain Monte Carlo (MCMC) analysis of large phylogenetic and phylogenomic datasets. *PLEX* is a platform for evolutionary analysis, and is under active development. It will therefore grow in its capabilities over time. At present it can perform a variety of the expected, standard evolutionary analyses using either nucleotide or amino-acid substitution models. In addition, it implements rapid variations of standard approaches, allowing, for example, the inference of general reversible or non-reversible amino-acid substitution models in only minutes from large datasets. As described below, additional non-standard models are currently available.

This "further documentation" file is intended to eventually be part of a full manual. It is a work in progress. Over time, as the program grows, tutorials and relevant material will be added. Prior to creation of an online Q & A forum to field general requests, please contact David or Jason for questions.

This extended documentation is intended to augment the README.txt, Quickstart, and Background documentation files. Please start with these files to obtain basic information and understanding on running the *PLEX* program. The extended information in this document will include discussion of output file format and the meaning of column information, as well as further information about the meaning and use of parameters in the control file.

2 MODEL AND SAMPLING CONTROL PARAMETERS

The model parameters are listed followed by curly brackets to indicate expected range of values, if any. Further explanation for each parameter ensues.

2.1 Branch lengths

updatebls {0|1}

This parameter controls whether or not the branch lengths will be updated or remain fixed during the run. 1 means update, 0 means not. Updates are done using conjugate Gibbs sampling by default. Other methods are in the code but not currently available.

maxbl ***{0.00001 – 10.0}***

This controls the maximum allowed branch or branch segment length for the B1 or B1u methods (see de Koning et al). 0.08 is a pretty good value. Shorter is more precise, but longer is faster.

hwang {0|1}

Flag to use the Hwang and Green method. This is basically deprecated but could be resurrected.

numbls {1}

This is for B1xM and related methods, and is basically deprecated.

bls {0.02,0.01,0.005 }

Distribution for B1xM. Deprecated. Most of the fun stuff we are doing now is specific to B1u.

2.2 Substitution rates matrix

ratemodel {0|1|2|3}

This controls the choice of input model. This can be (0) a general model (time reversible or not, depending on "reversible" parameter below), (1) a Poisson model (Felsenstein), and for proteins (2) the JTT (Jones, Taylor, Thornton; nuclear) or (3) mtMam (mammalian mitochondrial, from literature). Although please note that nuclear or mitochondrial models can be generated from modern data quite rapidly from any available dataset.

freefreqs {0|1}

Should the state (e.g., nucleotide, amino acid, etc.) be allowed to be free (estimated during the run), or be fixed? Note that for non-reversible models, the stationary distribution implied by the current rates matrix is always used (if solvable).

empiricalFreqs {0|1}

If using fixed frequencies (see above), there is the choice to use stationary frequencies (option 0) or to use the frequencies directly from the alignment (option 1).

reversible {0|1}

The model can be reversible (option 1) or not (option 0). For reversibility, the estimated equilibrium state frequencies are used as part of the model (along with "rate" parameters). For the more general non-reversible model, the rate parameters represent the probability of substituting from one state to another, and the stationary frequencies are used at the root of the tree.

fixedrates {0|1}

Should the rates be free? This is deprecated and it is pretty much all controlled by ratemodel.

2.3 Site heterogeneity in rates

ASRV {0|1}

ASRV = among site rate variation. The current choices are to have no variation in rates across sites (option 0) or to use a discrete gamma distribution for rate variation (option 1). Rate variation is implemented slightly differently than in most other programs (see de

Koning et al., 2010)

ASRV_numCat {2...}

This controls the number of discrete categories for ASRV. This is usually in the range of 4-8, but can be 2-20. Feel free to test higher numbers at your peril.

ASRV_gammaShape {close to 0 ... very large}

This is the gamma shape parameter (real number) used for initiation of runs with gamma rate variation, or for the fixed parameter for fixed shape runs (see below). A very small value will result in a near uniform distribution (less variation). It should be noted that PLEX implements rate variation in a way that is somewhat different from other programs (see de Koning et al., 2010). Likelihoods from PLEX should therefore ideally not be directly compared to other programs when using ASRV. Also note that PLEX uses data augmentation to integrate over rate variation across sites.

ASRV_update {0|1}

If 0, the gamma shape parameter is fixed at ASRV_gammaShape, and if 1 Metropolis-Hastings will be used to update the shape parameter. We will switch this to a slice sampler at some point.

outputSiteRates {0|1}

This true/false parameter controls whether or not site rate assignments are output. Under a gamma model it is which of the k categories is assigned at a given moment at each site.

2.4 Site heterogeneity in substitution process

Manuscript in preparation.

2.5 Co-evolutionary analysis and simulation

Manuscript in preparation.

2.6 Probabilistic Ancestral State Reconstruction

outseqsflag {0|1}

Flag to output ancestral state reconstruction, or not. Ancestral reconstructions are performed using the posterior ensemble of augmented ancestral states over the chain.

2.7 Statistics of Substitution mappings

outputSubs {0|1}

Flag to output substitution mappings, or not.

outputSubFrequency {1...}

Frequency of output for ancestral substitution mapping output. The number provided is

the generations in between output samples. This is separate from the main output frequency.

outputSubType {0|1|2}

Format for ancestral substitutions output. Option 0 outputs substitutions on the tree based on branch endpoints. Option 1 outputs substitutions in a table (based on branch endpoints), and output 2 puts output substitutions in a table (with branch endpoints and transient points).

2.8 Model Evaluation and Hypothesis Testing

generations {1000...1000000000}

This controls the number of generations to do the run. A rather important number. 50,000 is a good number to start with. Don't make it too big until you know how long it will take, and whether more generations are really needed. Remember, number of generations is not really comparable across different programs.

2.9 A bunch of Old Options that are Currently Ignored or Deprecated

These might be useful to know about only if you are really going through the code.

burnin	***0***
seqfreq	***10***
matfreq	***10***
probfreq	***10***
countfreq	***10***
testupdate	***0***
updatewindow	***0.005***
updatepercentage	***40***
totalwindowsize	***1000***
maxwindowsize	***40***
brlenupdatefreq	***{near zero to 1.0} 0.02***
model	***0***
site	***1***
probmethod	***{6}***
task	***{0}***

2.10 Data Input and Handling

gapsAsMissing {0|1}

If 0, columns with gaps are excluded. If 1, they are treated as missing data and imputed during the MCMC run.

treefile {filename}

Tree file for input, in Newick format, e.g., "5taxon.tree". Starting branch-lengths are expected by PLEX.

seqfile {filename}

Sequence file for input, in Fasta format, and pre-aligned, e.g., "5taxon.fasta". Gaps are dash.

2.11 Controlling the MCMC

This includes parameters to control the ancestral state updater, and how often (in run generations) to output the incomplete data log likelihood (the standard, pruning-based likelihood function).

ancstateupdatefreq {0...1.0}

This controls how often augmented ancestral states should be updated in the MCMC run. 0.01 is a good number, but can vary depending on application. This is the probability that any given update will be an ancestral/transient state update sweep. If mixing seems poor, try increasing this. For faster runs, try decreasing it.

forcefullupdater {0|1}

If this flag is 1, the ancestral/transient state updater is always forced to use a pruning-based sampler, which is basically the same as a marginal ancestral state updater. This updater mixes better but is more expensive (see Nielsen, 2002). The alternative is our local ancestral state updater that looks at the three surrounding nodes and propagates through the tree (see Krishnan et al., 2004).

liktemp {near zero ... 1.0}

This is used for thermodynamic integration, and controls the likelihood "temperature". Will be detailed more in publication *in prep*. A value of 1.0 means standard likelihood. You probably shouldn't use this unless you know what you're doing.

outputincompletelogl {1-100,000,000,000,000}

This controls how often (in number of generations) the incomplete-date log likelihood should be evaluated and output. This is expensive and slow, so we often do it only every 1000 generations or more. Note, this is not strictly required and is only done so the progress of the run can be examined via the likelihood trace (use plotInLikelihood script; see Quick Start).

outputFrequency {1...10,000}

This controls how often, in numbers of generation, the chain status should be output. Smaller numbers mean output is more often, which generates larger files and can be much slower. 100 is a good number.

2.12 Basic Run Options

seed {integer}

This is a seed for the random number generator. It should be an integer, smaller than the

maximum value of size int. If you don't set this it is initialized based on time.

debug ***0***

Flag to control debugging output, or not.

loudness ***1***

As Ziheng says, this is useful if you are lonely. How lonely are you, anyway?

treeoutfile ***{filename}***

This parameter and the next 6 parameters are output file names. They are optional, and if left out the parameter name will be used as the filename (e.g., treeoutfile = "treeoutfile").

suboutfile ***{filename}***

seqoutfile ***{filename}***

countoutfile ***{filename}***

likelihoodoutfile ***{filename}***

matrixoutfile ***{filename}***

outtreeflag ***{0|1}***

This flag and the following 4 flags control whether certain output files will be printed.

outseqsflag ***{0|1}***

outmatsflag ***{0|1}***

outcountflag ***{0|1}***

outlikesflag ***{0|1}***

end

When this word is read at the beginning of a line, the rest of the parameter file is ignored.

3. SHORT DESCRIPTIONS OF EACH FILE IN THE EXAMPLES DIRECTORY.

This can be found in the quick start guide.

4. FILES IN THE SCRIPTS DIRECTORY

The R scripts are all called by the Perl scripts, which are basically just wrappers.

4.1 Post-Processing Scripts

posteriorSummary.pl

plotInLikelihood.pl

posteriorVariance.pl

plotLikelihood.pl

5. POST-RUN ANALYSIS

One of the advantages of using *PLEX* is that the likelihood function need not be evaluated very often (if ever) in order to perform MCMC. For the purposes of diagnosing chain burnin and overall mixing, however, *PLEX* supports calculation of the full (incomplete-data) log likelihood every **`outputincompletelogl`** generations. *PLEX* will also recalculate the likelihood whenever a complete missing-data update is performed using pruning, since calculation of the likelihood is only a trivial sum at the root of the tree from the conditional site likelihoods.

Likelihood traces are saved in **`likelihoodoutfile`** (called 'likelihoodfile' by default) as part of the periodic chain output. To visualize this output, you can use a program such as *Tracer*, or the included Perl script (which also assumes R is installed):

```
perl plotInLikelihood.pl <burnin> <likelihoodfile>
```

Note here that ***`burnin`*** is the number of chain samples at the beginning of the chain that should be excluded from the plot. To see the full chain, a value of 1 can be used. If the script is run on a Mac, the PDF of the likelihood trace will automatically open in a Preview window.

Further explanation of post-processing scripts is intended for planned tutorials. Also, we are happy to include useful scripts and recommendations from users.

6. PLANNED TUTORIALS

6.1 General non-reversible model estimation (Conjugate Gibbs)

6.2 General reversible model estimation + rate variation (Slice sampling; MH)

6.3 Site Specific Mixture Model Estimation

6.4 Ancestral State Reconstruction

6.5 Inferring Convergent Evolution

6.6 Bayes Factor Estimation

7. REFERENCES

Not yet included.