Inference-Based Posterior Prediction Tutorial (with PuMA v0.906 and AMP v0.99)

Jeremy M. Brown
Dept. of Biological Sciences
Louisiana State University
www.phyleauxgenetics.org
jembrown@lsu.edu

Bodega Bay Applied Phylogenetics Workshop
3.9.14

Disclaimer

Both PuMA and AMP are under active development to facilitate larger-scale analyses. Right now, there are parts that could be more user friendly and don't natively facilitate the use of high-performance computing resources. This will change soon! Keep checking back for upcoming releases.

Also, we are hoping to implement all of these methods into the forthcoming RevBayes software. However, this is a long-term goal.

Installation and Dependencies

PuMA

PuMA and associated files are currently only available for download as a disk image (.dmg) for Macs. If you are using a PC, you'll need to find a partner to work through these tutorials.

From the PuMA downloads page (http://code.google.com/p/phylo-puma/downloads/list), download all the available files (an R script, a disk image, and a manual). After download, mount the disk image and move all the files that it contains into whatever folder you have established to run

this tutorial. PuMA comes with a compiled version of seq-gen that it will need to use to simulate datasets. To see if the already compiled version of seq-gen works for you, open up Terminal, change directories to your tutorial directory and try running seq-gen directly:

```
$ ./seq-gen
```

If seq-gen doesn't execute properly, delete the executable:

```
$ rm seq-gen
```

and then change directories into the "seq-gen source" folder:

```
$ cd cd seq-gen\ source/
```

You will now need to re-compile seq-gen. Look at the contents of this source code folder. If there are any files ending with .o, you need to delete these. To do so, type:

```
$ rm *.o
```

Once all the .o files have been removed, simply type make

```
$ make
```

You should now have a functioning version of seq-gen. To make sure it runs properly, try it:

```
$ ./seq-gen
```

If it runs properly, move the executable to the parent directory:

```
$ mv ./seq-gen ..
```

Now, just change your working directory back to the parent directory:

```
$ cd ..
```

You can check to make sure PuMA runs properly by just double-clicking on either the “PuMAv0.906” native Mac application or the “PuMAv0.906.jar” file. You could also start the .jar program from the command line:

```
$ java –jar PuMAv0.906.jar
```

If a PuMA GUI window pops up, you’re all set.

AMP

AMP can be downloaded from https://code.google.com/p/phylo-amp/downloads/list. It is written in Python and depends on the DendroPy library, so you will need to have both some version of Python 2 installed (http://www.python.org/getit/) as well as DendroPy (http://pythonhosted.org/DendroPy/). Note that DendroPy does *not* work with Python 3.

Data Pre-Processing

Missing Data or No Missing Data?

If your data set includes missing data and/or gaps, you have two choices for running your analysis. **As long as you add gaps to simulated datasets after you run PuMA**, you can still use inferential test statistics as calculated in AMP. However, if you want to interpret the multinomial-based test in PuMA, you should remove gappy sites from your dataset prior to analysis in MrBayes. If you want to both use the multinomial test statistic and inferential test statistics, you can either only use a dataset free of missing data or you can do the analysis twice (inferential test statistics calculated on datasets with missing data and multinomial test statistics calculated on datasets *without* missing data). The easiest way to remove sites with missing or ambiguous characters is to use PAUP*. Simply load your dataset, use the *exclude missambig* command to remove all sites with gaps or ambiguity codes, then export the filtered dataset.

Run MrBayes

Once you've decided whether to use a dataset with or without missing data (or both). Run the preferred data through MrBayes under any standard or partitioned nucleotide model you want. Right now, PuMA can't handle 'exotic' models like the covarion, or models built around other state spaces like the doublet, codon, or amino acid. Store your MrBayes commands for each analysis in a dedicated file that only has the MrBayes block. If you use a partitioned model, PuMA will need to use this file to figure out how the data are partitioned. Please see the .bayesblock file in the PuMA example folder as a template.

Subsample MrBayes Output (If Necessary for Inferential Test Statistics)

Depending on how long your focal (e.g., empirical) data set took to run in MrBayes, and how often you've decided to sample the MCMC during your MrBayes run, you might need to subsample your MrBayes output so that you're not trying to analyze several hundred datasets or more. You should do at least 100 posterior predictive replicates, but you might not want to go much higher than that if your runs are going to take a long time. You can use the Python script included with this tutorial to subsample .p and .t files from your MrBayes analysis. You'll need to open up the Python script in a simple text editor and set a few options at the top (don't worry, you don't need to know anything about Python).

For the sake of time during this tutorial, make sure you've done some subsampling to keep things reasonable. Subsample the analysis of the complete data set (no missing) to 100 samples and the original data set with missing data to 25 samples.

Note that if you're going to use the inferential test statistics, you'll want to keep your original output files around so that you are comparing comparable output between empirical and simulated analyses. To make later scripts work, put these in a folder called "originalOutput".

Running PuMA (GUI)

To run PuMA, simply click on either the native Mac OS X application or the .jar file. If you prefer, you can also start it from the command line with the command:

```
$ java –jar PuMAv0.906.jar
```

For starters, let's work with an analysis of primates.nex example file that's distributed with MrBayes. If you don't have access to PAUP*, I've already created a version of this dataset that doesn't have missing data (there are only about 10 positions with missing data). We'll set a Jukes-Cantor model and analyze this locus in MrBayes. Do one analysis on the original data set and one on the data set with no missing data. I've included output files from these analyses with the tutorial files, if you want to compare them to yours. To speed things up, just use 1 chain and only run the analysis for 20,000 generations. I set up my analyses to record samples from the chain every 100 generations and I used 4 independent runs.

Once you have these output files from MrBayes, let's analyze each of them in PuMA. Note that the results from posterior predictive tests with the multinomial likelihood are best interpreted from the dataset with no missing data. Nonetheless, the analysis with missing data can be used to generate simulated data for analysis with AMP, as long as the patterns of missing data are superimposed on the simulated data after they are generated.

Input/Output Tab

In the PuMA GUI, click the button next to "Data File:" and select your alignment. Remember, if you want to interpret the corresponding multinomial posterior predictive analysis, only select a data set free of missing data. If you're using PuMA to simulate data that will be run through AMP, you can select any data set as long as you replicate the patterns of missing data after simulation.

Click the button next to "Directory for .t/.p Files:" and select one of your .p or .t files. It doesn't matter which one. PuMA is just using this to tell it which folder these files are in. Just make sure all of them are in that same

folder. NOTE: PuMA will use *any* .p and .t files that it finds in that folder. If you've created subsampled .p and .t files, move the original .p and .t files into a new subfolder.

In the box next to "Test Statistic Output File:" you should enter the name of whatever file you want to use to store your test statistic values and resulting *p*-value for the multinomial posterior predictive test. Even if you're using a data set with missing data and don't plan on using this file, just enter something here to remind you to delete the file later.

Bayesian Analysis Tab

PuMA currently can accept samples generated by either MrBayes or BayesPhylogenies. In this case, select the radio button for "MrBayes".

For this tutorial, we won't deal with partitioned analyses (although PuMA can handle partitioned models). Since we've just applied a homogeneous Jukes-Cantor model, select the "Single Partition" radio button.

If you do select "Multiple Partitions" for other analyses in the future, you'll need to provide PuMA with a file containing the MrBayes block (.bayesblock) file that you used to set up your partitioning scheme. Please follow the .bayesblock file provided in the PuMA example folder as closely as possible. While PuMA has some robustness built in, it's still pretty finicky about the format of that file. See the PuMA manual for more details.

Burnin Tab

If you've been able to coax a copy of MrConverge from Alan Lemmon and have used it to run your Bayesian analyses, then you can specify here that you'd like PuMA to pull your burnin from a MrConverge log file. If you haven't used MrConverge, you'll need to manually enter a burn-in value. If you've already subsampled your .p and .t files (as we have in this tutorial), simply enter a manual burn-in of 0, since burn-in was already removed during the sub-sampling.

MrConverge Tab

If you want to run your Bayesian analysis with MrConverge from within PuMA, you can tell it to do so here.

Running PuMA

Once you've set up all the options in the four tabs (go back through them and make sure you really have), you can click the Submit button. After you do, just sit back and watch the "Status:" window until it says Run Complete. If you're doing single-partition analyses for a small number of samples (< 200), this should run relatively quickly.

Running PuMA (Batch)

If you're going to be doing PuMA analyses for more than a handful of datasets, setting the options manually through the GUI is going to become very cumbersome. To alleviate this, PuMA input can be specified in a file that can be provided to PuMA if started on the command line. To try this, copy the "example.in" file from PuMA's example folder to one of the folders where you're running a primates analysis. Delete any previous PuMA output (the output file containing the multinomial test statistics and the "SeqOutfiles" and "TREEOutfiles" folders). Rename this input file to something meaningful for your analysis (e.g., primates.in). Open the .in file in a text editor (I prefer TextWrangler) and alter the options as needed. Leave all lines in the file, even if your analysis won't use certain options. Many of the options in the input file will be self-explanatory, but check the PuMA manual for further explanation, if needed. Do NOT alter the names of the options to the left of the equals signs. Also, make sure to maintain semi-colons at the end of each line.

Disclaimer 1 (use v0.905 for batch runs for now): I just discovered a bug in PuMAv0.906 that prevents batch mode from running properly. I am working to correct it. In the meantime, you can use v0.905. Go to the PuMA downloads page and select "All downloads" from the dropdown menu. Then download v0.905. The only major difference between v0.905 and v0.906 is that v0.906 allows linking/unlinking of values across subsets of the data to be different for different parameters of a partitioned analysis.

If you don't need this functionality, v0.905 is fine to use. In any case, the bug in v0.906 should be fixed shortly.

Disclaimer 2 (your system still needs to support a GUI): While input can be provided in a file, PuMA is still built on a GUI framework and silently opens a non-existent GUI window in the background when run in batch mode. This means that it currently can still only run on systems that support GUIs (this excludes most clusters). This is obviously poor design and we're working to fix this. Thankfully, most PuMA runs are not terribly computationally intensive.

Interpreting PuMA Output

The "TREEOutfiles" folder contains individual trees taken from the original posterior distributions that were used to simulate the posterior predictive data sets. The "SeqOutfiles" folder contains all of the simulated posterior predictive data sets. Each of these should be the same size as the original empirical data set.

The test statistic output file (whatever you chose to name it) contains the multinomial likelihood values for each posterior predictive data set, the multinomial likelihood for the empirical data set, and the resulting posterior predictive *p*-value. If your *p*-value falls in the tails of the null distribution (near 0 or 1), the multinomial likelihood from the empirical data is improbable under the assumed model, suggesting that some aspect (or several aspects) of the assumed model and priors may not be appropriate for those data. DO NOT interpret these values if your input data set contained missing data. None of the simulated datasets will contain missing data, so this *p*-value (or any comparison of the multinomial likelihoods between the empirical and simulated data) will not be meaningful.

If you would like to visualize the null distribution of multinomial likelihoods, and compare it to the empirical value, copy and paste the "pumaHist.r" file into your analysis folder. This is an R script. If you open it in R (or a text editor), simply change the name of the file containing your test statistic values. You may also need to change the y-coordinates of the arrow used to denote the empirical value (the 2nd and 4th values passed to the arrows()

command on the last line of the script) to make it appear on your plot. If it doesn't appear at first, look at the y-axis scale and then pick appropriate values. After setting them, re-run the script. Copy and paste these commands into an R terminal to create the plot.

PuMA Out to AMP In

Replicating Patterns of Missing Data

If you would like to use inferential test statistics (comparing inferences between empirical and simulated data) and your input data file contained missing data, you should replicate the same pattern of missing data in each simulated data set. I've included a Python script with this tutorial that will copy the pattern of missing data in an empirical dataset into the posterior predictive data sets simulated by PuMA. It is called "repMissPatterns.py". Before running it, you will need to open it and set the name of your empirical data file at the top. Also, make sure the script is located in the same folder as your PuMA output. To run it, simply type:

```
$ ./repMissPatterns.py
```

This script will create another folder called "SeqOutfiles_wMiss" that includes the same simulated datasets as the "SeqOutfiles" folder, but with the appropriate distribution of missing data. Note that this script only deals with full ambiguity codes ('-', 'N', or '?') and not partial codes.

Running MrBayes on PP Datasets

The most time-consuming step in performing posterior predictive model assessment with inferential test statistics is the analysis of the simulated data sets. Here we're limiting the number of replicates we're running (and the intensity of the inference) to make this tractable. Once you've appropriately replicated any patterns of missing data that may be present in your data, you need to set up MrBayes analyses for each PP simulation. To do this for our example, you can run

```
$ ./setupSimAnalyses.sh
```

to create a new folder “mbSimAnalyses” that contains all of the new data sets and have appropriately modified your original MrBayes command file to run for each one separately. There is one option at the top of that script that allows you to tell the script what suffix you’re using to denote MrBayes command files. Once your new MrBayes command files are set up, you need to then run them. A very simple script to do this for the tutorial example can be run as

```
$ ./runSimAnalyses.sh
```

Once these analyses have finished running, you can set up your MrBayes output files according to AMP naming conventions in a separate folder. This can be automatically set up for this tutorial with

```
$ ./ampSetup.sh
```

This script has user options at the top to set the root file names for MrBayes output from your empirical and simulated analyses. It will then create a folder call “ampAnalyses” and place AMP v0.98 in that folder as well (assuming you have a folder called “AMP” that contains the amp python script and this folder is in the same parent folder as your primates analysis – see the structure of the tutorial files and folders).

Running AMP

Once all of the MrBayes output files are in the right folder and named properly, we can simply run AMP to perform posterior predictive assessment of model fit based on inferences. To get some verbose description of AMP’s options and usage (there’s currently not a manual – we’re working on it), try this

```
$ ./amp0.98.py --help
```

If this doesn’t work, amp may not have execute permissions. To change this, type

```
$ chmod +x amp0.98.py
```

You can explore all of AMP's options by trying different combinations of test statistics, but here's one command line to get you going

```
$ ./amp0.98.py -q 9,10,99,100,999,1000 -eiTV -m 50 -lut
  -o ampPrimatesJC.out -v primates_JC 25 4
```

Note: don't copy and paste this line from Word or a pdf! You'll need to type it by hand on the command line to make sure the hyphens are proper and can be read by AMP.

This might run kind of slowly, but it's calculating a bunch of different test statistics. Also, we are about to release a new version of AMP that speeds this process about an order of magnitude. I wrote versions of AMP up to 0.98 mostly as a proof-of-principle program. My graduate student, Brad Nelson, has improved upon all of my naïve, horribly inefficient code. If you're interested in using AMP for your own analyses, keep checking the program webpage for updates.

Interpreting AMP Output

All of the AMP output is stored in the output file that you specified. Simply open it up in a text editor. If you specified the –v flag, it will include the test statistic values for all of the simulated and empirical data sets. In either case, it will include the *p*-values for each test statistic. Scroll through the file. Note which of the inferences have low *p*-values. Are some more plausible than others?