Genomics of Disease in Wildlife Beast Tutorial

First. Use this link to find the files associated with this exercise:

https://drive.google.com/drive/folders/0B8EJy8MFHGLnOW1LX1NsM05uNEk?usp=sharing

Second. Some helpful resources. Some of these are more essential than helpful!

Keep the manual in hand and refer to it often when using Beast! It comes with the main Beast package when you download it but you can also download it here:

http://www.molecularevolution.org/molevolfiles/beast/BEAST14_MANUAL-7-6-07

There is a great Beast user forum, which should be your first place to go when you run into a roadblock. Chances are others have had the same issue so the answer may already be there but if not you can post your question and someone will probably answer you soon.

https://groups.google.com/forum/#!forum/beast-users

There are great tutorials online for most of the analyses Beast can perform so if you want to learn how to do more cool stuff with the program check them out:

http://beast.bio.ed.ac.uk/tutorials

If you don't want to bog down your computer for days to weeks when running Beast on large datasets there is a great open access server online. All you need is your Beast input file (which you are about to learn how to make!).

https://www.phylo.org/

Finally, there is a book from the creators of Beast that provides background on many of the core functions of the software and goes into more detail about

parameters, priors, etc. I find it extremely useful to keep it by my side when using Beast.

Today we are going to use an alignment of feline immunodeficiency virus isolates to build a dated tree in Beast and along the way learn all of the additional programs that are necessary when using Beast.

As a quick overview…

Beauti is a program that is used to create the input file for Beast. This is where you will select the type of analysis you want to run and the provide information about priors for the parameters that the analysis will be estimating.

Beast uses a Bayesian MCMC algorithm to produce rooted phylogenetic trees and estimate many values for evolutionarily important parameters along the trees (node dates, evolutionary rates, etc.).

Tracer is a program used to evaluate .log files that are produced during a Beast run so that you can determine the quality of the run and the quality of the output parameter estimates.

TreeAnnotator will find the 'best' tree from the thousands of trees sampled during the Beast MCMC run and summarizes posterior parameter estimates on the tree.

FigTree is used to visualize the final tree and can be used to label nodes, branches, etc. for publication.

Ok. Lets get Started!

Go to the Beast folder in the GDW_apps on your desktop All of the following programs are located here except for FigTree which is in its own folder within GDW_apps.

Open Beauti

File > Import Data
Select '**PLVAB_aln.nex**'

You should a summary of the file that you loaded under the 'Partitions' tab. Don't worry about the details now. If you don't get an error message you're doing well.

If you want to get complicated later on and analyze multiple data sets (i.e. different loci) or simultaneously analyze non-genetic traits you would import those additional data sets here before moving on.

The **Taxa tab** is primarily used when you want to separate samples a prior by taxa to determine the time to most recent common ancestor of a pre-defined group of samples. We won't use this today…but feel free to explore later☺

Select the **Tips tab**.

Select 'Use tip dates'

Click ‘Guess Dates’

Now click ‘Defined by a prefix and its order’

Under ‘Order’ select “last”

Under ‘Prefix’ select ‘.’

Select ‘Parse as a number’

Leave the rest of the settings at default (unchecked) and hit ‘OK’

The Date column of the ‘Tips’ tab should now be populated with dates ranging from the 80’s to the 2010’s. Ignore the precision column for now.

Q: What is the height column and why does that matter for this analysis?

Click on the **Traits tab** (even though we aren’t using it today☺).

This is where you would define traits that you want to model over the tree. Just like I used serotype in my bluetongue virus example. You could also use location, date, phenotype, etc. But this is beyond the scope of today’s introduction so let’s move on to the ‘**Sites’ tab**.

Ok. This is where we start to get into the good stuff.

Here’s where you should ideally know enough about your sequences and the genes/organisms they came from that you can inform the model you’re using to analyze them. But since we are analyzing my data and not yours…you’ll just have to take my word for it that the following parameters are good ones for these data.

Under ‘Substitution Model’ select ‘HKY’

Use estimated base frequencies

Use a gamma distribution to estimate site heterogeneity (Q: WTF is this referring to?)

Keep the number of Gamma Categories low unless you have reason to do otherwise…select 4.

Partition codons into two groups ‘(1+2),3’

Use the first two check boxes to unlink the substitution rate and rate heterogeneity across the codon partitions we just defined. This will allow different rates to be estimated for the (first/second) and (third) codon positions. I

tend not to estimate different base frequencies across the codon positions but the third box is up to you.

The last two boxes on this page…'Yang96' and 'SRD06' are options for 2 predefined sets of the above parameter options that you can use if you want. We have basically used the 'SRD06' set of parameters today.

Move on to the **'Clocks' tab**.

Under 'Clock type' chose: 'Uncorrelated relaxed clock'

The default 'Relaxed Distribution' is Lognormal. Leave this as is.

Q: What is the difference between a strict clock and relaxed clock?
Q: What is an 'uncorrelated' relaxed clock?
Q: Which type of data sets (taxa, genes, evolutionary questions, etc) might one or the other of these options be most appropriate?

Next up: **Trees tab!**

For today we will keep it simple and use a 'Constant Size Coalescent' Tree Prior but this is one area that BEAST has grown in recent versions. I honestly don't know what several of these Tree priors do but this there is real potential here to apply some cool analyses to different types of data and different evolutionary questions. For example, the Bayesian Skyline option can estimate the timeline of historical changes in effective population size based on the pattern of coalescence in your dataset. Pretty cool right??? Most of these options have papers that describe them so as your BEAST skills become more advanced you can explore these…

Leave the default of 'Random Starting Tree' and move on to the 'States' tab.

If you want to do ancestral reconstruction you can specify some options here. But we don't. So let's move on to the **'Priors' tab**.

This is another place it is important (although not necessarily critical) that you have some knowledge of the sequences/taxa that you are analyzing. Each of the lines on this page are present because of the parameter choices we have made

thus far. If you select different model parameters next time you use Beauti…you will have different priors when you open this tab.

For today we will leave them all at default except the following:

'allMus': Change this to lognormal with an initial value = 1 and log(stdev) = 1

'ucld.mean': change this to lognormal with an intial value of 0.1 and a standard deviation of 0.5

Click on the **'Operators' tab**. Today (and generally) you don't need to mess with this stuff as long as the 'Auto Optimize' box is checked in the upper left corner. Sometimes the output from a run will give you a warning that how the chain samples a certain parameter needs to be tweaked and you can use this tab to do just.

Click on the **'MCMC' tab**.

Set the length of the chain to 1,000,000 and log the parameter estimates every 1,000. These numbers are inadequate but it will let you get output files fast (and I have an output file from longer chains we can look at ☺).

[As a rule of thumb you want to end up with 10,000 logged parameters/trees at the end of a Beast run. This means that if you run the chain for 10^8 steps you'll log parameter estimates (and trees) every 10^4 steps. Make sense? If you run the chain less steps you log less frequently. This is at least a good way to start. There are reasons we will get into later why you'd want to log more frequently too.]

You can change the file stem name if you want or leave it as is. If you start to do multiple combinations of Beauti parameters and Beast runs from the same alignment file it is nice to label the output differently. I use numbers, then letters, then obscenities in that order as I go through the process many times until it comes out right!

Ok. Click 'Generate BEAST File' and save it where you want.

A pop up window till appear…click ok. This is your chance to change any additional parameters from default but today we will leave them alone.

Wahoo! Step one done!

Open Beast

Load your newly created 'file.xml'. Unclick the 'Use Beagle' option and leave the rest of it at default. Select 'Run' and you are off to the races. Easy compared to Beauti right?

[I have a .xml file ready to go if you didn't make it through Beauti or if you run into errors when you run Beast 'PLVAB_aln_GDW.xml'.

While that is running…

Open Tracer

File > Import Trace File

Select your 'file_stem.log' file that is in progress from your current Beast run (yes you can view it before the Beast run is complete). You should also open '**PLVAB_aln_GDW.log**' which is a Beast output file from the same alignment we used earlier but with a few extra parameter estimates. It will work well to let you see what a log file will look like after the end of a sufficiently long MCMC chain.

You can also drag and drop files into the 'Trace Files' area and open multiple trace files simultaneously to compare runs.

Look at the mean posterior estimate of each parameter value within the **'Traces' pane** on the bottom left….do they make sense? Ok. Probably none of this will make sense because you don't know much about the dataset used but this is where you will evaluate parameter estimates to make sure they make sense when you analyze your own data! This is a good time to go back to the prior distributions and values we entered into Beauti to see how our choices may have influenced the outcome, or which parameter estimates may be wildy different than we thought they would be.

The **effective sample size ('ESS')** value is an important metric to use to evaluate if you have enough samples from your chain to have accurate estimates of your parameters. I forget the exact numbers here but if this is less than 100 it will be red to give you a warning that estimates should not be trusted. Between 100 and 200 they are yellow…caution. Above 200 is considered acceptable and there is likely little benefit to going beyond that. To increase ESS low ESS values you can: 1) run your chain longer (remember where you change this setting in Beauti?), 2) sample your chain more frequently, 3) run multiple independent runs of Beast and combine them (not covered here but you use LogCombiner to do this), 4) chose a less complex model as your data may not be informative enough for highly complex parameter estimates.

You can visualize the distribution/frequency of values sampled throughout the length of the MCMC chain by clicking on the tabs above the right pane. The **Trace** is especially useful to know if you have achieved good sampling of the prior you defined in Beauti…it should look like a '**spiny caterpillar**' (or at least that's how it was taught to me). You can also select two parameters (by holding cmd) and then click on **Joint Marginal** to see how they relate to one another. Try this for the CP1+2 and CP3 kappa values…were correct in estimating these codon partitions separately? Do the same for CP1+2 and CP3 mu values. Thoughts? What does this say about the flexibility of different codon positions to mutate/evolve over time?

Ok. Spend as much time as you want with Tracer but when you're ready let's move on.

Open TreeAnnotator

Go down toward the bottom and choose your **input file** which is one of the files output from Beast ending in '.trees'.

I have a file you can use here as well since your Beast run may not be done yet and (it probably isn't the best example anyway): '**PLVAB_aln_GDW.trees**'

Choose your **output file** name ending with '.tre' and the location you want to save it.

Now go back up and enter 10% of the number of steps in your chain into the "Specify the burnin as the number of states" field.

i.e. if you are analyzing your Beast output from a run of 10,000,000 steps then you'd have a burnin of 1,000,000. For a chain length of 1,000,000 steps (like the inadequate example I had you do) enter 100,000. (Hint: If you forget the number of steps you have in a given chain you can open the log file in Tracer and it will tell you how many steps were in the run.)

Select '**Maximum clade credibility tree**' and '**Median node heights**'

Sweet.

Now select 'Run' and let it go! Don't worry…It doesn't take too long.

This program is annotating **a single maximum clade credibility tree** (the tree with the highest node support values from all 10,000 or so trees logged in your .trees file) with lots of the posterior parameter values that were estimated across all of the 10,000 or so states in your .log file. Pretty cool right?

Now let's open the MCC tree and check it out. This is the moment we've been working towards. Exciting I know.

Open FigTree (within a separate folder in GDW_apps)

Use File > Open

Select your output.tre file that you made in TreeAnnotator

Now you can use the panel on the left to play with how the tree looks, color branches and nodes, and visualize many of the parameters we estimated during the Beast run. For example, click '**Node Labels**' and look through the options of

what can be displayed then click on '**Node Ages**'. Because we entered our sample dates in years (when making the input file in Beauti) the numbers now displayed on each node correspond to median posterior estimate of the age of that node (the number of years since the most recent sample that each node coalesced/diverged). You can also use the 'Node Labels' to view the **posterior support** for each branch/node (similar to a bootstrap value on ML trees).

Play around with the different options and see what you can learn about the samples and model estimates.

Congratulations! That's your first run through Beast. If you've made it this far you deserve a pat on the back and a beer.