

Phylodynamics and Phylogeography Practical

Dr Samantha Lycett, Roslin Institute, University of Edinburgh

Workshop information

Pathogen sequences (especially viruses) mutate rapidly over time, and this information can be used to infer how the disease is spreading. In an outbreak, virus sequence data can be used to infer possible source dates, locations and possibly species. In this workshop you will learn how to run a basic analysis in BEAST on virus sequences and create and interpret a time scaled tree.

Software:

MEGA

<http://www.megasoftware.net/>

For sequence alignment and simple tree building (note it runs on Windows and Macs, but for Macs there is sometimes a software issue and I had to have a couple of attempts to get it to run properly)

Tempest

<http://tree.bio.ed.ac.uk/software/tempest/>

For checking clock-like-ness using the trees (non-time scaled) created in MEGA.

BEAST 1.10

<http://beast.community/>

This is the main software for time scaled trees, BEAUTi and TreeAnnotator are also in the download zip / dmg file and these are also required.

Note version 1.8.4 is mostly OK for this too – but BEAST 2 is something different.

[Also note – you need Java if you don't have this already – http://beast.community/install_on_windows]

BEAGLE

<http://beast.community/beagle>

High performance library, speeds up certain calculations within BEAST. Also required to be installed for spatial analysis (and you definitely need this for version 1.10).

Tracer

<http://beast.community/tracer>

To analyse and summarise the BEAST MCMC trace output files

FigTree

<https://github.com/rambaut/figtree/releases> and <http://beast.community/figtree>

To visualise the time scaled tree (and any nexus or newick format tree)

Other software:

Useful for post-run analysis but not covered in detail in this practical:

Spread <https://rega.kuleuven.be/cev/ecv/software/spread> (original)

Spread3 <http://beast.community/spread3> (improved)

Google Earth https://www.google.co.uk/intl/en_uk/earth/

R (3.3.2+) + packages: ape, maps, mapdata, mapproj, OutbreakTools, RgoogleMaps, shiny

Data

Sequences: [cov_net_sim_mper2_120genomes.fas](#)

120 sequences of SARS-CoV-2 from humans. These are actually from a simulated epidemic from a stochastic individual based metapopulation model, but are based on a real sequence and real sequence parameters. Just like the real thing, only slightly 'cleaner'.

Traits table: [cov_net_sim_mper2_120genomes_discrete_traitsTbl.txt](#)

This tab separated data file contains the corresponding 'trait' data for each sequence, in this case the discrete Place values (cities).

Neighbour joining tree: [cov_net_sim_mper2_120genomes_ape_tn93_nj.nwk](#)

A simple neighbour joining tree from the sequences has already been made (see Step 2 for example), and is included here for reference.

BEAST parameter Log files: [*.log.txt](#)

These are the log files from Step 3

BEAST Tree files: [*.trees.txt](#)

These are the tree files from Step 3

MCC tree files: [*_mcc.tre](#)

The Maximum clade credibility tree files (Step 5)

Step 1 – Sequence Alignment / Check Sequence Alignment

Data familiarisation: Open the *.fas file in MEGA, there you will see the nucleotide alignment. If you click on the translate tab, you will see the corresponding protein sequence starting at ORF1a, but beyond this it is not in-frame.

These sequences are aligned to each other. In comparison to the standard COG-UK SARS-CoV-2 sequences, then they are missing the first 266 flanking N's; and also any deletions in the Spike protein have additionally been removed.

For RNA viruses bear in mind that there are sometimes well known insertions/deletions in the surface proteins coding genes. This is true for Spike in SARS-CoV-2, and also in Influenza (Hemagglutinin for highly pathogenic H5 or H7 strains, and in the Neuraminidase stalk region). However, for a within-year within-country single outbreak (of a lineage), there should not be many, or even any, insertions / deletions (so if your alignment has a lot of gap regions, then it is probably not OK and needs further work).

If you adjust the alignment, save the file by "exporting" as fasta format with the name of *_al.fas (or similar); however this is not necessary to do with this practical data.

<http://www.megasoftware.net/> (link for MEGA software for reference)

Step 2 – Simple trees and molecular clocks

A neighbour joining tree has already been made and included, but below are the details of how this can be done; and in case you want to repeat this yourself. When you have a tree, go to Step 2b.

Step 2a: Create Neighbour Joining Trees (optional, you can use the supplied tree)

A neighbour joining tree (the most similar sequences are joined together first) can be created very quickly in MEGA. Open the aligned sequences in fasta format file *.fas, and chose Phylogenetic analysis (left hand side menu). As with many of these things there are a lot of options, but the ones which are good for this data are the following – these settings are applicable for SARS-CoV-2 and most other viruses (e.g. Influenza). But if you have bacterial SNP data, or not much diversity in your set, then try the HKY substitution model.

Test of Phylogeny = none
Substitution model = Nucleotide
Model / Method = Tamura-Nei model
Substitutions to include = d: Transitions and Transversions
Rates among sites = Gamma distributed
Gamma parameter = 1
Pattern among lineages = Different (heterogenous)
Gaps / Missing data treatment = Pairwise deletion

Note that when you click Phylogenetic Analysis, you will have to move the alignment window out of the way in order to see the other one behind. On this click Phylogenetic analysis again, choose Neighbour joining and then enter the model settings as above (these are Tamura-Nei Model, with transitions and transversions, gamma distributed rates among sites with gamma parameter = 1, different pattern among lineages and pairwise deletion for gaps and missing data treatment. No need to use a test of phylogeny here).

(you can formally test which are the best settings using Likelihood scores, and there are separate programs to do this, e.g. ModelTest. This is different to “test of phylogeny” which is about changing the shape of the tree to find the best one.)

When the tree is built – save it as the *.mts native format and also export as “newick” (use file extension *.nwk).

You can now use FigTree to open the exported newick file (*.nwk) – FigTree will let you change the order that the branches are displayed in on the vertical axis (allowed because it is not changing the underlying tree topology), make the tip labels (sequence names) bigger and also colour them.

<http://tree.bio.ed.ac.uk/software/figtree/> (link for FigTree software for reference)

Step 2b: Adding a time-scale (quick)

Open the supplied or previously generated newick tree file (*.nwk) in TempEst. This program tries a simple method to fit a molecular clock to the tree data, specifically it will try to perform a root-to-tip regression meaning it will fit a straight line to “genetic distance from root” (y-axis) vs time to sampling (x-axis). The genetic distance from root is given by the tree (add branch lengths from sampled sequence to root) and the sampling time is obtained from each sequence name.

The dates in the format yyyy-mm-dd have been appended to the end of the sequence names, so you can click “guess dates” to add dates of the sequences into the program.

However, when the tree was made, no particular root was specified; tick the find best root box to make the program search over different possible root positions.

- What is the estimated mutation rate (slope of the root-to-tip fitted line, units are substitutions per site per year) ?
- What happens with the different root-to-tip estimating methods – do the trees look different ?

If you notice that one sequence seems out of place in the root-to-tip plot and the residuals plot, highlight it and click to the main tree plot pane to see which one it is. It is likely that this sample contains sequencing errors.

The re-rooted trees can also be exported for further use, click File -> Export tree. Note that this will actually export as a nexus format tree, so give it a *.nex filename when saving.

<http://tree.bio.ed.ac.uk/software/tempest/> (link to TempEst for reference)

Step 3 – Timescaled trees using BEAST

Step 3a: Make the BEAUTi xml

Now the data is prepared, you are ready to make your first BEAST tree. To do this you need to make the *.xml configuration file in **BEAUTi**, and there are a lot of steps and settings, but essentially you work your way along the horizontal tabs at the top.

The ** are important parameters and you would normally try a few of these in an initial analysis

Tab 1 (Partitions) – do File -> Import data : import the fasta *.fas file you made (change the file selector to all files)

Tab 2 (Taxa) – no action

Tab 3 (Tips) – click use tip dates; then Guess dates. The sequence name have been appended with the decimal date, so select 'Defined just by its order' and Order: last

Tab 4 (Traits) – will be used in the next section (no action just now)

Tab 5 (Sites) – Set the nucleotide substitution model, I recommend using the TN93 for this data and including Gamma site heterogeneity (this is also generally good for SARS-CoV-2) **

Tab 6 (Clock) – Choose Strict Clock ** other choices would be Uncorrelated relaxed clock with Lognormal distribution.

Tab 7 (Trees) – Choose between Coalescent: Constant Size, Coalescent: Exponential Growth or Skygrid **

Tab 8 (States) – No action

Tab 9 (Priors) – some can be set here (although the defaults in 1.10.4 are often OK)

- clock.rate (if Strict clock) or ucl.d.mean (if relaxed clock): from the TempEst analysis you have an estimated clock rate of around $1e-3$ per site per year, you can use this to set a normal distribution prior for the clock rate if you like (a strong prior). Normal with mean = $1e-3$, std = $1e-3$, range 0-0.1 is OK on SARS-CoV-2 data **

Tab 10 (Operators) – no action

Tab 11 (MCMC) – for a small dataset, the default MCMC parameters are OK (that is the chain length = 10,000,000 and 'log parameters every' = 1000, which gives an output of 10,000 trees and the corresponding log parameters file). For larger data sets increase these values.

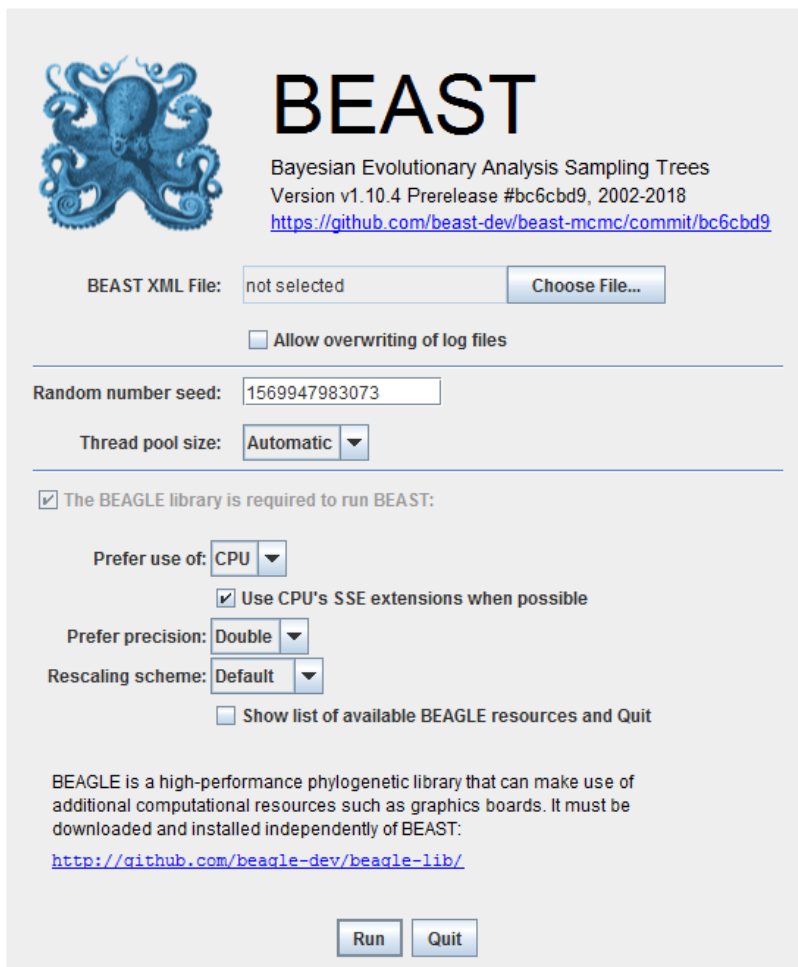
Change the filename stem to reflect the model choices – I use things like this which seem to work well: (fasta file name)_(subst model)_(clock model)_(trees prior model)_(replicate) for example:

- (fasta file name)_TN93G4_strict_constPop_1
- (fasta file name)_TN93G4_strict_expGrowth_1

Finally you are done, now click the generate BEAST file to make the (filename stem).xml

Step 3b: Run BEAST

Open **BEAST** (double click), you will get an xml selector window - Choose your xml and click RUN !



BEAST
Bayesian Evolutionary Analysis Sampling Trees
Version v1.10.4 Prerelease #bc6cbd9, 2002-2018
<https://github.com/beast-dev/beast-mcmc/commit/bc6cbd9>

BEAST XML File: not selected

☐ Allow overwriting of log files

Random number seed: 1569947983073

Thread pool size: Automatic

☒ The BEAGLE library is required to run BEAST:

Prefer use of: CPU

☒ Use CPU's SSE extensions when possible

Prefer precision: Double

Rescaling scheme: Default

☐ Show list of available BEAGLE resources and Quit

BEAGLE is a high-performance phylogenetic library that can make use of additional computational resources such as graphics boards. It must be downloaded and installed independently of BEAST.
<http://github.com/beagle-dev/beagle-lib/>

Actually you can do it from the command line too – there will be a `beast.jar` in the downloaded part, something like this would be OK:

```
java -jar beast.jar your.xml
```

(do `java -jar beast.jar` with no xml to get the BEAGLE & GPU etc run time options)

The 120 sequences will run for approx. 30 mins if allow to go to completion

A parameters log file (*.log.txt) and trees file (*.trees.txt) will be generated as well as a diagnostic *.ops file (at completion).

I suggest that you stop the run once it has gone for a short while and use the runs I prepared earlier !

Step 3c: Analyse Log File Output using Tracer

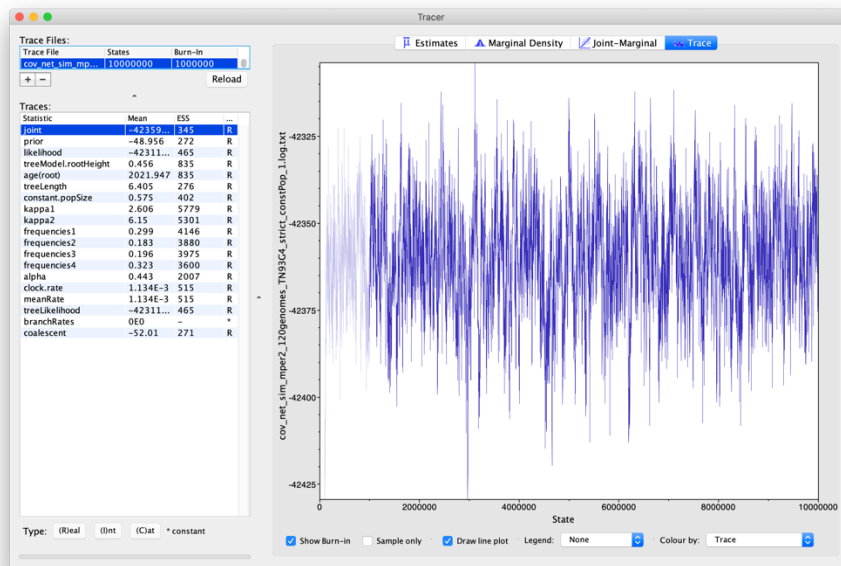
In the `beast_output_files`, there are 3 log files corresponding to the 3 effective population size models.

Open a `*.log.txt` file in **Tracer**. For each parameter in the model (and MCMC output) you can see its statistics and trace.

To quickly see if the MCMC is OK, look to see whether the Effective Sample Size of all parameters is ≥ 200 .

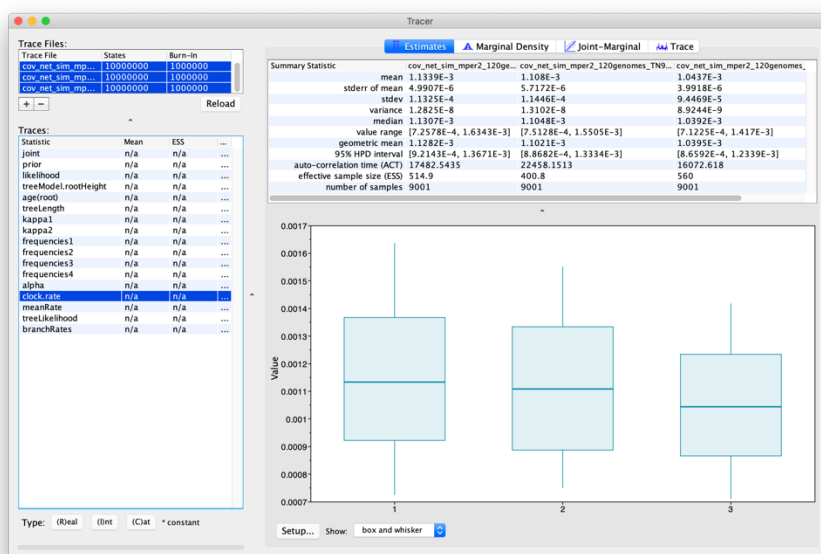
Tracer automatically assumes a burn-in of 10% (the greyed out part), but you can change this if you like.

A nice trace looks something like this:



From the log files of `strict`, `expGrowth` and `skygrid` compare the common parameters – you can load multiple files at once and show box-and-whisker (or violin) plots of all three at once (Estimates tab).

- What is the estimated root height and `age(root)` ?
- What is the estimated clock rate ?

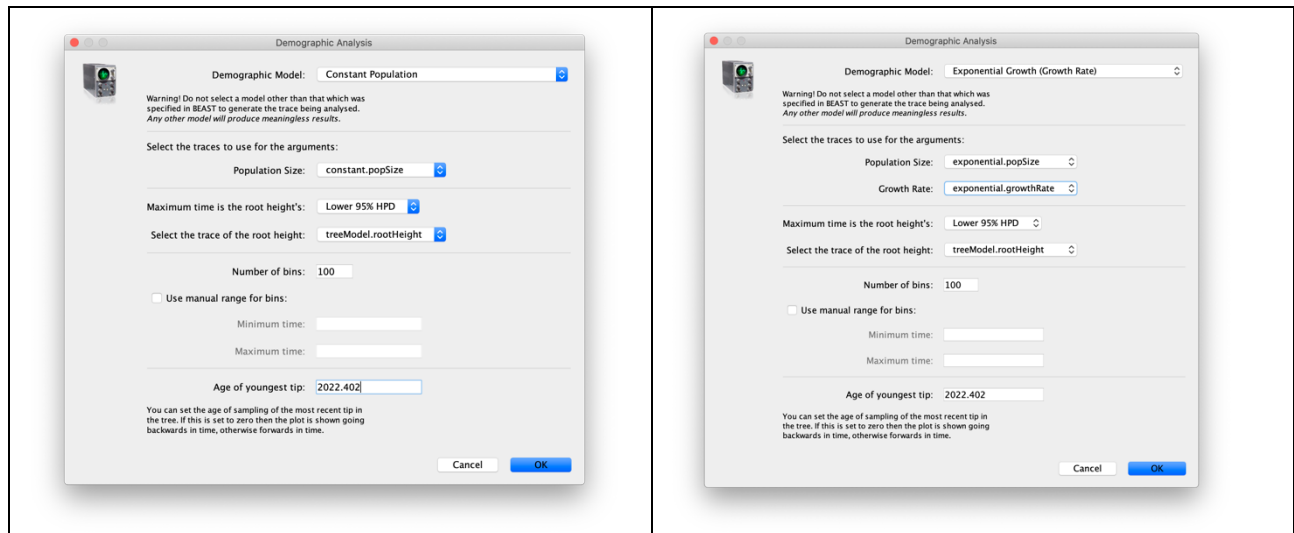


Step 3c continued: Effective population size estimation

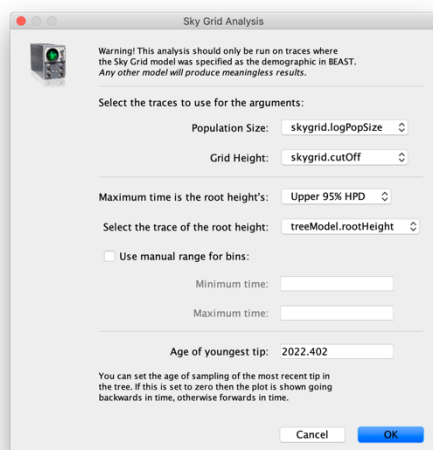
The effective population size can be reconstructed as a function of time from the Analysis menu (see top of tracer)

For constant population size and exponential growth, highlight the log file you want to analyse (left hand side) and choose 'Demographic reconstruction'. Here you need to enter the age of the youngest tip, which in this case is 2022-05-28 i.e. 2022.402 in decimal dates.

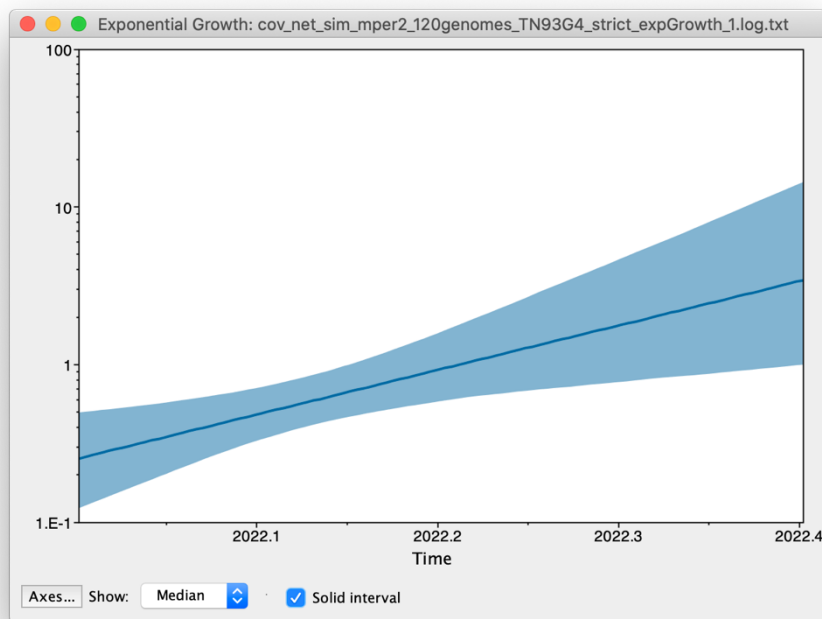
When switching between Constant population and Exponential Growth, make sure that the correct parameters have been selected in the pane.



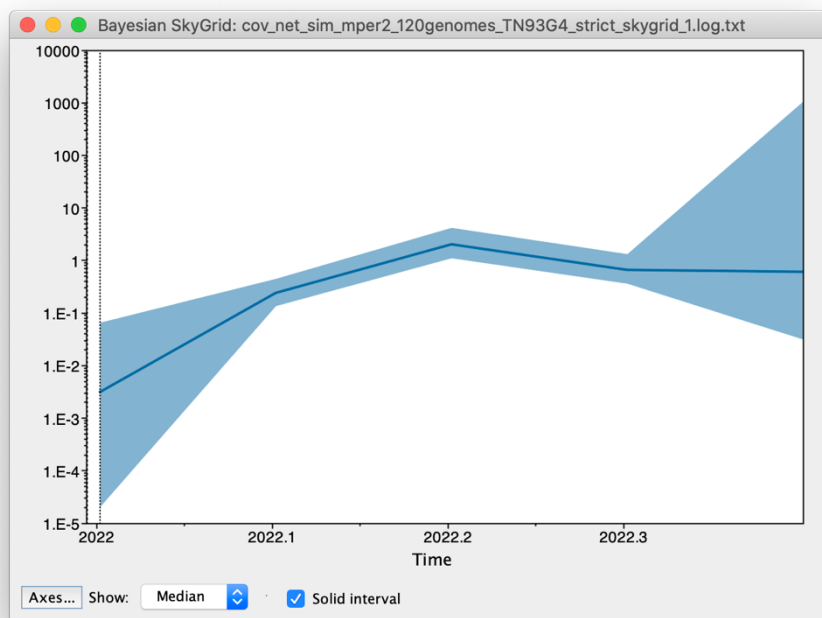
For the flexible population size over time skygrid model, choose Sky grid analysis rather than Demographic Reconstruction (because it is a special case)



The results from Exponential and Skygrid should look like the following. You can change the axis from log to linear, and also export the data as a file (e.g. to re-plot in R) and image as PDF.



Exponential model effective population size over time



Skygrid model effective population size over time

Step 3d: Make the MCC Tree

The posterior set of trees are in the *.trees.txt file. DO NOT OPEN THIS FILE IT WILL BE TOO BIG, instead use **TreeAnnotator** to summarise the trees for human viewing.

From a default run, there would be 10,000 trees in the posterior tree set.

Tree Annotator settings:

- There are 10,000 posterior trees, and a burn-in of 10% seems good from tracer, so you would specify a burn-in of 1000 trees.
- Also, the default settings of Posterior probability limit (=0), Target Tree Type (Maximum Clade Credibility tree, i.e. the MCC tree), and Node heights (median heights) are good for this data.
- Choose your *.trees.txt file as input, and give it a sensible name for output – I recommend: *.tre
- Click run and create the file (it will take a few moments)

TreeAnnotator v1.10.4 Prerelease #bc6cbd9

☐ Specify the burnin as the number of states

Burnin (as states): 0

☒ Specify the burnin as the number of trees

Burnin (as trees): 1000

Posterior probability limit: 0.0

Target tree type: Maximum clade credibility tree

Node heights: Median heights

Target Tree File: not selected Choose File...

Input Tree File: G4_strict_constPop_1.trees.txt Choose File...

Output File: G4_strict_constPop_1_mcc.tre Choose File...

Quit Run

Step 3e: Viewing the MCC in FigTree

Using the MCC tree file: *.tre (DO NOT USE THE *.trees.txt FILE IT WILL BE TOO BIG), you may now see the results in Fig Tree.

There are a lot of options in FigTree, these are down the left hand side and also in the bar at the top. When you first open the *.tre it will look abit like this:

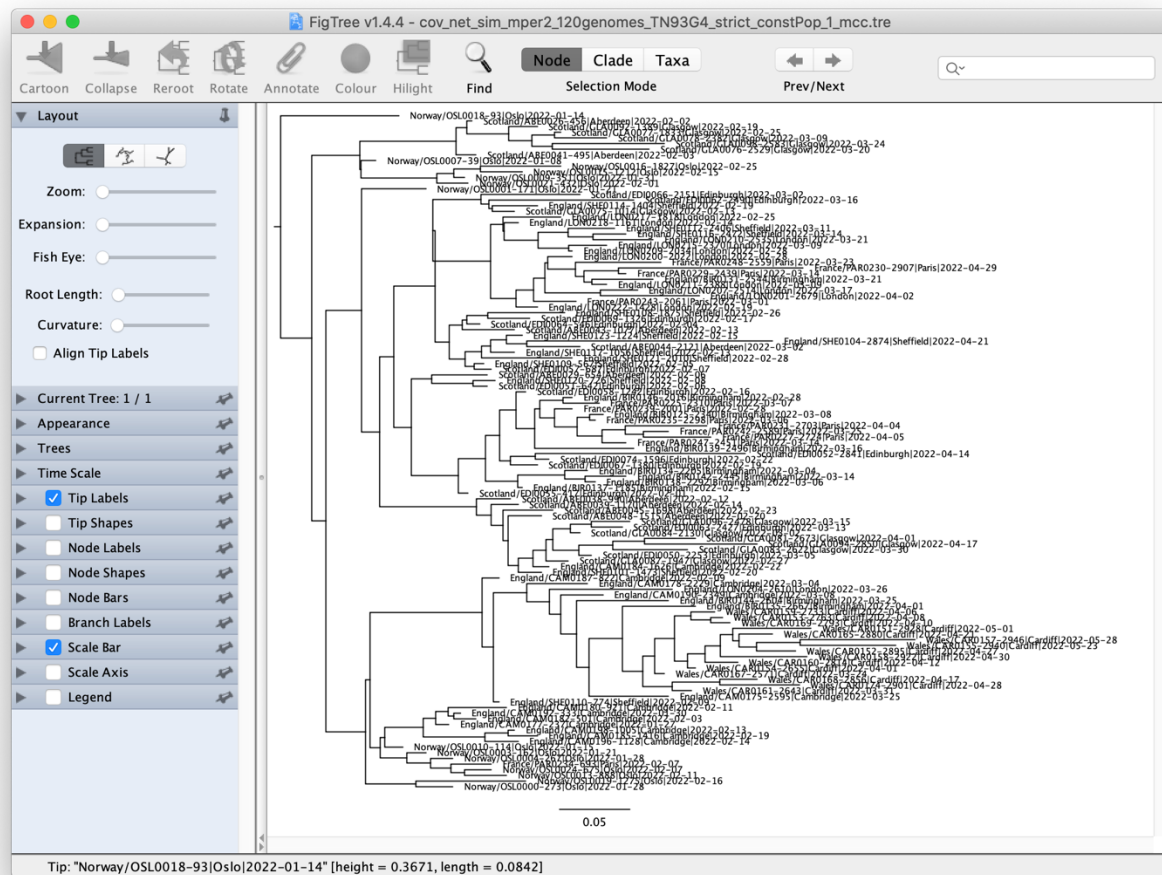


Fig Tree settings:

- Set the time scale – offset by = 2022.402 (the youngest tip = 2022-05-28, this is the one with height = 0 in Tempest)
- Click scale axis – then click reverse axis (increase font size too) - remember coalescent theory, everything goes backwards from the present
- Un-click scale bar
- (Top bars) – Tree: click decreasing node order
- Tip labels – increase the font size, and then click align tip labels (left hand bar)
- Node bars: display 95% HPD
- Node labels: display Node ages, but turn the sig digits down to 2
- Save this as a *.figTree file
- Export as an image – e.g. *.png

Step 4 – Adding traits to the BEAST analysis

Step 3 was about getting the basic time-scaled tree, but now we will add a discrete trait – Place.

I have already done these, and run the analysis to completion, but here are the instructions incase you want to make the xml (otherwise just go to Step 5).

To Configure an XML for Discrete Traits (e.g. Place)

Use BEAUTi, if you have left it open from step 3 then you can just add the new parts (otherwise you will need to repeat step 3 with the appropriate model settings).

Tab 4 (traits): Import the traits file *traitsTbl.txt

- This is a tab separated file with the first column of sequence names, and the other columns of traits.
- Click on Place, then create partition from Place (this will take you to the Partitions tab)

Tab 5 (Sites):

- Select the Places partition and choose Asymmetric model and BSSVS

Tab 8 (States):

- check that the reconstruct states at all ancestors is selected for Place

Now to steps 3b – 3e as before to run BEAST, examine the log parameters file and make the MCC tree.

Step 5 – Displaying Trees with Discrete Traits

I have made and run a BEAST analysis with Place as discrete trait, and the MCC tree is *_Place_1_mcc.tre

This tree is like the 'plain' tree, except it also has Place annotations and these can be displayed in FigTree.

In addition to the basic steps (3e), add the Place annotation as colour.

Appearance (left hand side): Colour by Place, and increase the line weight to 3

Align the tip labels

Tip labels (left hand side): Colour by Place

Tip shapes (left hand side): Colour by Place and use diamonds, size = 5

Node Shapes (left hand side): Max size 5, Size by = Place.prob, Min size 1, Colour by Place, use circles

Click Legend (left hand side): Attribute Place, increase the font size to 10

