Tutorial using BEAST v2.6.7

Introduction to phylodynamic models

Louis du Plessis

This is a tutorial to introduce you to Bayesian phylodynamic inference using BEAST2.

Contents

1 Background

Whereas phylogenetics is concerned with describing the evolutionary history of evolutionary sequences (the phylogeny), phylodynamics aims to describe the process that generated the phylogeny. Thus, whereas phylogenetics gives us a qualitative description of the data, phylodynamics gives us a quantitative description.

In this tutorial we will use different models of population dynamics to describe how the phylogeny grows over time. These models are used as priors for the tree in the Bayesian analysis (the so-called tree-prior).

This tutorial is adapted from https://taming-the-beast.org/tutorials/Skyline-plots/ (by Nicola F. Müller and Louis du Plessis), but changed to use a heterochronous dataset and with some extra analyses added on. The dataset and analyses are adapted from Hill et al. (2022).

2 Programs used in this Exercise

BEAST2 - Bayesian Evolutionary Analysis Sampling Trees 2

BEAST2 (http://www.beast2.org) is a free software package for Bayesian evolutionary analysis of molecular sequences using MCMC and strictly oriented toward inference using rooted, time-measured phylogenetic trees. This tutorial is written for BEAST v2.6.7 (Bouckaert et al. 2014; Bouckaert et al. 2019).

BEAUti2 - Bayesian Evolutionary Analysis Utility

BEAUti2 is a graphical user interface tool for generating BEAST2 XML configuration files.

Both BEAST2 and BEAUti2 are Java programs, which means that the exact same code runs on all platforms. For us it simply means that the interface will be the same on all platforms. The screenshots used in this tutorial are taken on a Mac OS X computer; however, both programs will have the same layout and functionality on both Windows and Linux. BEAUti2 is provided as a part of the BEAST2 package so you do not need to install it separately.

TreeAnnotator

TreeAnnotator is used to summarise the posterior sample of trees to produce a maximum clade credibility tree. It can also be used to summarise and visualise the posterior estimates of other tree parameters (e.g. node height).

TreeAnnotator is provided as a part of the BEAST2 package so you do not need to install it separately.

Tracer

Tracer (http://beast.community/tracer) is used to summarise the posterior estimates of the various parameters sampled by the Markov Chain. This program can be used for visual inspection and to assess convergence. It helps to quickly view median estimates and 95% highest posterior density intervals of the parameters, and calculates the effective sample sizes (ESS) of parameters. It can also be used to investigate potential parameter correlations. We will be using Tracer v1.7.2

FigTree

FigTree (http://beast.community/figtree) is a program for viewing trees and producing publication-quality figures. It can interpret the node-annotations created on the summary trees by TreeAnnotator, allowing the user to display node-based statistics (e.g. posterior probabilities). We will be using FigTree v1.4.4.

2.0.1 R / RStudio

We will be using R to analyze the output of the Birth-Death Skyline plot. RStudio provides a user-friendly graphical user interface to R that makes it easier to edit and run scripts. (It is not required to use RStudio for this tutorial).

3 Practical

In this tutorial we will analyse two datasets of SARS-CoV-2 genomes sampled from the UK. One dataset contains genomes belonging to the Alpha Variant of Concern (VOC), which is also known as the B.1.1.7 Pango lineage. This was the first SARS-CoV-2 VOC to emerge, in September 2020 (although it was only the second to be identified, in December 2020, shortly after Beta was identified in South Africa). The other dataset contains background non-Alpha SARS-CoV-2 genomes from the UK.

The main aim of this tutorial is to introduce you to using phylodynamic models in BEAST2 to estimate epidemiologically relevant parameters. We will do this by using two widely used models for inferring changes in the population dynamics over time in a well-mixed population, the coalescent Bayesian Skyline plot (Drummond et al. 2005) and the Birth-death Skyline (BDSKY) plot (Stadler et al. 2013).

After completing this tutorial you should be able to:

- Set up coalescent and birth-death skyline models in BEAST2
- Make informed choices to parameterise the models
- Sample from the prior
- Post-process skyline output and visualise the results

3.1 The Data

We use two datasets, Alpha and background. The Alpha dataset was produced by subsampling all Alpha genomes from the UK that were sequenced between 1 August 2020 and 1 February 2021 (the oldest Alpha genome only dates from 20 September 2020). We sampled 7 genomes at random from all genomes sequenced in every epiweek during this period. For epiweeks where fewer than 7 genomes are available we used all available genomes. This resulted in 154 genomes. The background dataset was produced by subsampling all other SARS-CoV-2 genomes from the UK that were sequenced during the same time period, but sampling 6 genomes per epiweek. This resulted in 160 genomes.

The genomes were all sequenced by COG-UK and the sequences are freely available on their website (be careful if you download the full alignment, the dataset contains millions of sequences and takes up a lot of space).

If you cloned the Github repository you should already have the subsampled alignments on your drive. Otherwise, you can download the file from here. Please make sure you download the raw files and that your browser doesn't insert HTML code into the file!

3.1.1 Install BEAST 2 packages

We will use BEAUti to generate the BEAST2 XML files. While the coalescent Bayesian Skyline plot is integrated in the BEAST2 core, we need to install the BDSKY package, which contains the Birth-Death Skyline model. Installation of packages is done using the package manager, which is integrated into BEAUti.

Begin by starting **BEAUti2**.

Open the **BEAST2 Package Manager** by navigating to **File > Manage Packages**.

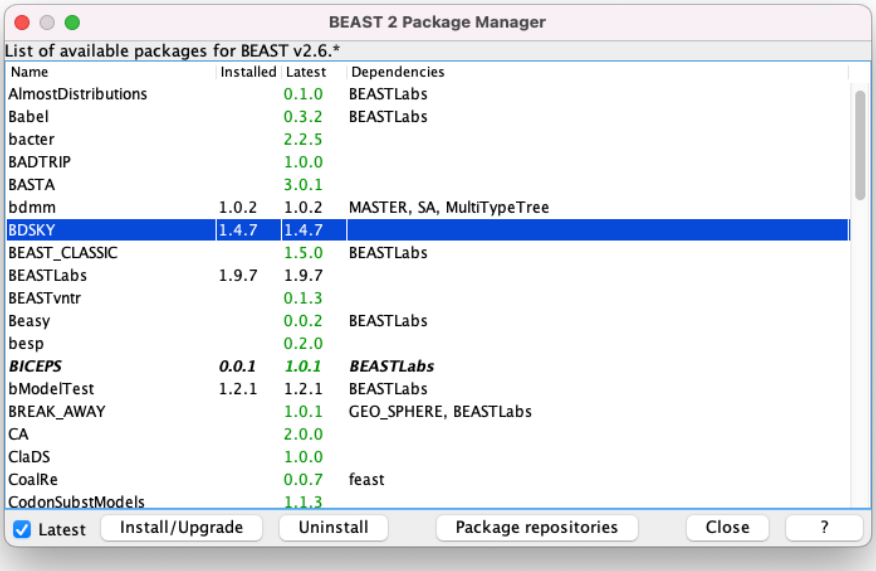

Figure 1: Install the BDSKY package which contains the Birth-Death Skyline model.

Install the **BDSKY** package by selecting it and clicking the **Install/Upgrade** button (Figure 1).

After the installation of a package, the program is on your computer, but BEAUti is unable to load the template files for the newly installed model unless it is restarted. So, let's restart BEAUti to make sure we have the **BDSKY** model at hand.

Close the **BEAST2 Package Manager** and ***restart*** BEAUti to fully load the **BDSKY** package.

3.2 Setting up a Coalescent Bayesian Skyline analysis

We will set up one analysis file that simultaneously runs the Bayesian skyline model on both the Alpha and the background datasets. In the analysis we will assume that sequences in both datasets evolve under the same site model, but on independent trees with different molecular clock models. We will use a GTR model without rate heterogeneity as the site model and a simple strict clock as the clock model.

Importing the alignments

BEAUti should already be open on the **partitions** tab.

- Load both alignments (`alpha.fas` and `background.fas`) and set the data type to **nucleotide** for both. Don't split either dataset into any partitions!

- Link the **Site** models, but leave the **Clock** and **Tree** models unlinked.
- Rename the shared **Site Model** to `site`.

Sampling dates

Select the **Tip Dates** tab.

- Select the **alpha** partition and check the **Use tip dates** option.
- Set **Dates specified** to the `as dates with format` option.
- Select `yyyy-M-dd` from the dropdown box.
- Click the **Auto-configure** button. A window will appear where you can specify how BEAUti can find the collection dates in the sequence headers.
- Select **use everything** and specify **after last |** and click **OK**.

Now repeat the steps above for the **background** partition. (You can also clone the settings as you would for a Site Model, but this doesn't work perfectly and you would still need to make some manual changes).

Site model

Select the **Site Model** tab and select **GTR** in the **Subst Model** drop-down menu. Make sure that the **Substitution Rate** is not being estimated and that the **Gamma Category Count** is set to 0.

Clock model

Select the **Clock Model** tab and check that the clock model is set to **Strict Clock** for both partitions.

Next, we will select the coalescent Bayesian Skyline plot as the tree prior for both trees.

Select the **Priors** tab.

Select **Coalescent Bayesian Skyline** in the drop-down menus next to **Tree.t:alpha** and **Tree.t:background**.

By default the skyline model has a dimension of 5, meaning that the period between the time of the most recent common ancestor (tMRCA) and the most recent sequence is divided into 5 intervals or segments. We would like to set the dimension to 10, to allow some more flexibility.

To change the dimensions of parameters we have to navigate to the **Initialialization** panel, which is by default not visible. Navigate to **View > Show Initialization Panel** to make it visible and navigate to it (Figure 2).

Set the dimensions of **bPopSizes.t:alpha**, **bGroupSizes.t:alpha**, **bPopSizes.t:background** and **bGroupSizes.t:background** to 10 each. (the default value is 5) (Figure 3).

Next, we will set a more appropriate prior for the clock rate.

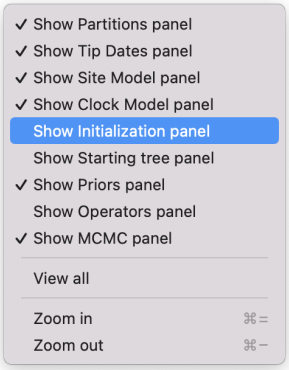

Figure 2: Show the initialization panel.

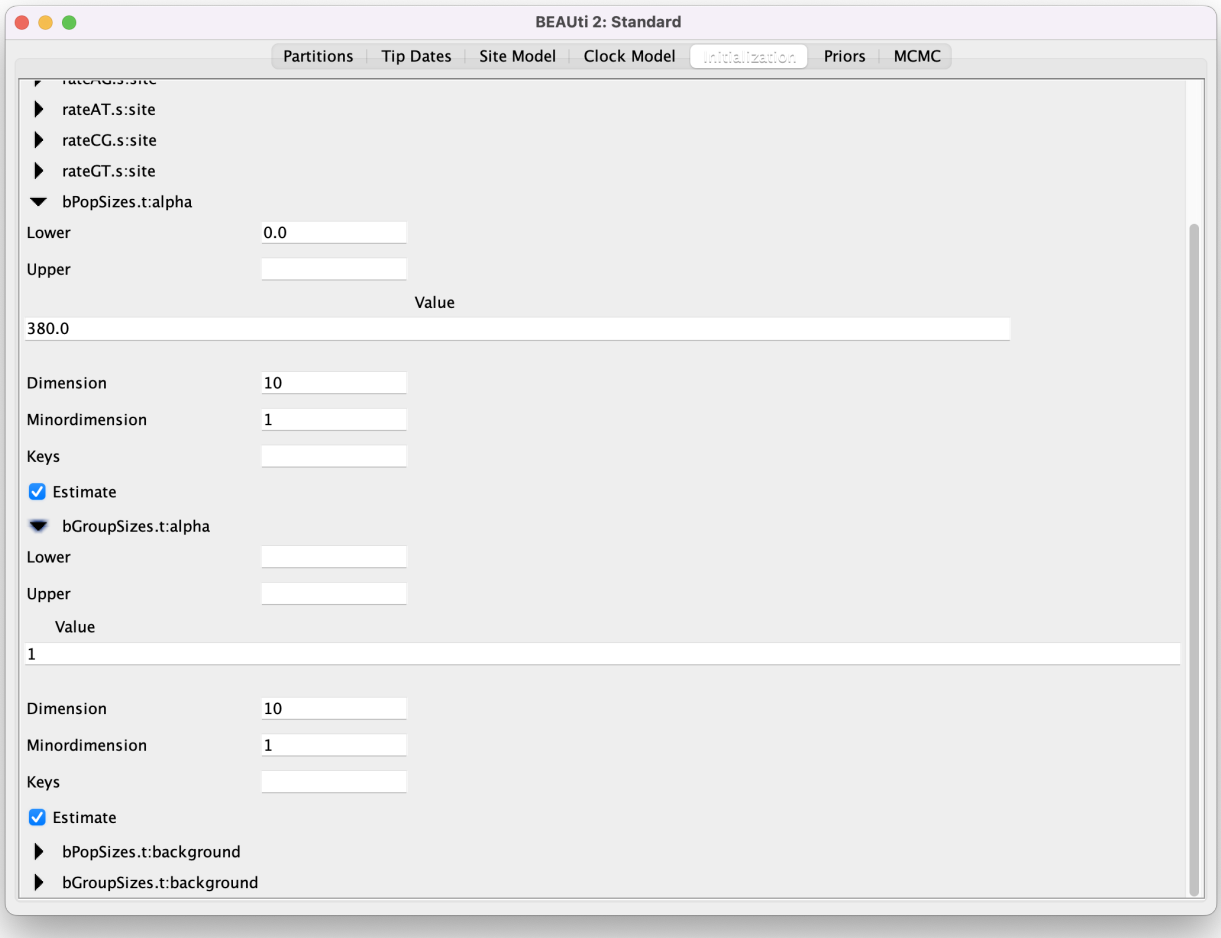

Figure 3: Set the dimension of bPopSizes and bGroupSizes to 10 (remember to do this for both trees).

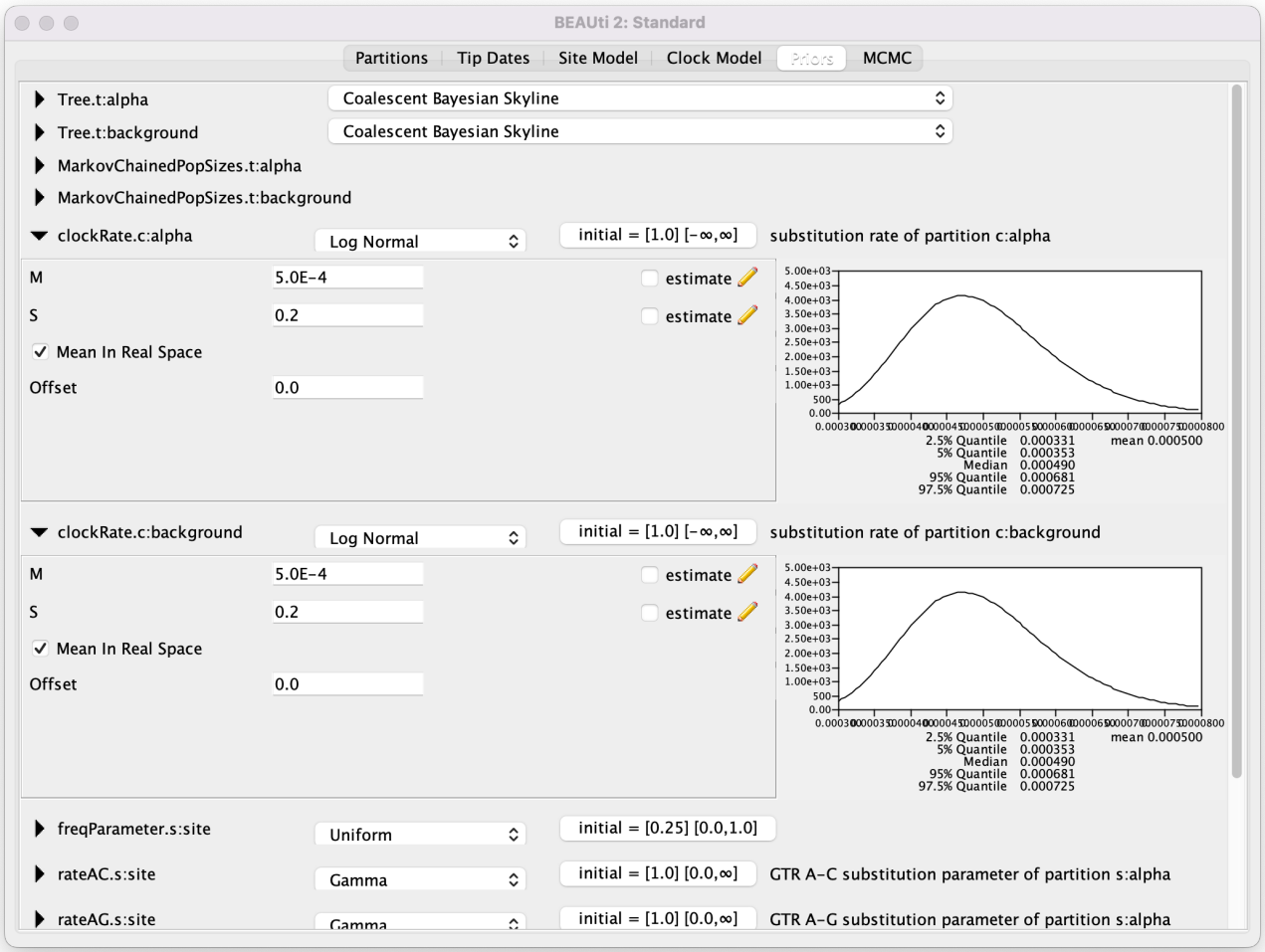

Figure 4: Prior setup.

For **clockRate.c:alpha** select **Log Normal** from the drop-down menu

- Expand the options for **clockRate.c:clock** using the arrow button on the left.
- Set the **M** parameter to **5E-4**.
- Set the **S** parameter to **0.2**
- Check the **Mean in Real Space** box

Now repeat the steps above for **clockRate.c:background**.

The BEAUti panel should look as shown in Figure 4.

Finally, we will set the MCMC settings and the output file names.

Go to the **MCMC** tab.

- Set the **Chain Length** to **10'000'000**.
- Expand the **tracelog** options, change the **File Name** to `$(filebase)_$(seed).log` and set the **Log Every** parameter to **1000**.
- Expand the **screenlog** options and set the **Log Every** parameter to **10'000**.
- Expand the **treelog.t:alpha** options, change the **File Name** to `$(filebase)_$(tree)` and set the **Log Every** parameter to **1000**.
- Expand the **treelog.t:alpha** options, change the **File Name** to `$(filebase)_$(tree)` and set the **Log Every** parameter to **1000**.
- Save the XML file under the name `bsp.xml` using **File > Save**.

When we run the analysis `$(filebase)` in the name of the `*.log` and `*.trees` files will be replaced by the name of the XML file, `$(tree)` by the tree parameter names and `$(seed)` by the random number seed. This is a good idea, since it makes it easy to keep track of which XML files and which random number seeds produced which output files.

When using the coalescent Bayesian Skyline plot it is very important that the **Log Every** parameter for the trace and tree log files are set to the same frequency, or else Tracer won't be able to reconstruct the skyline plot!

Now we are ready to run the analysis. Do **NOT** close BEAUti, as we will return to it in the following sections!

Run the **BEAST2** program.

- Select `bsp.xml` as the **Beast XML File**.
- Set the **Random number seed** to **777** (or pick your favourite number).
- Check the **Use BEAGLE library if available** checkbox. If you have previously installed BEAGLE this will make the analysis run faster.

Run **BEAST2** by clicking the `Run` button.

The analysis will take some time to complete (allow at least 15-20 minutes). Read through the next section while waiting for your results or start preparing the XML file for the birth-death skyline analysis.

3.3 The Coalescent Bayesian Skyline parameterization

The Kingman coalescent model that the Coalescent Bayesian Skyline is based on assumes that the sequences represent a small sample from a haploid population evolving under Wright-Fisher dynamics (Figure 5). The model works by calculating the probability of the tree under this assumption. This essentially boils down to repeatedly asking the question of how likely it is for two lineages to coalesce (have a common ancestor) in a given time.

The effective population size (N_e) is the inverse of the rate of coalescence λ. The larger N_e is, the less likely lineages are to coalesce. Thus, intervals in a sampled tree with many branching events often coincide

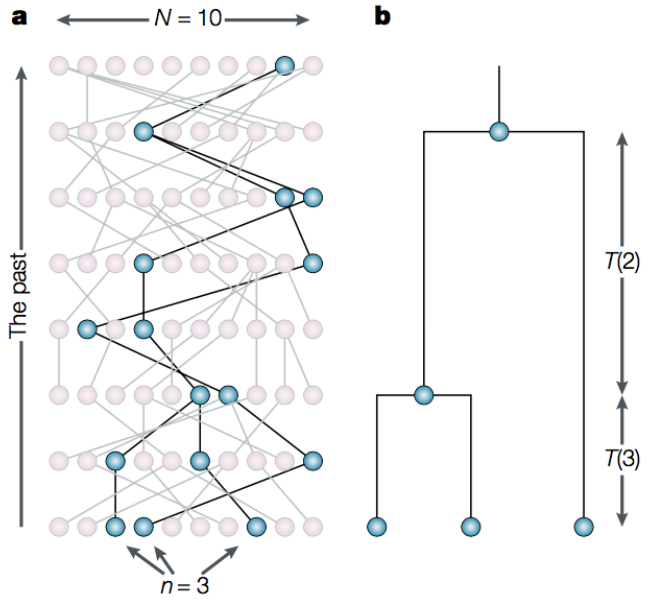

Figure 5: The basic principle behind the coalescent. Figure from (Rosenberg and Nordborg 2002).

with periods when the population size was small. Similarly, few branching events occur during periods of large population size. (Note that these results are conditioned on sampling only a small fraction of the population).

$$\lambda = \frac{1}{N_e} \tag{1}$$

For an SIR model (**S**usceptible, **I**nfected and **R**ecovered), N_e is proportional to the overall population size N and the number of infected I and inversely proportional to the transmission rate θ.

$$N_e = \frac{I}{\theta}\frac{N}{S} \tag{2}$$

Estimates of N_e therefore do not directly tell us something about the number of infected, nor the transmission rate. However, changes in N_e can be informative about changes in the transmission rate or the number of infected (if they do not cancel out).

The Coalescent Bayesian Skyline model allows N_e to change over time in a nonparametric fashion (i.e. we do not have to specify an equation governing changes in N_e over time). It divides the time between the present and the root of the tree (the tMRCA) into segments, and estimates a different effective population size (N_e) for each segment. The endpoints of segments are tied to the branching times (also called coalescent events) in the tree (Figure 6), and the size of segments is measured in the number of coalescent events included in each segment. The Coalescent Bayesian Skyline groups coalescent events into segments and jointly estimates the N_e (**bPopSizes** parameter in BEAST) and the size (**bGroupSizes** parameter) of each segment. To set the number of segments we have to change the dimension of **bPopSizes** and **bGroupSizes** (note that the dimensions of both parameters always have to be the same). Note that the

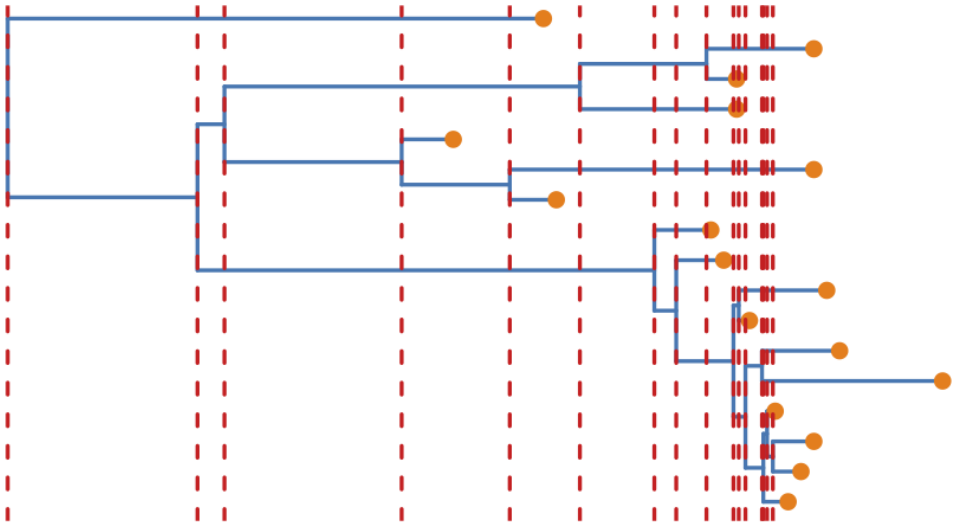

Figure 6: Example tree where the red dotted lines show the time-points of coalescent events.

length of a segment is not fixed, but dependent on the timing of coalescent events in the tree (Figure 6), as well as the number of events contained within a segment (**bGroupSizes**).

Another way to think about the model is as maximally-parameterized, since it infers d change-point times (segment boundaries) and a value for N_e in each segment. This makes the Bayesian Skyline flexible enough to model very complicated N_e dynamics, provided that enough segments are specified. It may be tempting to specify the maximum dimension for the model (each group contains only one coalescent event, thus N_e changes at each branching time in the tree), making it as flexible as possible. This is the parameterization used by the Classic Skyline plot (Pybus et al. 2000), which is the direct ancestor of the Coalescent Bayesian Skyline plot. However, the only informative events used by the Coalescent Bayesian Skyline plot are the coalescent events. Thus, using a maximally-flexible parameterization with only one informative event per segment often leads to erratic and noisy estimates of N_e over time (especially if segments are very short). Grouping segments together leads to smoother and more robust estimates.

Choosing the dimension for the Bayesian Skyline can be rather arbitrary. If the dimension is chosen too low, not all population size changes are captured, but if it is chosen too large, there may be too little information in a segment to support a robust estimate. When trying to decide if the dimension is appropriate it may be useful to consider the average number of informative (coalescent) events per segment. (A tree of n taxa has $n-1$ coalescences, thus N_e in each segment is estimated from on average $\frac{n-1}{d}$ informative data points). Would this number of random samples drawn from a hypothetical distribution allow you to accurately estimate the distribution? If not, consider decreasing the dimension. There are descendants of the coalescent skyline in BEAST that either estimate the number of segments (Extended Bayesian Skyline (Heled and Drummond 2008)) or do not require the number of segments to be specified (Skyride (Minin et al. 2008) or Skygrid (Gill et al. 2013) models), but instead makes very strong prior assumptions about smoothly how N_e changes through time.

3.4 Setting up a Birth Death Skyline analysis

We will now repeat the analysis using the Birth Death Skyline model. Either set up a new XML file in BEAUti and follow the same steps as above until you've specified the site and clock models or else simply go back to BEAUti and edit the previous analysis file.

We will need to set the prior to **Birth Death Skyline Serial**, since the sequences were sampled at different times. For homochronous data (all sequences sampled at the same time), we would use **Birth Death Skyline Contemporary**.

Select the **Priors** tab.

Select **Birth Death Skyline Serial** in the drop-down menus next to **Tree.t:alpha** and **Tree.t:background**.

Note that priors for two **reproductiveNumber**, **origin**, **becomeUninfectiousRate** and **samplingProportion** parameters have been added to the panel; one set for each tree. Besides R_e (**reproductiveNumber**), the **Birth Death Skyline Serial** model has 3 more parameters, **becomeUninfectiousRate** (the rate at which infected patients become uninfectious, δ, through recovery, death or isolation), **samplingProportion** (the proportion of removed lineages during a time period that are sampled) and the **origin** (the time at which the index case became infected, which is always earlier than the tMRCA of the tree). We may know some of these parameters from literature or be able to estimate them from external sources. For example, the average time that patients are able to transmit a disease is informative about the **becomeUninfectiousRate**. This prior knowledge we can incorporate in our analysis by setting appropriate priors for these parameters.

As with the Coalescent Bayesian Skyline, we need to set the number of dimensions. We will set the dimension of R_e, which denotes the average number of secondary infections caused by an infected person at a given time during the epidemic, i.e. an R_e of 2 would mean that every infected person causes two new infections on average. In other words, an R_e above 1 means that the number of cases are increasing, therefore the disease will cause an exponentially growing epidemic, and an R_e below 1 means that the epidemic will die out. We will also set the dimension of the **samplingProportion** parameters, so that we can also estimate changes in the sampling proportion over time.

In general, we need to fix one of the rates of the birth-death skyline model in order to make the model identifiable, although in some cases we can get away with strong priors on all rates. Here we will fix the **becomeUninfectiousRate** to 36.5 for both trees, which translates to a mean infected period of 10 days (the inverse of the rate), which we know to be a realistic estimate for SARS-CoV-2.

Navigate to the **Initialialization** panel.

- Set the dimensions of both **samplingProportion** parameters to 10.
- Double-check that both **reproductiveNumber** parameters have a dimension of 10.
- Uncheck the **Estimate** checkboxes after expanding the options for both **becomeUninfectiousRate** parameters. As soon as you uncheck the box they will disappear from the panel!

You could also change the dimension of parameters in the **Priors** tab. In the **Priors** tab, click on the

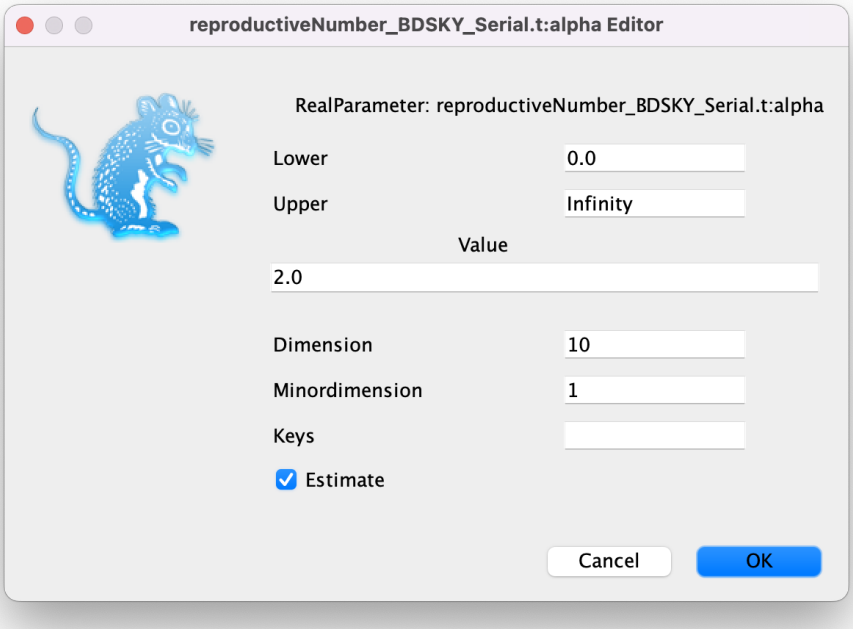

Figure 7: Setting the dimension of the reproductiveNumber parameter.

button where it says **initial = [2.0] [0.0, Infinity]** next to **reproductiveNumber_BDSKY_...**. A pop-up window will open which allows us to change the dimension of the parameter (Figure 7).

Now R_e and the sampling proportion will be allowed to change at 9 times equally spaced between the origin of the epidemic and the time of the most recent sample, in our case March 1st, 2021. Choosing this dimension can be arbitrary and may require the testing of a few different values. Too few intervals and not all rate shifts are captured. Too many intervals and the intervals may not contain enough information to infer parameters. We still need to set the **becomeUninfectiousRate** parameters to 36.5.

Return to the **Priors** tab.

- Expand the options for **Tree.t:alpha**.
- Enter **36.5** for **Become Uninfectious Rate** (Figure 8).

Now repeat the above steps for **Tree.t:background**.

We will use a lognormal prior for R_e. This is a good prior distribution to use for rates since it is always positive (a rate cannot be negative) and has a long tail defined over all positive numbers. The long tail allows arbitrarily high estimates of R_e, but does not place much weight on very high rates. This agrees with our prior knowledge about R_e (most diseases have an R_e between 1.2 and 8. Measles is one of the most infectious diseases we know of and has $R_e \approx 18$).

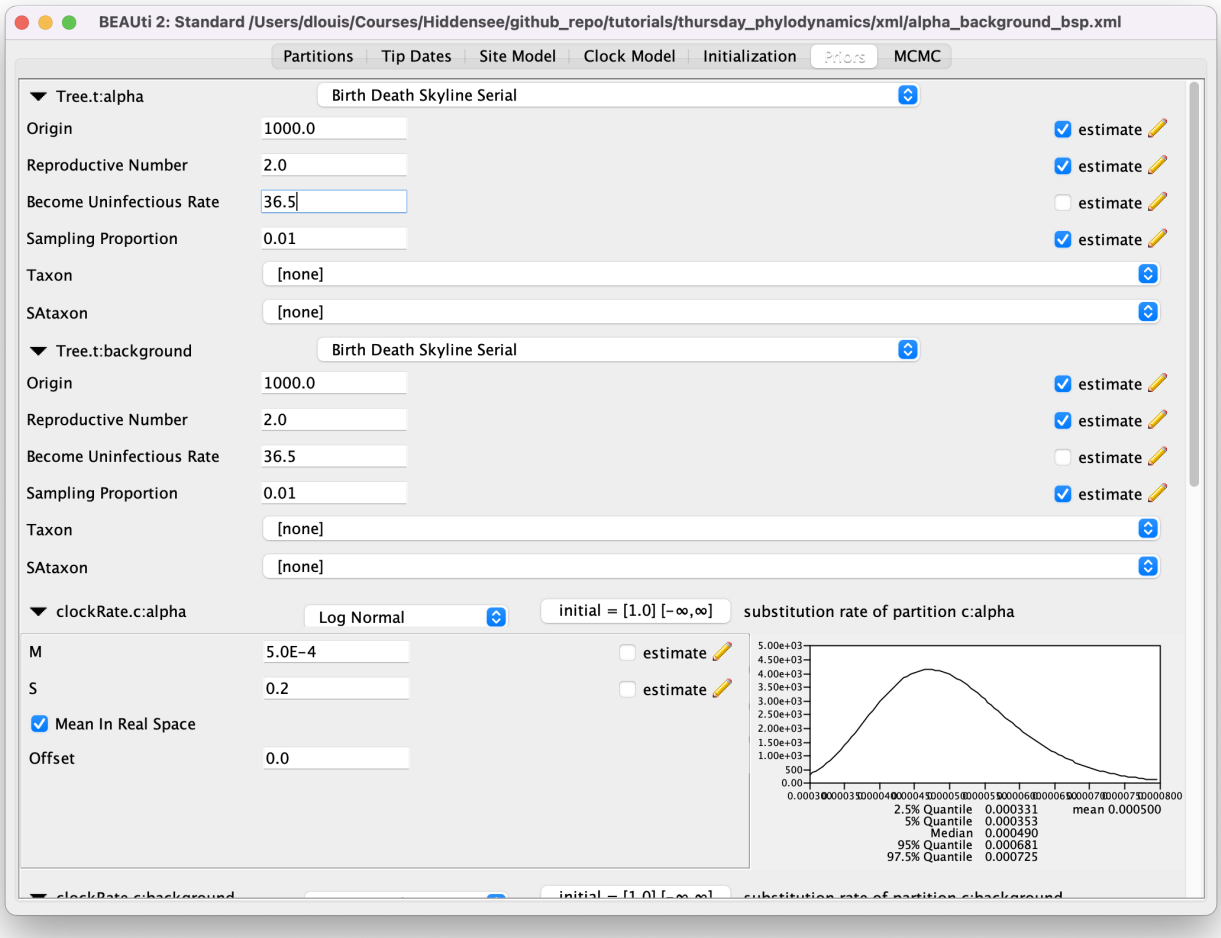

Figure 8: Fixing the become uninfectious rate.

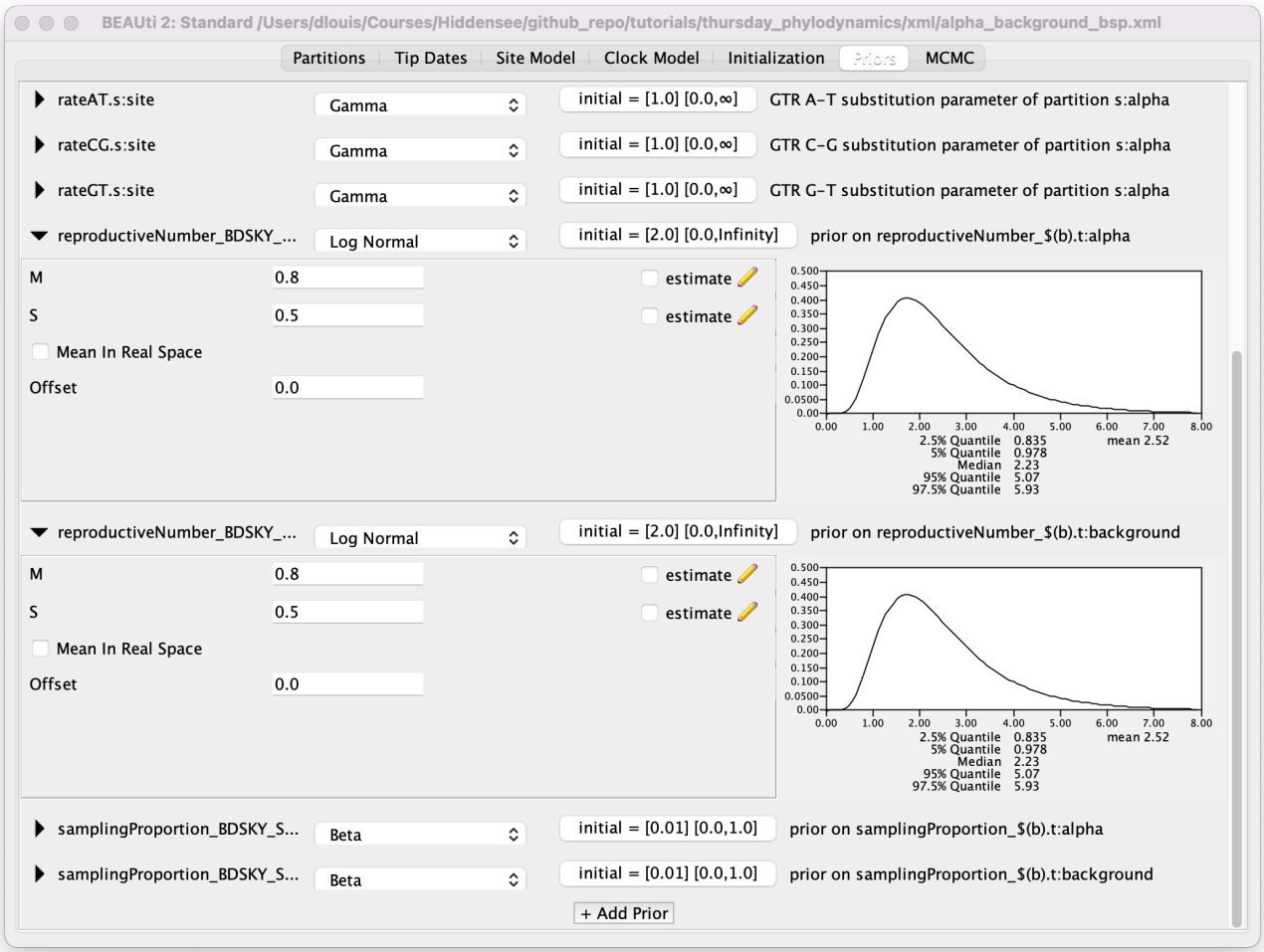

Figure 9: Setting the 'R_e' priors.

Select a **Log Normal** distribution for the first **reproductiveNumber_BDSKY_...** prior.

Click on the arrow to expand the options and set **M** to 0.8 and **S** to 0.5.

Now repeat the above steps for the second **reproductiveNumber_BDSKY_...** parameter (Figure 9.

This prior has a median above 1 and places most of the weight below 5. This agrees with what we know of the SARS-CoV-2 epidemic in the UK during the second half of 2020 and the start of 2021, when cases grew with the second and third waves of the epidemic. Note that this prior is used for each of the R_e intervals (the Birth-Death Skyline assumes that R_e is independent in each of the intervals).

The sampling proportion represents the proportion of removed lineages during each interval that were sequenced (included in the alignment). We will use a Beta distribution for the sampling proportion prior. Beta distributions are a very flexible class of distributions that are only defined between 0 and 1, making

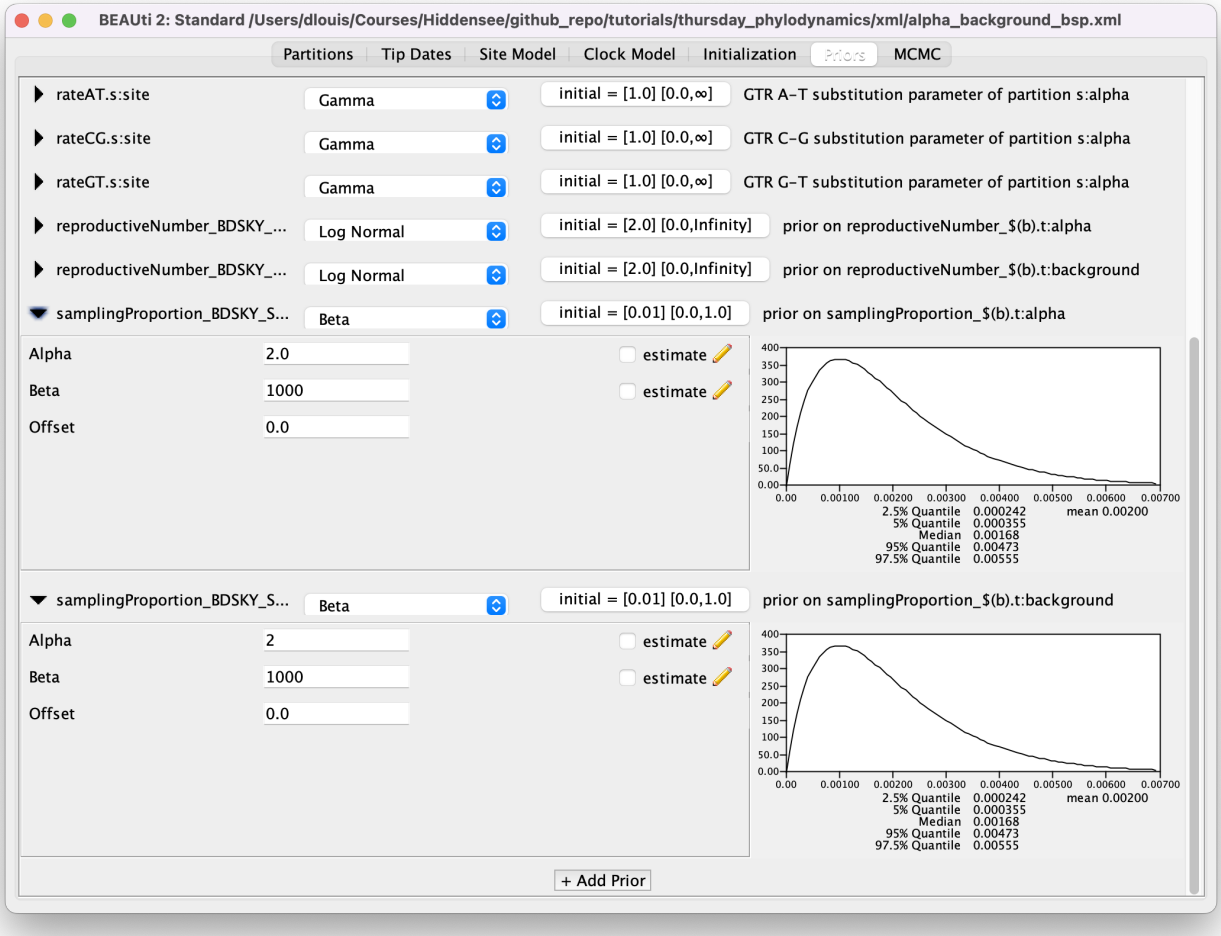

Figure 10: Setting the sampling proportion priors.

them well-suited to use for proportions. A Beta distribution can be interpreted as modelling the probability of success, out of a number of trials, which is ideal for a sampling model. To decide which parameter values to use for the Beta distribution you can think of $\alpha - 1$ as the number of successes (cases that were sequenced/observed) and $\beta - 1$ as the number of failures (cases that were not sequenced/observed).

Select a **Beta** distribution for the first **samplingProportion_BDSKY_S...** prior.

Click on the arrow expand the options and set **Alpha** to **2** and **Beta** to **1000** (Figure 10).

Now repeat the above steps for the second **samplingProportion_BDSKY_S...** prior.

Thus, the prior we set assumes we are sequencing roughly 1 out of every 1000 cases (the distribution has a median probability of sequencing a removed lineage of 0.000168).

Finally, we need to set a prior for the origin of the epidemic. We will once again use a log normal distribution

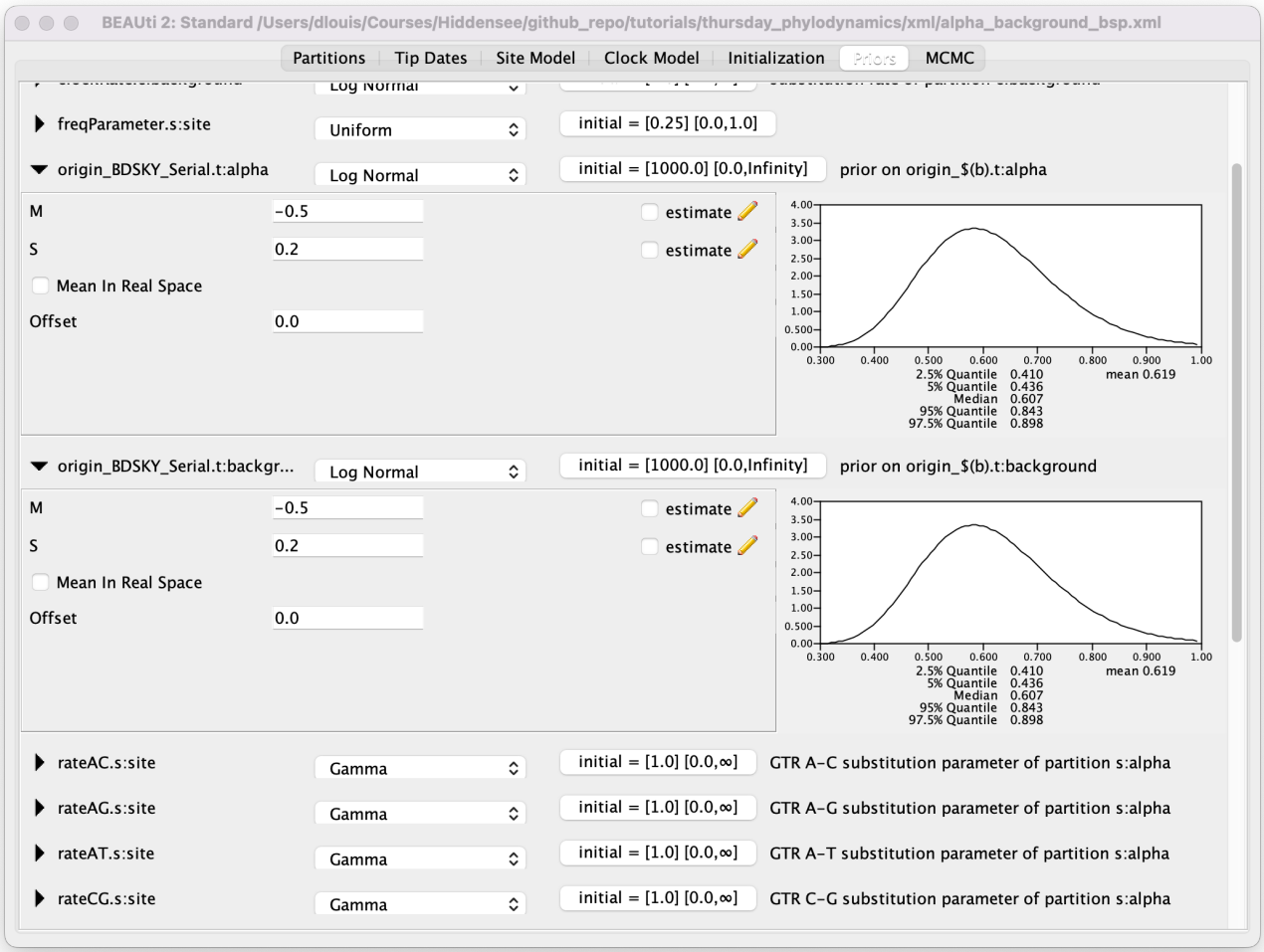

Figure 11: Setting the priors on the origins of the epidemics.

for this parameter. Note that the origin also has to be positive and needs to be bigger than the MRCA of the tree. At this point in the COVID-19 pandemic it had only been circulating for a little more than a year, thus we want to set a prior that reflects this knowledge. Keep in mind that the origin of the Alpha alignment would be the index case of the Alpha VOC and not of SARS-CoV-2! Similarly, the origin of the background alignment would be the index case that lead to all genomes represented in the dataset, which may not be the same as the index case of the entire pandemic.

Set a **Log Normal** prior for both **origin** parameters with **M = -0.5** and **S = 0.2** (Figure 11).

Double-check that the clock rate priors are the same as those set earlier for the coalescent Bayesian Skyline plot. The rest of the priors pertain to the site model parameters and we can leave them as they are.

Now save the file as `bdsky.xml` and run it in BEAST2. Since we used the `$(filebase)` placeholder in the file names we don't need to change them before saving!

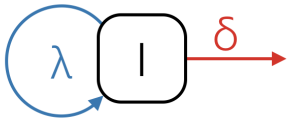

Figure 12: A schematic of a simple birth death model.

Read through the next section and set up the next XML file while waiting for the analysis to finish.

3.5 The Birth-Death Skyline parameterization

The birth-death model is parameterized very differently from the coalescent model, using per lineage rates and an explicit sampling model (whereas the coalescent model conditions on the samples). This makes the birth-death model more powerful, but also much more complex. A basic birth-death model has a birth rate (λ), the rate at which lineages are added to the tree, and a death rate (δ), the rate at which lineages are removed from the tree (Figure 12). In an infectious disease epidemic λ can be thought of as the transmission rate, the rate at which rate infected individuals infect susceptibles, while δ can be thought of as the becoming uninfectious rate, the rate at which infected individuals recover, die or are isolated. In species tree inferences these rates can be thought of in terms of speciation and extinction.

In addition, the model has an origin parameter. Whereas coalescent models work backward-in-time from the sampled sequences, birth-death models work forward-in-time from the origin. Hence, the model needs an origin time, which can also be jointly estimated along with the other parameters. The origin will always be bigger than the tMRCA of the sampled tree, since the sampled tree is by definition smaller than the complete tree.

The **Birth Death Skyline Serial** model we used was parameterized in terms of R_e and δ. Recall that $R_e > 1$ means that an epidemic will keep growing. We can see this from the definition of R_e as the ratio of the birth and death rates.

$$R_e = \frac{\lambda}{\delta} \tag{3}$$

We used this paramerization simply because it is often easier to specify priors for R_e than the transmission rate, and because R_e is often more informative for prevention efforts.

In addition, the model assumes that the data are heterochronous (sampled at different times). It assumes that:

$$\delta = \psi + \mu \tag{4}$$

where ψ is the rate at which lineages are sampled through time and μ is the rate at which lineages are removed from the tree for any other reason (death, recovery, extinction etc.). (In this case the ρ parameter is no-longer available, because samples are collected through time, and not just at one timepoint). By default, the model is parameterized in terms of R_e, δ and p, the sampling proportion:

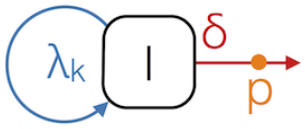

Figure 13: A schematic of the Birth Death Skyline Serial model.

$$p = \frac{\psi}{\psi + \mu} \tag{5}$$

The sampling proportion is the proportion of all removed lineages that were sampled, and can be used to obtain a rough estimate of the total population size. This model is useful for studying infectious disease dynamics, because samples are often collected over the course of an epidemic.

You can also see that the model **Birth Death Skyline Serial** assumes that upon sampling a lineage is removed from the tree (e.g. in a disease model the sampled individual cannot transmit the disease after sampling). The consequence for the phylogeny is that a sampled lineage cannot be a direct ancestor of any other lineage in the tree. This assumption can be relaxed, but we will not do so during this tutorial.

You may have noticed that there are many Birth-Death Skyline models available in BEAUti. For example, the **Birth Death Skyline Contemporary** model is used for homochronous data (all sequences sampled at the same time) and parameterized in terms of λ, δ and ρ, whre ρ is the sampling probability at the present time.

The Birth-Death Skyline model is very flexible and allows any or all of these rates to change independently over time. This is done by dividing the time from the origin to the most recent sample into dimension d equally spaced intervals (see Figure 14). The rates are then allowed to change between intervals. Since some rates (e.g. λ and δ) are highly correlated, it is not always a good idea to let all rates change over time because it can lead to poor mixing or biased estimates. It is also possible to specify the change-point times more flexibly, or even estimate them, however for now this requires editing the XML file. Some examples are available here.

3.6 Analyzing the Coalescent Bayesian Skyline results

Once BEAST2 has finished running the coalescent Bayesian Skyline analysis, open Tracer to get an overview of the BEAST2 output. When the main window has opened, choose `File` > `Import Trace File...` and select the file called `bsp_777.log` that BEAST2 has created, or simply drag the file from the file manager window into Tracer.

> Open **Tracer**. Drag and drop the `bsp_777.log` file into the open Tracer window.
>
> Alternatively, use **File > Import Trace File...** (or press the + button below the **Trace Files** panel) then locate and click on `bsp_777.log`.

The Tracer window should look as shown in Figure 15. Because we only ran a short chain for a complicated analysis many parameters have very low ESS values. You can find a log file of an identical analysis that was run for 100 million MCMC steps in the `precooked_runs/` folder (`bsp_long.log`).

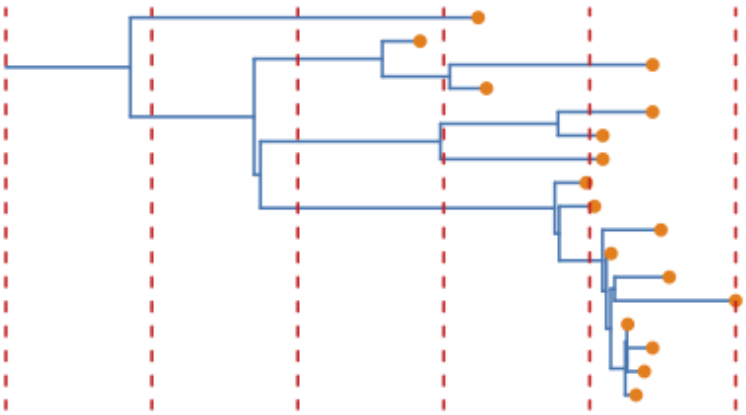

Figure 14: Example tree where the red dotted lines are an example of where rates could be allowed to change on the tree. The branch at the root (compare Figure 6) is indicating the origin of the epidemic, which is also estimated in the BDSKY.

Note that many parameters have either `.alpha` or `.background` appended to their names. These are parameters that we inferred independently for the Alpha and background alignments/partitions. The `popSizes` parameters represent the effective population sizes for each alignment in each interval, whereas the `groupSizes` parameters are the sizes of each interval (more specifically, the number of coalescent events within an interval). Since populations naturally grow/decline through exponential growth/decay, it is more natural to visualise the population sizes on a log scale.

Select the **Marginal Density** tab.

- Select all of the **popSizes.alpha** parameters using **shift + click**.
- Open the **Display** drop-down menu and select **Violin**.
- Click on the setup button at the bottom and check **Log axis**.

We see that the effective population size of Alpha appears to have grown linearly on a log scale, which translates to exponential growth. Similarly, the background appears to have first grown exponentially, and then stabilised at a constant population size. On the other hand, the group sizes parameters were not inferred very precisely and there is a large variation in their sizes (Figure **??**). This indicates smooth population size changes.

However, these are just indications, and we need to combine both parameters to reconstruct the population dynamics. To do that Tracer needs to know when the coalescent events on each posterior tree was, and therefore also needs the `*.trees` file. This is the reason why it's important to log the trace and tree files at the same frequency for the coalescent Bayesian Skyline plot. But before we can do that, we need to know

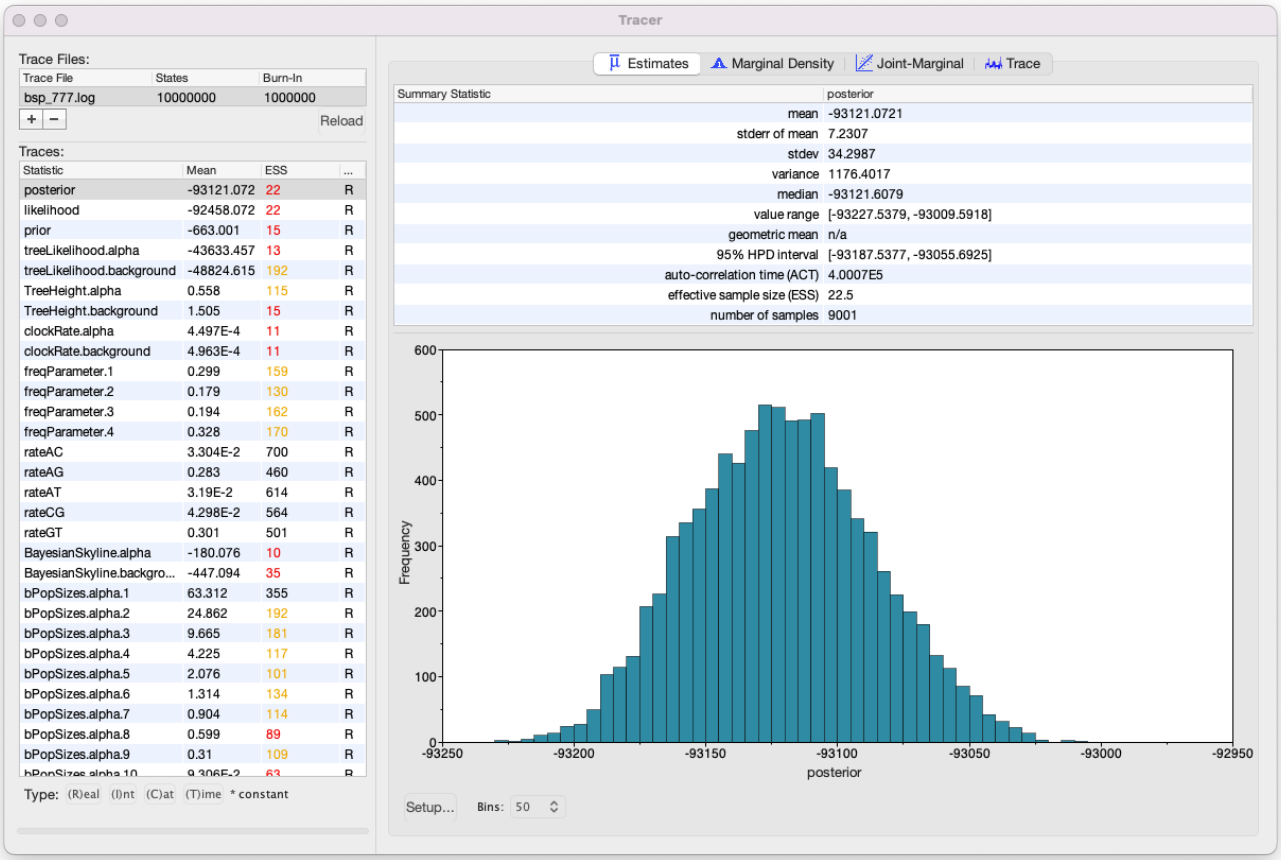

Figure 15: Tracer showing a summary of the BEAST2 run of the coalescent Bayesian Skyline plot with an MCMC chain length of 10'000'000 and no constraints.

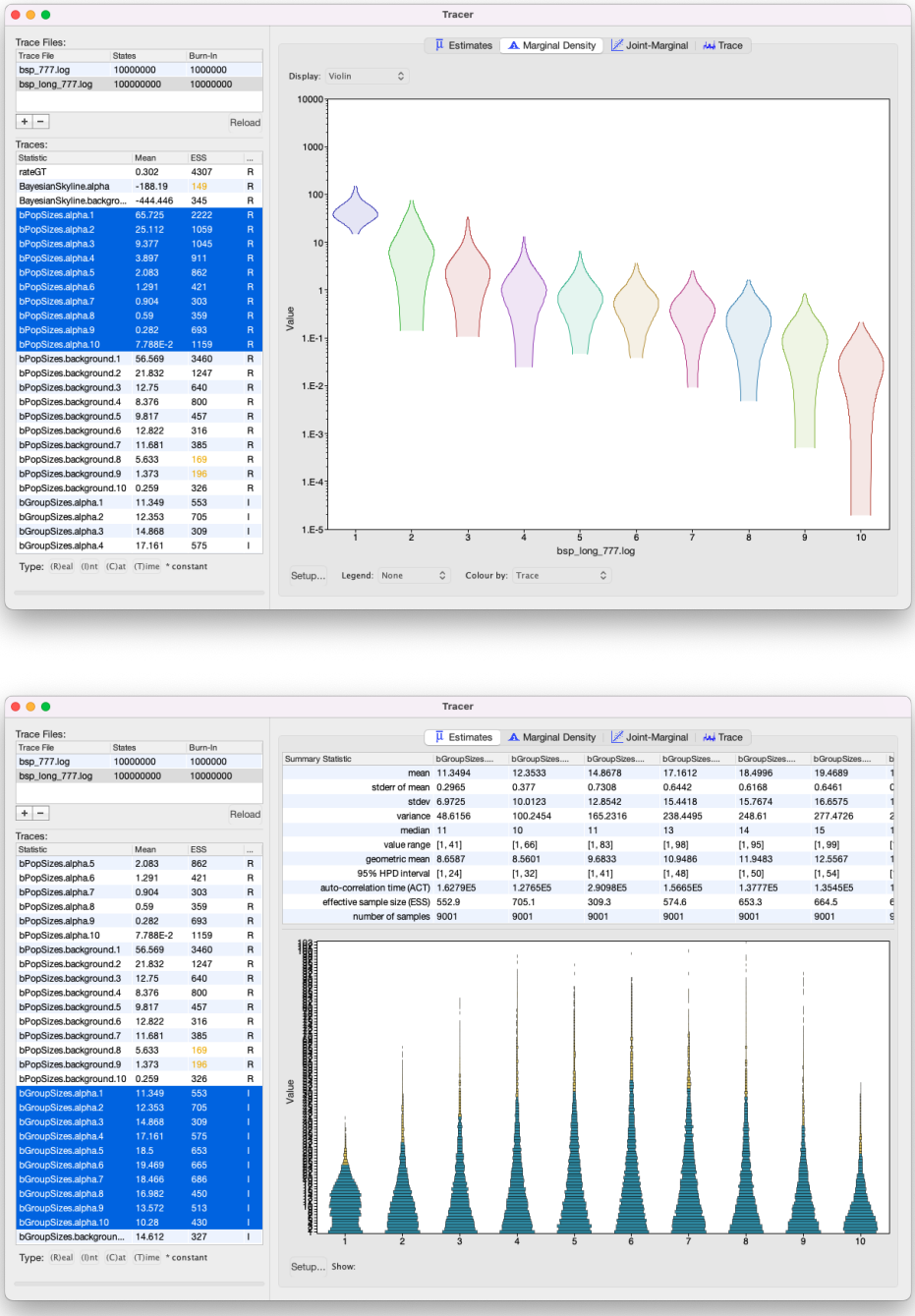

Figure 16: Visualising the population sizes and group sizes in Tracer.

the decimal date at the time of the most recent sample in our dataset (March 1st, 2021).

Open R or Rstudio.

Type in:

```
library(lubridate)
lubridate::decimal_date(ymd("2021-03-01"))
```

This should give:

```
[1] 2021.162
```

Return to Tracer and navigate to **Analysis > Bayesian Skyline Reconstruction**.

- Choose `bsp_alpha.trees` next to **Trees Log File**.
- Set **Population Size** to **bPopSizes.alpha** and **Group Size** to **bGroupSizes.alpha**.
- Set the **trace of the root height** to **TreeHeight.alpha**.
- Check the **Use manual range for bins** checkbox and enter **2020** and **2021.162** for the minimum and maximum times.
- Enter **2021.162** for the **Age of youngest tip**.
- Press **OK** to reconstruct the past population dynamics (Figure 17).

Without closing the resulting plot, go back to the main Tracer window and repeat the same steps for the background alignment.

This should reconstruct the population dynamics for both alignments across the same time period (Figure 18). In general it is not necessary to use a manual range for the bins, but we used it here to ensure that both plots are over the same X-axis range and more easily comparable. There are two ways to save the analysis, it can either be saved as a `*.pdf` for display purposes or as a tab delimited file, which can be loaded in R and used for custom plots (Figure 19).

Navigate to **File > Export Data Table**.

Enter the filename as `hcv_coal.tsv` and save the file.

3.7 Analyzing the Birth Death Skyline results (i)

If we open the trace file for the Birth Death Skyline analysis we notice that the ESS values are even lower. The trace for the origin of the Alpha alignment is especially bad and highly autocorrelated (Figure 20). Recall that the samples in the trace should approximate uncorrelated draws from the target distribution, which should result in the traces resembling white noise. Obviously we would have to run this analysis much, much, much longer for this parameter to begin mixing well. Even looking at the results from an analysis that was run for 100 million steps, the parameter still has an ESS below 100. Another concern is the origin of the background alignment. Although it mixes well, it is estimated to be greater than 5

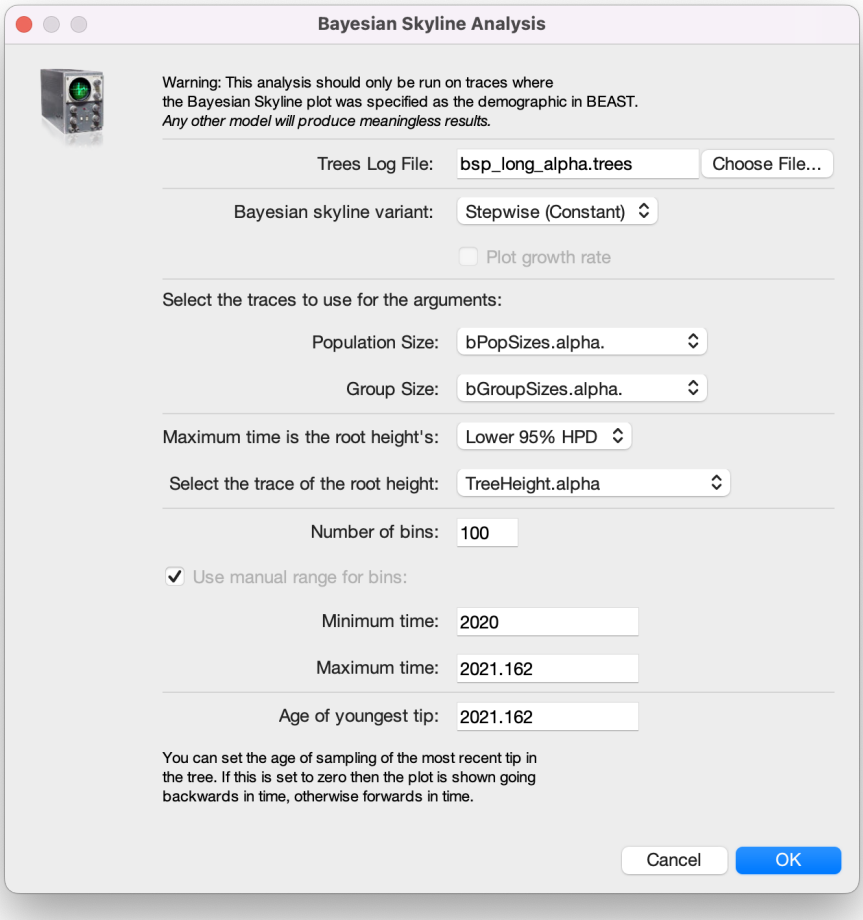

Figure 17: Reconstructing the Bayesian Skyline plot in Tracer.

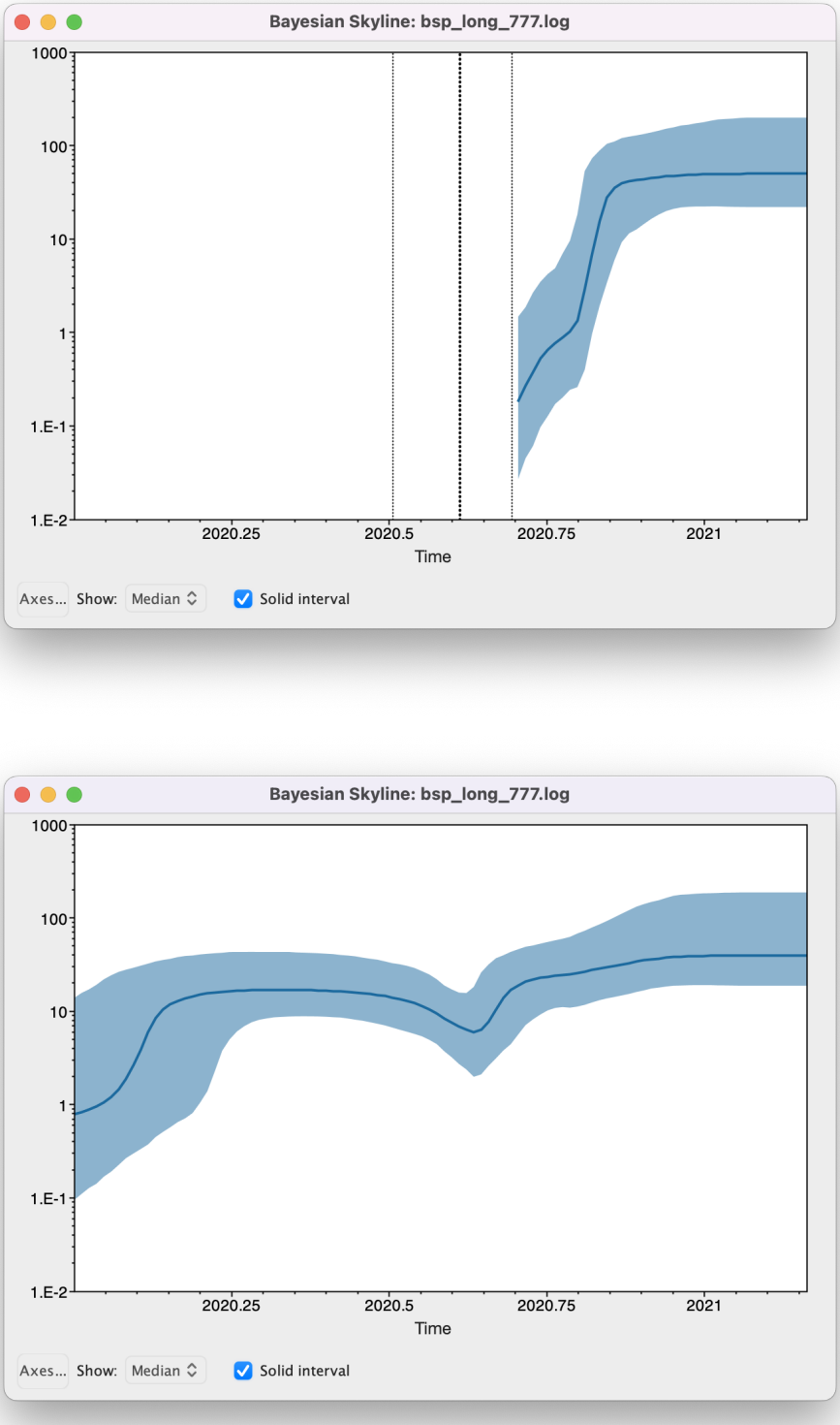

Figure 18: Coalescent Bayesian Skyline analysis output. The thick line is the median estimate of the estimated effective population size (can be changed to the mean estimate). The solid interval represents the upper and lower bounds of the 95% HPD interval. The x-axis is the time in years and the y-axis is on a log-scale.

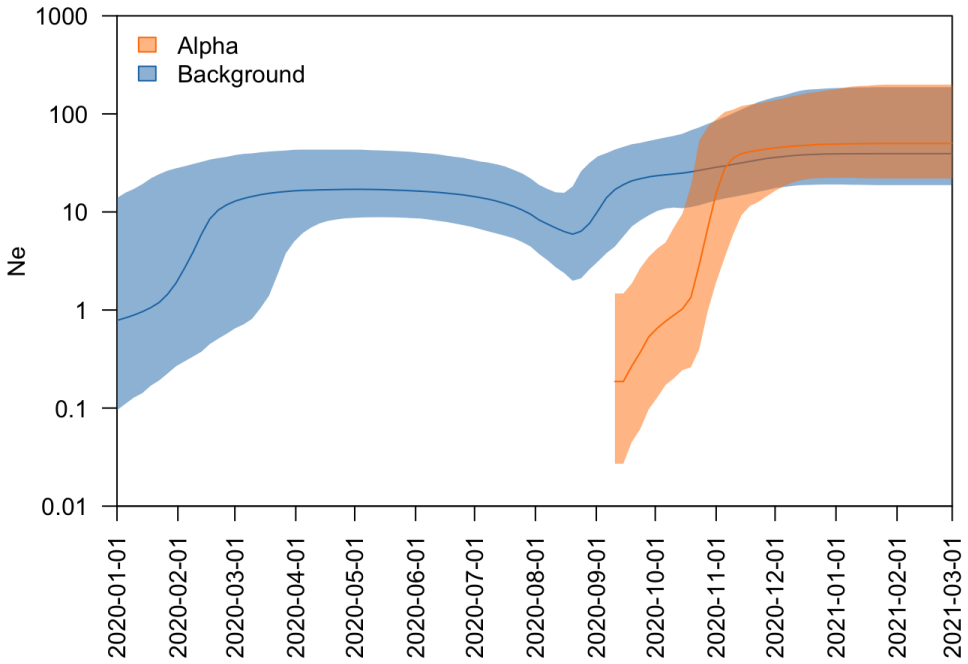

Figure 19: Both Bayesian Skyline Plots on one set of axes.

years! The implication would be that the index case of SARS-CoV-2 existed some time in 2015! This seems counter to everything we know about the pandemic. Not even the most outrageous conspiracy theories hypothesize that the virus already circulated in 2015. Even if we choose to believe these results, this analysis wouldn't tell us much about the population dynamics of the background dataset during the COVID-19 pandemic. Because the period from the origin to the most recent sample is divided into 10 equidistant intervals, all of the dynamics during the pandemic are encapsulated within the last 3 intervals and all other intervals simply recover the prior expectation we set on R_e.

3.8 Conditioning the Birth Death Skyline on the root

The Birth Death Skyline model occasionally has difficulty inferring the time of the origin of an epidemic. This parameter represents the index case and may have existed long before the oldest sample in the dataset was collected and sometimes also long before the tMRCA of all sequences in the dataset. If the epidemic has a constant growth rate over time it is straighforward to infer the time of the origin. However, the Birth Death Skyline explicitly allows the growth rate to change over time in a nonparametric fashion. The end result is that there may not be enough information in the dataset to infer both changes in the growth rate over time as well as the time of the origin. We can circumvent this limitation by eliminating the origin parameter and "conditioning on the root." In this alternative parameterisation of the Birth Death Skyline model, the model is parameterised to start not at the index case, but at the time when there are already two lineages in the sampled tree, i.e. the tMRCA of the tree. In this case, the intervals are spaced equidistantly between the tMRCA and the most recent sample.

Select the **Priors** tab.

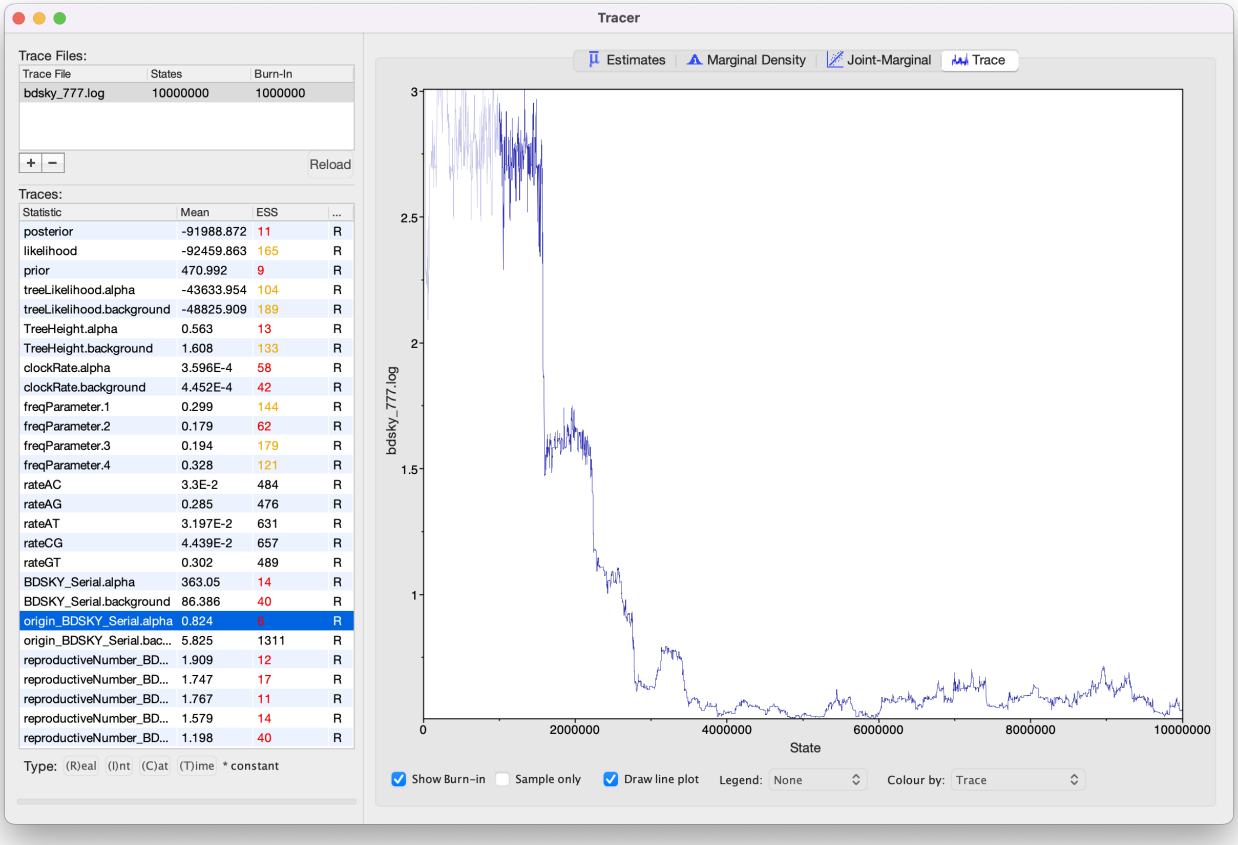

Figure 20: Trace of the origin of the Alpha alignment after 10'000'000 MCMC steps.

Select **Birth Death Skyline Serial Cond Root** in the drop-down menus next to **Tree.t:alpha** and **Tree.t:background**.

Now repeat the steps for the **Birth Death Skyline Serial** model to set the reproductive number and sampling proportion priors, change the dimension of the sampling proportions and fix the become uninfectious rates (with this model there is no origin parameters and therefore no origin priors).

Now save the file as `bdsky_condOnRoot.xml` and run it in BEAST2.

3.9 Sampling from the prior

It is always a good idea to sample from the prior. Ideally this should be done before running the analysis, as sampling from the prior can point out issues with the model. In particular, it allows us to see how much information is encoded in the priors and model structure. For example, suppose our aim is to decide if non-pharmeceutical interventions (NPIs) were effective by looking at whether R_e dropped below one after they were implemented. If the model is constrained such that R_e is always below one after that time, the model is worthless for answering our question. We can't simply look at the prior distributions we set in BEAUti to reach such insights, because priors often interact with each other in complex ways Heled and Drummond 2011. The result is that the induced prior of the complete model may not be the same as the priors we set in BEAUti.

Select the **MCMC** tab.

- Set the **Chain Length** to **50'000'000**.
- Expand the **tracelog** options and set the **Log Every** parameter to **5000**.
- Expand the **treelog.t:alpha** options and set the **Log Every** parameter to **5000**.
- Expand the **treelog.t:alpha** options and set the **Log Every** parameter to **5000**.
- Check the **Sample From Prior** checkbox at the bottom of the window (Figure 21.

Now save the file as `bdsky_condOnRoot_sampleFromPrior.xml` and run it in BEAST2.

Sampling from the prior runs the analysis without using the data. In BEAST2 this means simply masking the alignment and not calculating the tree likelihood. Since calculating the tree likelihood is usually the most time-consuming part of any analysis, sampling from the prior is usually much faster than running the analysis. However, since there is no data, it also often has difficulty mixing and needs to be run for longer.

Note that in the case of a heterochronous dataset, as we have here, sampling from the prior still uses the sampling (tip) dates in the analysis. Strictly speaking, these can also be considered part of the data. However, in our case we are also interested in seeing precisely how much information the sequences are adding on top of the collection dates.

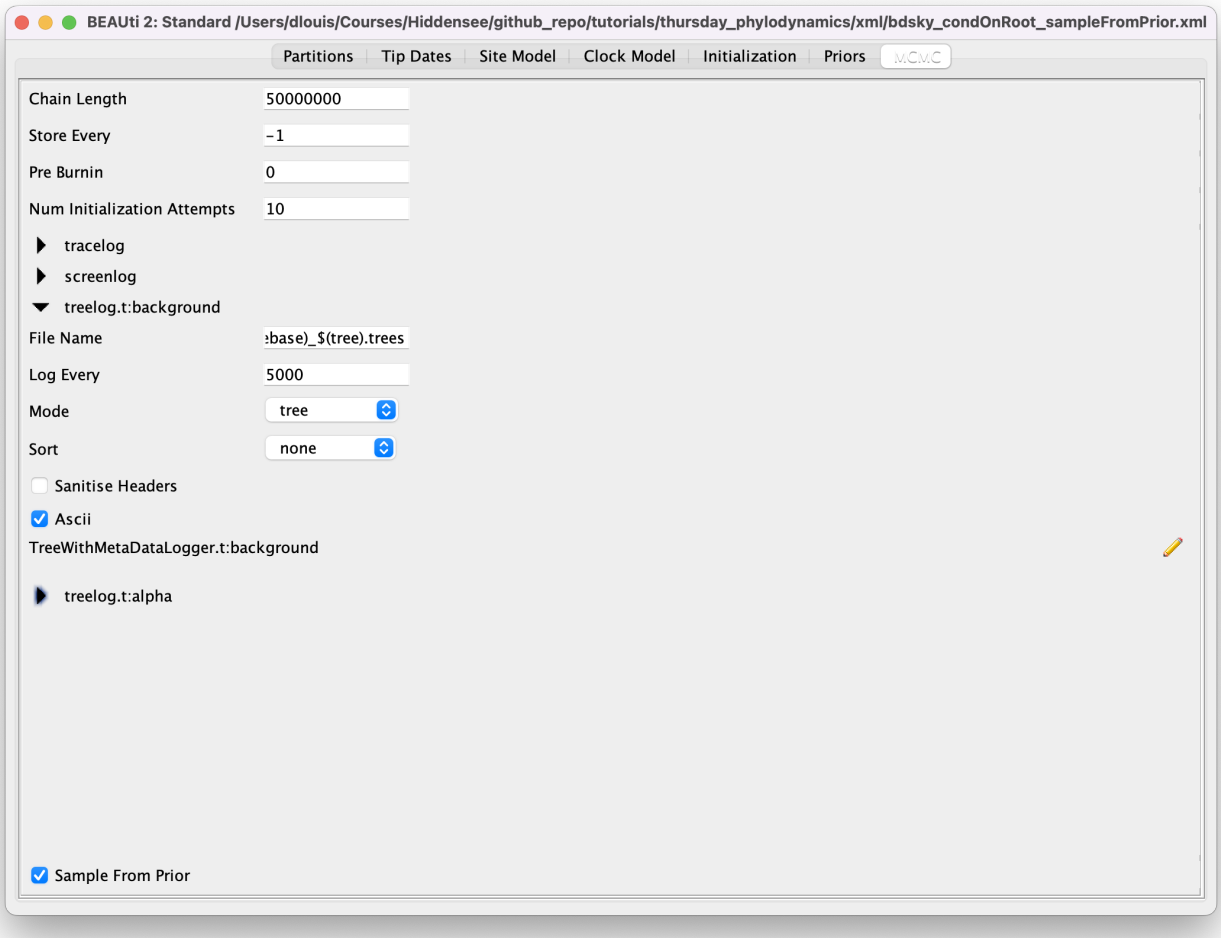

Figure 21: Sampling from the prior.

3.10 Analyzing the Birth Death Skyline results (ii)

The trace file of the Birth Death Skyline analysis conditioning on the root should have better ESS values than before, although we still wouldn't expect it to mix well after only 10 million steps.

Open **Tracer**. Drag and drop `bdsky_condOnRoot_777.log` and `bdsky_condOnRoot_sampleFromPrior_777.log` into the open Tracer window.

3.10.1 Visualising the Birth Death Skyline

There is no equivalent visualization of the skyline plot of a Birth-Death Skyline (BDSKY) analysis in Tracer as there is for the Coalescent Bayesian Skyline. But because BDSKY separates the full tree into equally spaced intervals, we can already get an idea of the inference just by looking at the inferred R_e values (see Figure **??**). This gives us a good idea of the trend, but it is not completely accurate. Since we are also estimating the tMRCA, the interval times are slightly different in each posterior sample and overlap slightly. The advantage of this is that we get a smooth estimate through time. The disadvantage is that we need to do some extra post-processing to plot the smooth skyline. We will use R to post-process and plot the Birth Death Skyline. The below steps are also in an RMarkDown notebook in the `scripts/` directory.

First, install the necessary packages. Once installed, we don't need to install the packages again.

```
install.packages("devtools")
install.packages("lubridate")
install.packages("coda")
install.packages("RColorBrewer")
devtools::install_github("laduplessis/bdskytools")
devtools::install_github("laduplessis/beastio")
```

Once installed, we have to load the packages into our R workspace before we can use the functions in the packages.

```
library(lubridate)
library(coda)
library(bdskytools)
library(beastio)
library(RColorBrewer)
```

To plot the results, we need to first tell R where to find the `*.log` file of our run and then load it into R (discarding 10% of samples as burn-in). If you are using RStudio, you can change the working directory to the directory where you stored your log files, which makes it easier to load the files in R.

```
bdsky_trace   <- beastio::readLog(bdsky_condOnRoot_777.log, burnin=0.1)
```

With the trace loaded as an `mcmc` object from the coda package we can use coda functions to investigate the trace and check convergence (Figure 23).

Next we can extract the R_e parameter values for the Alpha alignment and their HPDs.

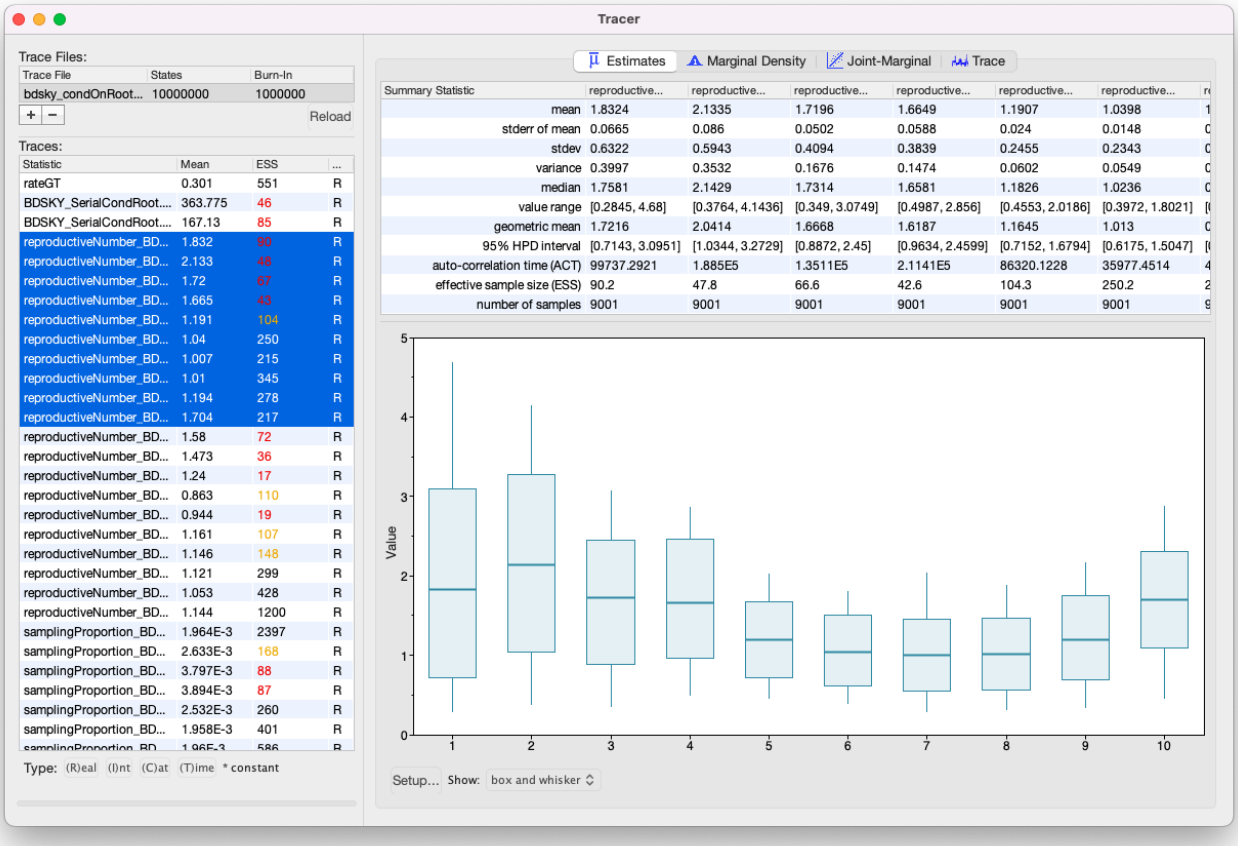

Figure 22: Estimated population dynamics by BDSKY in Tracer.

Figure 23: The ESS values of the trace.

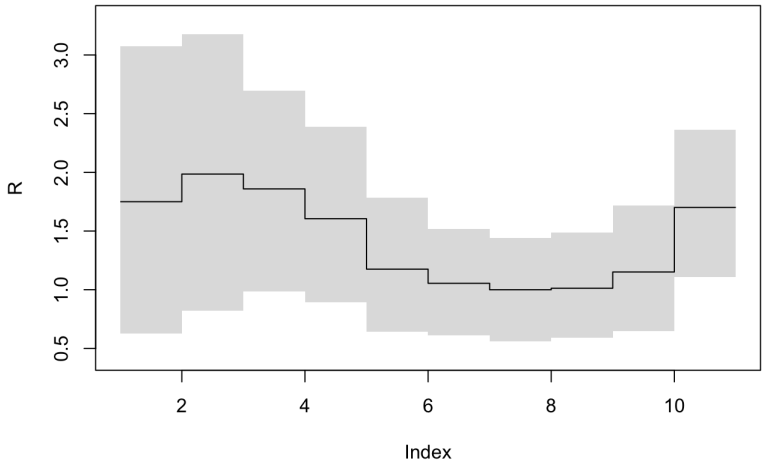

Figure 24: The HPDs of 'R_e' (equivalent to the previous figure from Tracer).

```
Re_alpha <- beastio::getLogFileSubset(bdsky_trace, "reproductiveNumber_BDSKY_SerialCondRoot.alpha")
Re_alpha_hpd <- t(beastio::getHPDMedian(Re_alpha))
```

We can plot the raw R_e HPD intervals. This is equivalent to the output in Tracer (Figure 24).

```
plotSkyline(1:10, Re_alpha_hpd, type='step', ylab="R")
```

In order to plot the smooth skyline we have to marginalise our R_e estimates on a regular timegrid and calculate the HPD at each gridpoint. It is usually a good idea to use a grid with more cells than the dimension of R_e. To do this we first calculate the marginal posterior at every time of interest using the function `gridSkyline()` and then calculate the HPD for each of the finer time intervals. The times to grid the skyline on (`gridTimes_alpha`), refers to years in the past. Since we conditioned on the root we have to use the tMRCA (**TreeHeight**) as an anchor point. If we didn't condition on the root we would have to use the **origin** parameter.

```
tmrca_alpha      <- bdsky_trace[, "TreeHeight.alpha"]
gridTimes_alpha  <- seq(0, median(tmrca_alpha), length.out=params$gridsize)

Re_alpha_gridded <- mcmc(bdskytools::gridSkyline(Re_alpha, tmrca_alpha, gridTimes_alpha))
Re_alpha_gridded_hpd <- t(getHPDMedian(Re_alpha_gridded))
```

Now we are ready to plot the smooth skyline (Figure 25).

```
times <- lubridate::decimal_date(enddate)-gridTimes_alpha
plotSkyline(times, Re_alpha_gridded_hpd, xlab="Date", ylab="Re")
```

We can plot the gridded R_e skyline (not its HPDs) for a few of the MCMC samples to see what it really

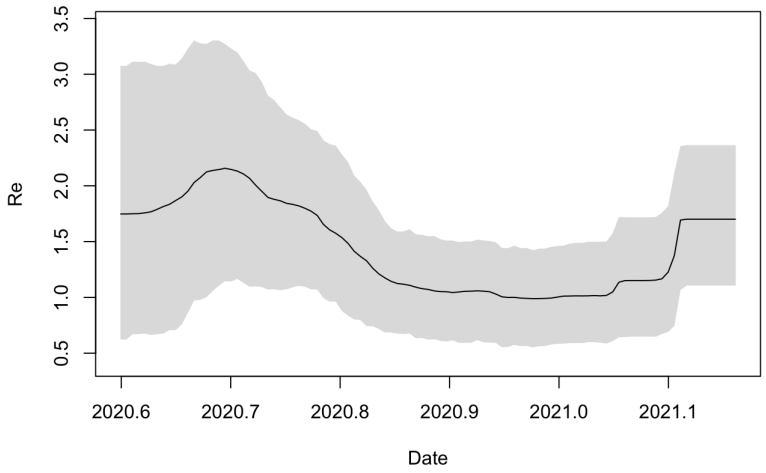

Figure 25: The smooth R_e skyline.

looks like as the Markov chain samples parameters. Note that the intervals overlap between different posterior samples. This is because the tMRCA is different in each of the plotted samples. As we add more samples to the plot we start to see the smooth skyline appear (Figure 26).

```
plotSkyline(times, Re_alpha_gridded, type='steplines', traces=1,
            col=cols$blue,ylims=c(0,3.5), xlab="Time", ylab="R", main="1 random sample")
plotSkyline(times, Re_alpha_gridded, type='steplines', traces=10,
                col=set_alpha(cols$blue,0.5),ylims=c(0,3.5), xlab="Time", ylab="R", main="10 ↩
                    random samples")
plotSkyline(times, Re_alpha_gridded, type='steplines', traces=100,
                col=set_alpha(cols$blue,0.5),ylims=c(0,3.5), xlab="Time", ylab="R", main="100 ↩
                    random samples")
plotSkyline(times, Re_alpha_gridded, type='steplines', traces=1000,
                col=set_alpha(cols$blue,0.1),ylims=c(0,3.5), xlab="Time", ylab="R", main="1000 ↩
                    random samples")
```

We can do the same for the sampling proportion estimates for the Alpha alignment (Figure 27).

```
samplingProp_alpha <- beastio::getLogFileSubset(bdsky_trace, "samplingProportion_BDSKY_↩
    SerialCondRoot.alpha")
samplingProp_alpha_gridded <- mcmc(bdskytools::gridSkyline(samplingProp_alpha, tmrca_alpha, ↩
    gridTimes_alpha))
samplingProp_alpha_gridded_hpd <- t(getHPDMedian(samplingProp_alpha_gridded))
plotSkyline(lubridate::decimal_date(enddate)-gridTimes_alpha, samplingProp_alpha_gridded_hpd,
            xlab="Date", ylab="Sampling proportion")
```

Now we can follow the same steps to also extract and plot the skylines for the background alignment. Finally, we can plot both Alpha and the background datasets on one set of axes for comparison (Figure 28).

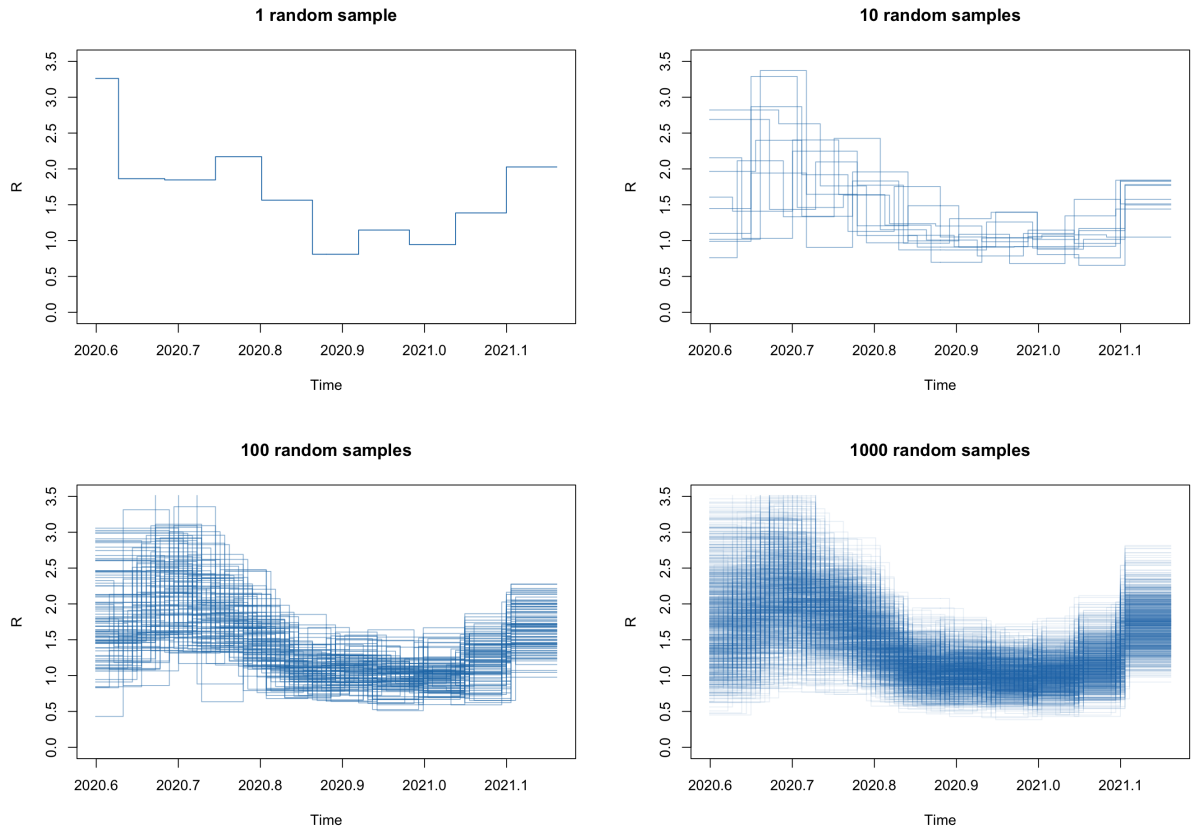

Figure 26: Increasing the number of traces plotted from 1 to 10, to 100, to 1000.

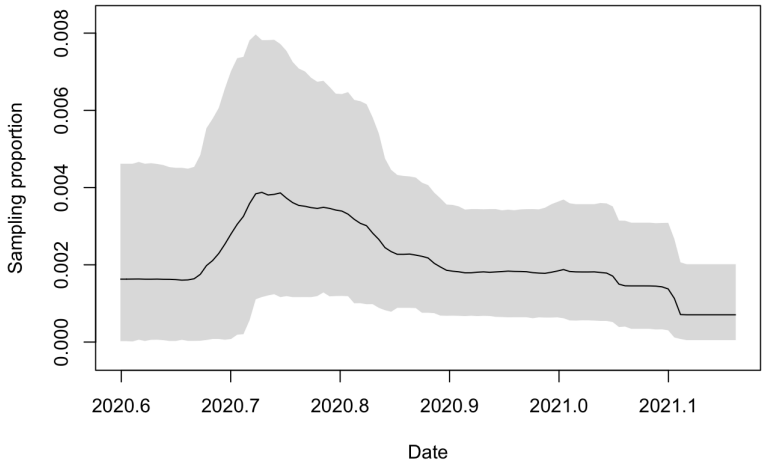

Figure 27: The smooth sampling proportion skyline.

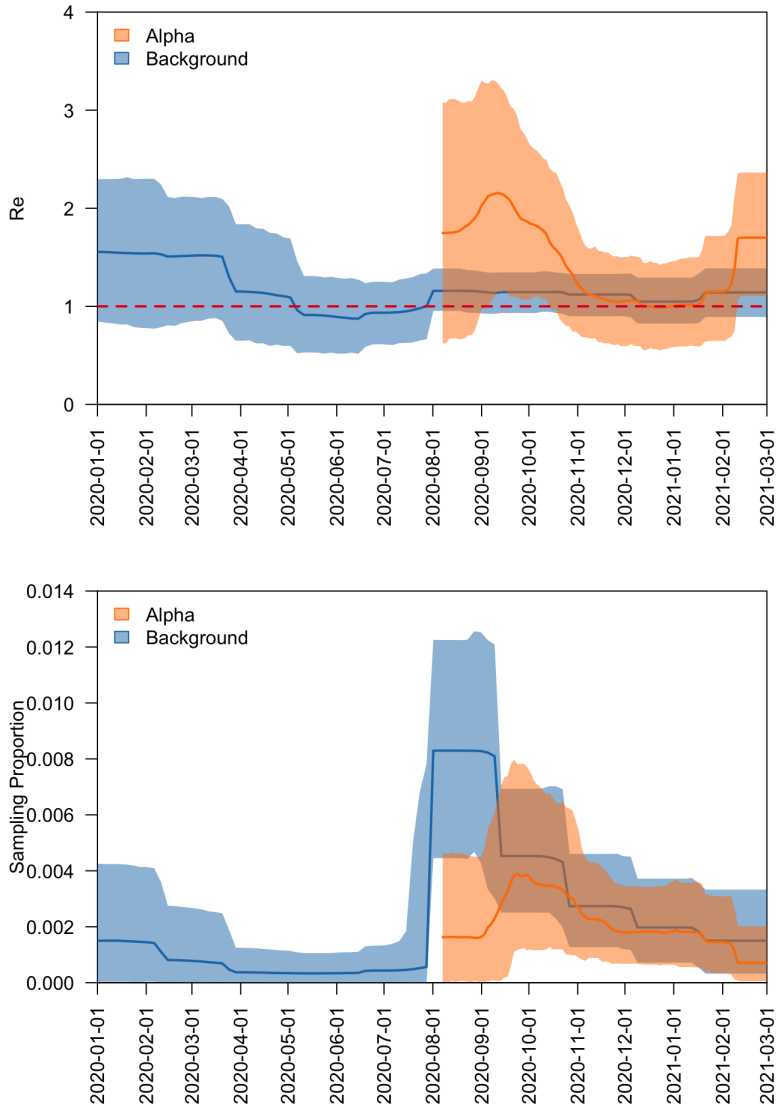

Figure 28: Plotting the skylines of the Alpha and background datasets on one set of axes.

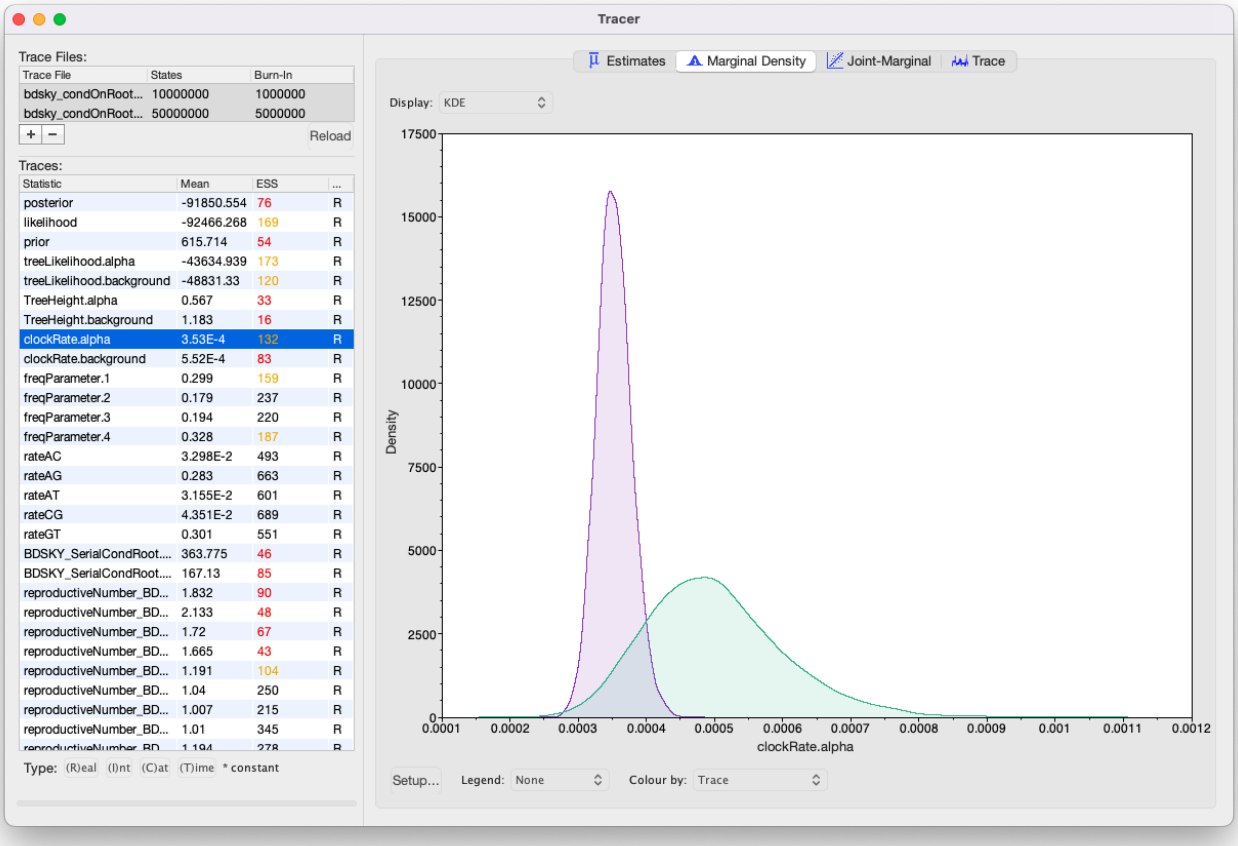

Figure 29: Comparing prior and posterior distributions in Tracer.

3.10.2 Comparing to the prior

When comparing the posterior and prior estimates in Tracer we can see that for some parameters, such as the clock rates, the posterior is a much tighter and more peaked distribution than the prior (Figure 29). But for the Skyline parameters, the picture is much less clear. To get a better idea we have to plot the smooth skylines in R.

We can follow the same steps as above to load the log file from `bdsky_condOnRoot_samplingFromPrior_777.xml` into R, grid and plot the skylines, and then compare them to the posterior estimates (Figure 30).

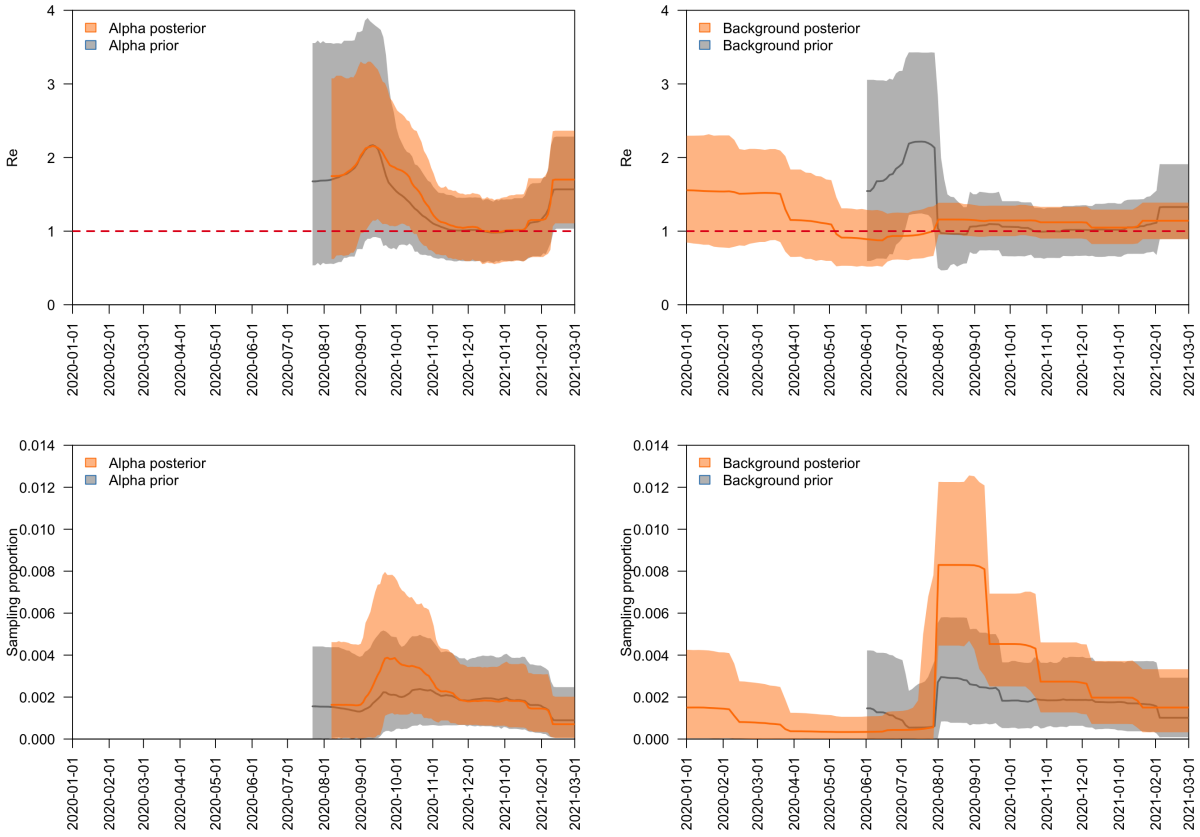

Figure 30: Comparing prior and posterior distributions of the Birth Death Skyline parameters.

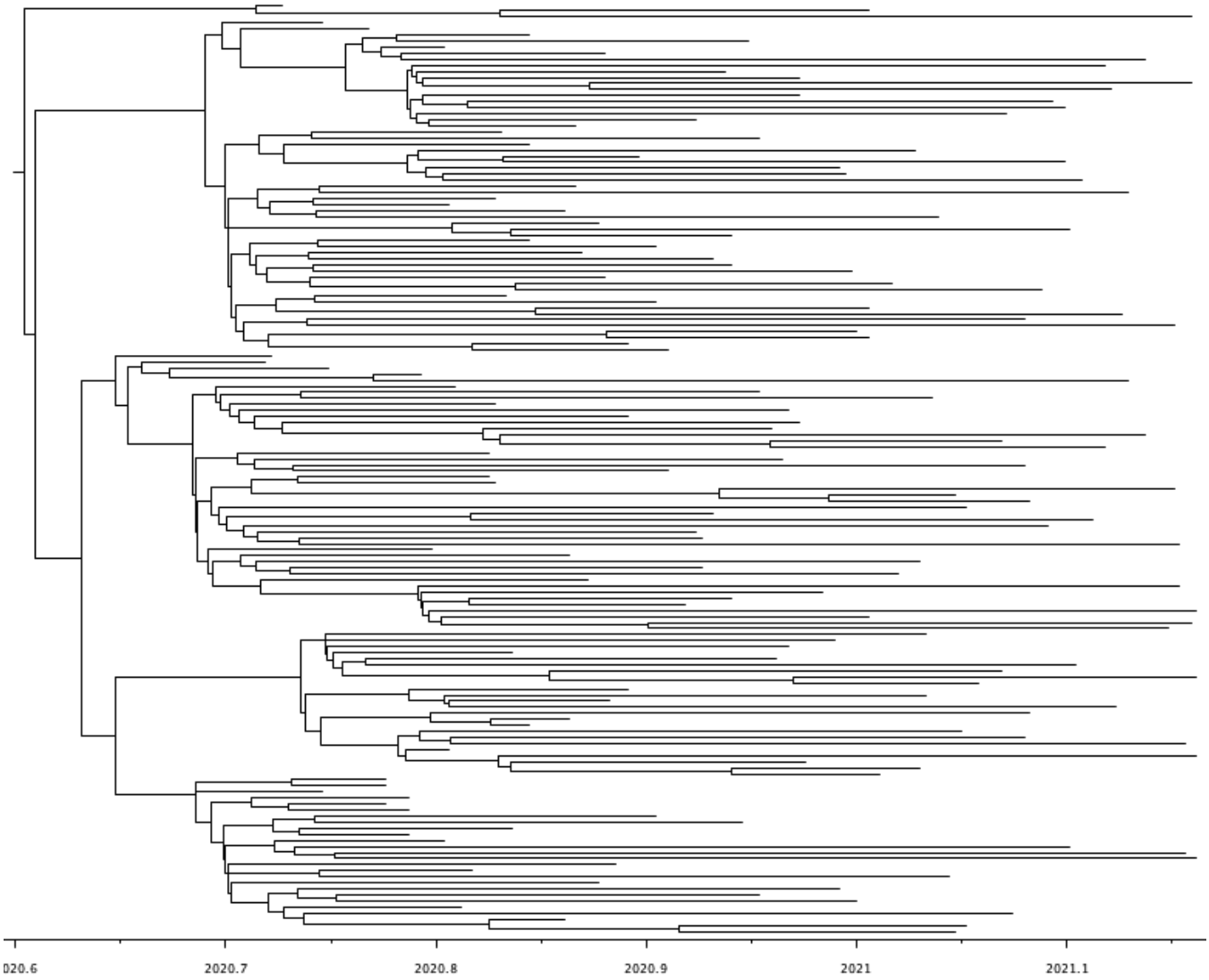

Figure 31: The MCC tree of the Alpha alignment inferred by the Bayesian Skyline Plot.

3.11 Limitations

Both the coalescent and the birth-death skylines assume that the population is well-mixed. That is, they assume that there is no significant population structure and that the sequences are a random sample from the population. However, if there is population structure, for instance sequences were sampled from two different villages and there is much more contact within than between villages, then the results will be biased (Heller et al. 2013). Instead a structured model should then be used to account for these biases.

For the coalescent Bayesian Skyline plot we noticed that both skylines show a constant effective population size after November 2020 (Figure 19). But this doesn't agree with what we know about the COVID-19 epidemic in the UK, where cases (especially of the Alpha VOC) increased steeply over December 2020 and then started falling steeply after the New Year with the start of a lockdown.

To understsand why the Bayesian Skyline Plot can't infer these dynamics on our dataset we have to create

the MCC trees in TreeAnnotator. We see that the MCC tree for the Alpha alignment has very long terminal branches, and many coalescent events deeper in the tree (Figure 31). This is indicative of fast exponential growth dynamics. When the population is large (in the present), the probability of two lineages coalescing is small, resulting in long terminal branches. When the population is small (in the past) the opposite is true, and brannch lengths are short, resulting in many coalescent events. For the The Bayesian Skyline Plot only the coalescent events are informative of N_e. Therefore, the model detects fast growth during the period of many coalescent events (October/November 2020), but has no power to detect any changes in N_e after this time. Because we only have a tiny sample of a huge epidemic wave, almost all of the coalescences in our tree occurred in October/November 2020. Had we taken a much larger sample it is possible that the model would have been able to infer more recent changes in N_e. In addition, there are variations of the coalescent model that either assumes the growth rate remains constant (i.e. N_e will keep growing) (Volz and Didelot 2018) or so-called preferential sampling coalescent models that uses information from the sampling times to also inform N_e (Karcher et al. 2020; Parag et al. 2020; Karcher et al. 2016).

The Birth Death Skyline model uses both coalescent and sampling events to inform itself about the rates. However, it is apparent that our datasets are not informative enough to allow for tight estimates of R_e (Figure 28). It is also apparent that most of the information for the Alpha R_e and sampling proportion estimates come from the sampling times, and not the sequences (Figure 30). This is also true to some extent for the background alignment, although the posterior estimates deviate more from the prior.

With the background sampling proportion it is interesting to note that the estimates include 0 in their HPDs until August 1st, 2020, the time of the oldest sample in our dataset. After that, the sampling proportion jumps and then decreases again. This agrees with the situation in the UK at the time, where cases were very low during August, but started increasing from September 2020. Usually, we would want to set the sampling proportion to 0 before our oldest sample. At the moment this requires editing the XML file. See here for more information on how to do this or **ask me for help**!

Because SARS-CoV-2 evolves relatively slowly (around 2 *de novo* mutations per month) sequences are not that informative over short time scales. That means larger datasets or extra information need to be included in order to make accurate inferences.

Version dated: October 2, 2022

References

Bouckaert, R, J Heled, D Kühnert, T Vaughan, CH Wu, D Xie, MA Suchard, A Rambaut, and AJ Drummond. 2014. Beast 2: a software platform for bayesian evolutionary analysis. *PLoS computational biology* 10: e1003537.

Bouckaert, R et al. 2019. Beast 2.5: an advanced software platform for bayesian evolutionary analysis. *PLOS Computational Biology* 15:

Drummond, AJ, A Rambaut, B Shapiro, and OG Pybus. 2005. Bayesian coalescent inference of past population dynamics from molecular sequences. *Molecular biology and evolution* 22: 1185–92.

Gill, MS, P Lemey, NR Faria, A Rambaut, B Shapiro, and MA Suchard. 2013. Improving Bayesian Population Dynamics Inference: A Coalescent-Based Model for Multiple Loci. *Molecular Biology and Evolution* 30:

Heled, J and AJ Drummond. 2008. Bayesian inference of population size history from multiple loci. *BMC Evolutionary Biology* 8: 289.

Heled, J and AJ Drummond. 2011. Calibrated Tree Priors for Relaxed Phylogenetics and Divergence Time Estimation. *Systematic Biology* 61: 138–149.

Heller, R, L Chikhi, and HR Siegismund. 2013. The confounding effect of population structure on bayesian skyline plot inferences of demographic history. *PloS one* 8: e62992.

Hill, V et al. 2022. The origins and molecular evolution of SARS-CoV-2 lineage B.1.1.7 in the UK. *Virus Evolution* 8: veac080.

Karcher, MD, LM Carvalho, MA Suchard, G Dudas, and VN Minin. 2020. Estimating effective population size changes from preferentially sampled genetic sequences. *PLOS Computational Biology* 16: 1–22.

Karcher, MD, JA Palacios, T Bedford, MA Suchard, and VN Minin. 2016. Quantifying and mitigating the effect of preferential sampling on phylodynamic inference. *PLOS Computational Biology* 12: 1–19.

Minin, VN, EW Bloomquist, and MA Suchard. 2008. Smooth skyride through a rough skyline: bayesian coalescent-based inference of population dynamics. *Molecular biology and evolution* 25: 1459–71.

Parag, KV, L du Plessis, and OG Pybus. 2020. Jointly Inferring the Dynamics of Population Size and Sampling Intensity from Molecular Sequences. *Molecular Biology and Evolution* 37: 2414–2429.

Pybus, OG, A Rambaut, and PH Harvey. 2000. An Integrated Framework for the Inference of Viral Population History From Reconstructed Genealogies. *Genetics* 155:

Rosenberg, NA and M Nordborg. 2002. Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. *Nature Reviews Genetics* 3:

Stadler, T, D Kuhnert, S Bonhoeffer, and AJ Drummond. 2013. Birth-death skyline plot reveals temporal changes of epidemic spread in hiv and hepatitis c virus (hcv). *Proceedings of the National Academy of Sciences* 110: 228–233.

Volz, EM and X Didelot. 2018. Modeling the Growth and Decline of Pathogen Effective Population Size Provides Insight into Epidemic Dynamics and Drivers of Antimicrobial Resistance. *Systematic Biology* 67: 719–728.