Tutorial BEAST

Luis Amador

2022-10-26

Descargar BEAST

El primer paso es descargar el software BEAST2 desde su página web https://www.beast2.org/>. Hay opciones para descargar el programa en Windows, Mac OS X y Linux. En este tutorial se presenta la descarga y posterior análisis desde un computador con Windows como sistema operativo.

Una vez que BEAST sea descargado, descomprimir el archivo y comenzar con el programa BEAUti dando click al icono dentro de la carpeta BEAST (Fig 1).

Beauti

Beauti (Fig. 2) es un programa dentro de BEAST, en este se importará el alineamiento (en formato NEXUS) con las secuencias a analizar, se subirá información sobre el modelo evolutivo y otros parámetros evolutivos y para el algoritmo Markov chain Monte Carlo (MCMC). Beauti es el programa necesario para crear el archivo xml que servirá como input para BEAST.

Cargar paquetes en Beauti

Es posible instalar otros programas de BEAST2 utilizando Beauti. Para eso tenemos que ir a File luego a Manage Packages y elegir los paquetes que queremos instalar. Es importante que para el buen funcionamiento de lo paquetes, Beauti sea reiniciado luego de la descarga. En la Figura 3 se muestra un ejemplo de cómo se ve la ventana del Package Manager y en azul tres de los paquetes que son necesarios descargar para este tutorial (si es que no están ya instalados).

Importar alineamiento

En Beauti ir a *file* y dar click en *Import Alignment* y escogemos el archivo del alineamiento en formato nexus que vamos a analizar (Fig. 4). En este ejemplo voy a usar un archivo con 230 secuencias de un grupo de camarones de vega chilenos secuenciados con el gen mitocondrial COI (fig. 5).

Incluir el modelo de sustitución nucleotídica

El siguiente paso es incluir el mejor modelo de sustitución nucleotídica obtenido para nuestros datos, es decir nuestro alineamiento. Para el alineamiento de este ejemplo, calculamos el mejor modelo en el programa ModelFinder implementado en IQ-TREE y el mejor modelo fue HKY+I+G4 (I= proporción de sitios invariantes, G4= Gamma Category Count 4). Agregamos esta información en Beauti donde corresponde (Fig. 6), se puede dejar la casilla de Shape marcada con el visto, pero como ModelFinder nos da tanto el valor de I (Proportion of invariable sites: 0.5069) como de Shape (Gamma shape alpha: 0.7863) los incluimos manualmente y desmarcamos la casilla 'estimate' de Shape.

Reloj molecular

En la pestaña Clock Model aparecen varias opciones para reloj molecular. Escogemos Optimised Relaxed Clock, el cual debe aparecer si instalamos previamente el paquete ORC (ver Fig. 3). El modelo de reloj relajado es ampliamente usado en filogenética y permite la estimación de diferentes tasas de sustitución entre

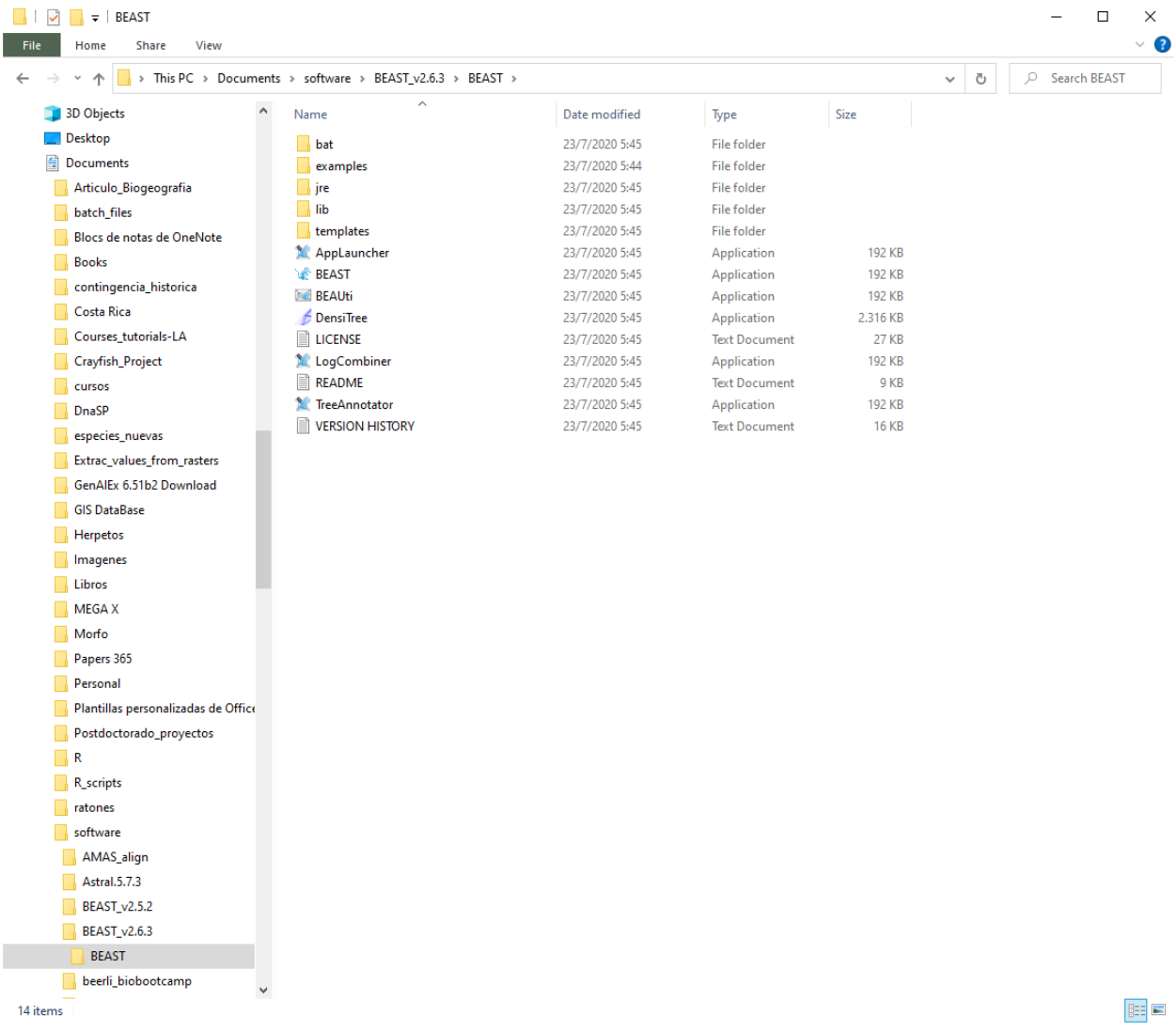

Figure 1: Carpeta con el contenido de los paquetes de BEAST incluido Beauti

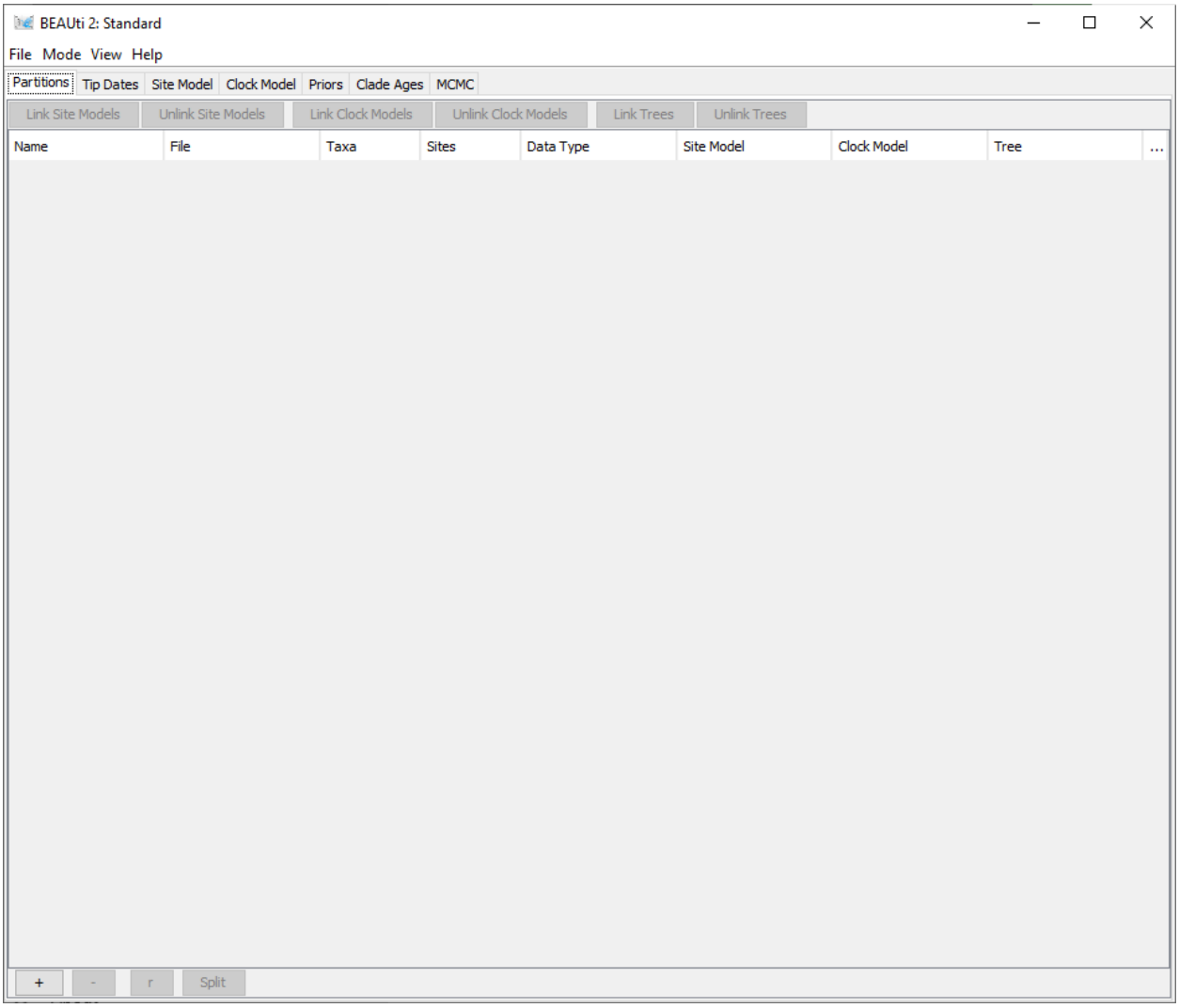

Figure 2: Ventana de inicio de Beauti

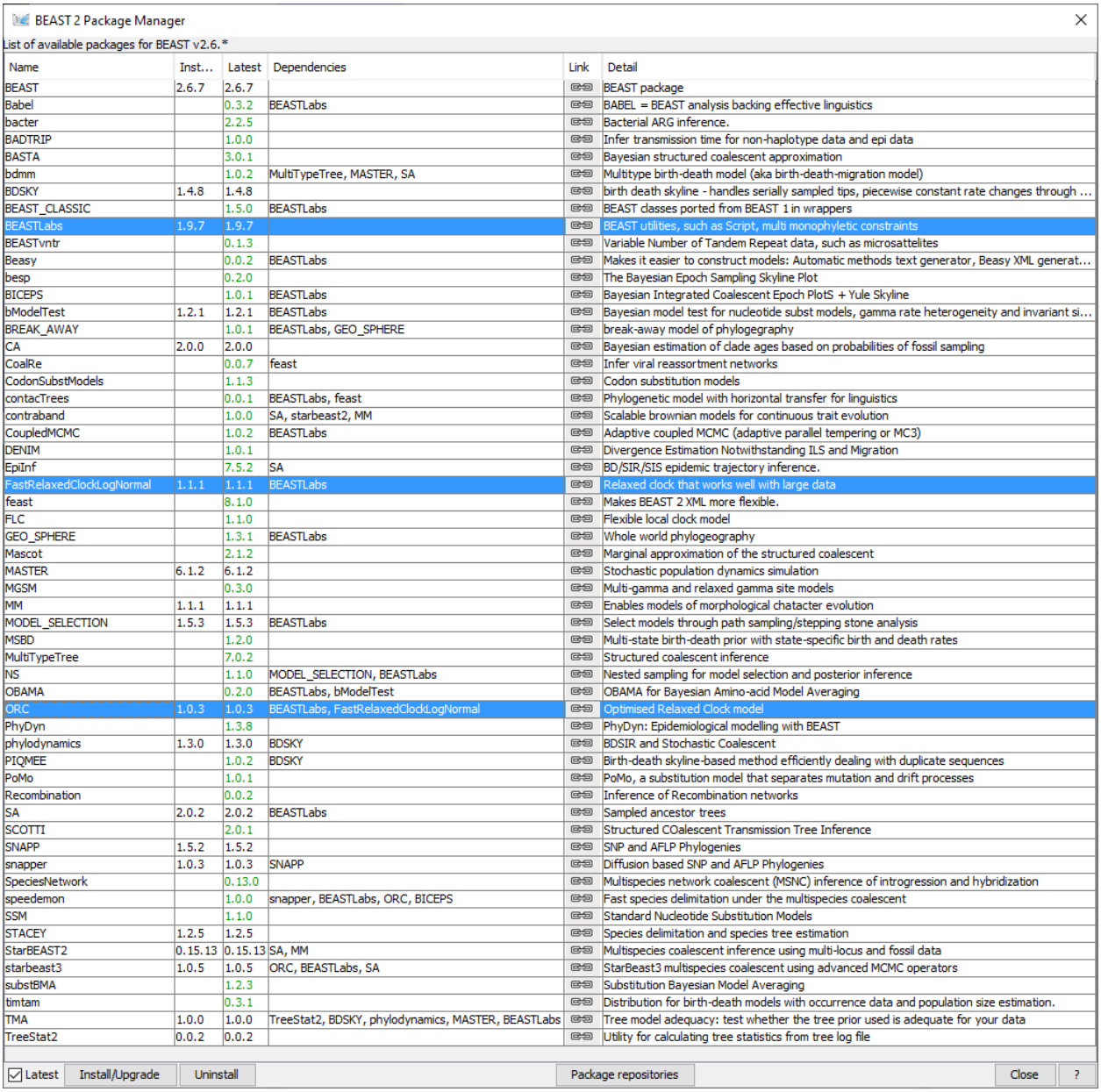

Figure 3: Ventana de Package Manager en BEAST2

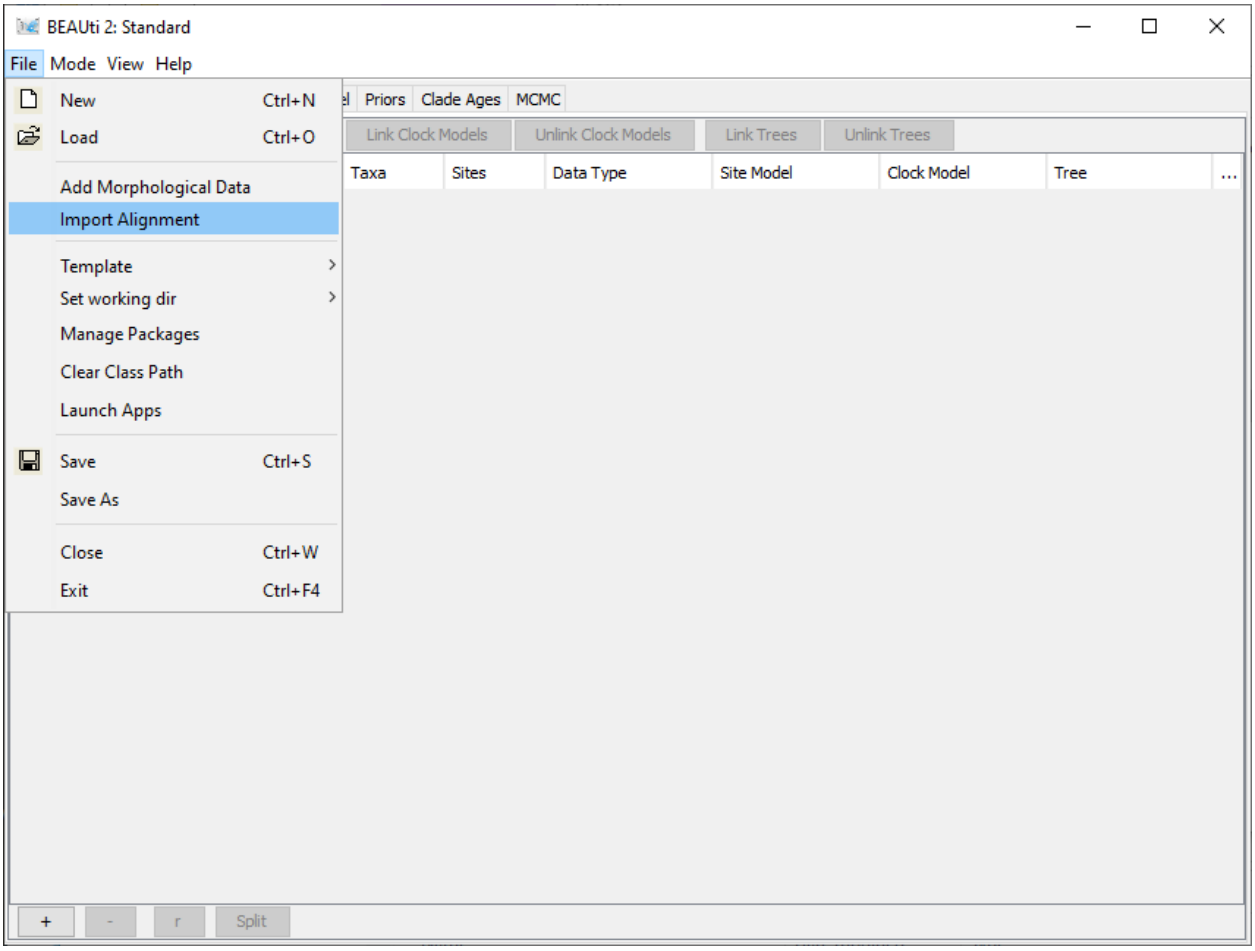

Figure 4: Subir el alineamiento en Beauti

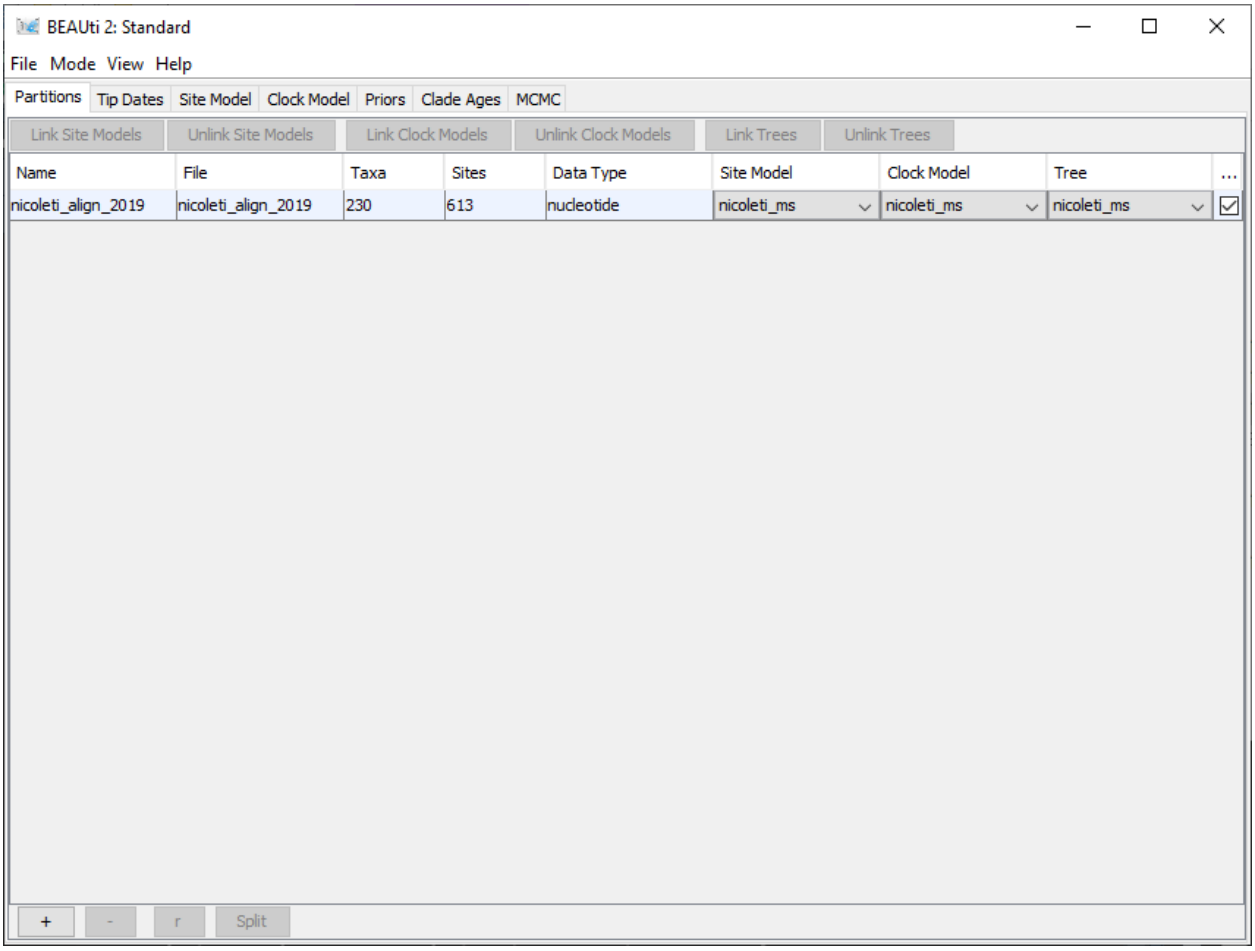

Figure 5: Alineamiento cargado en Beauti

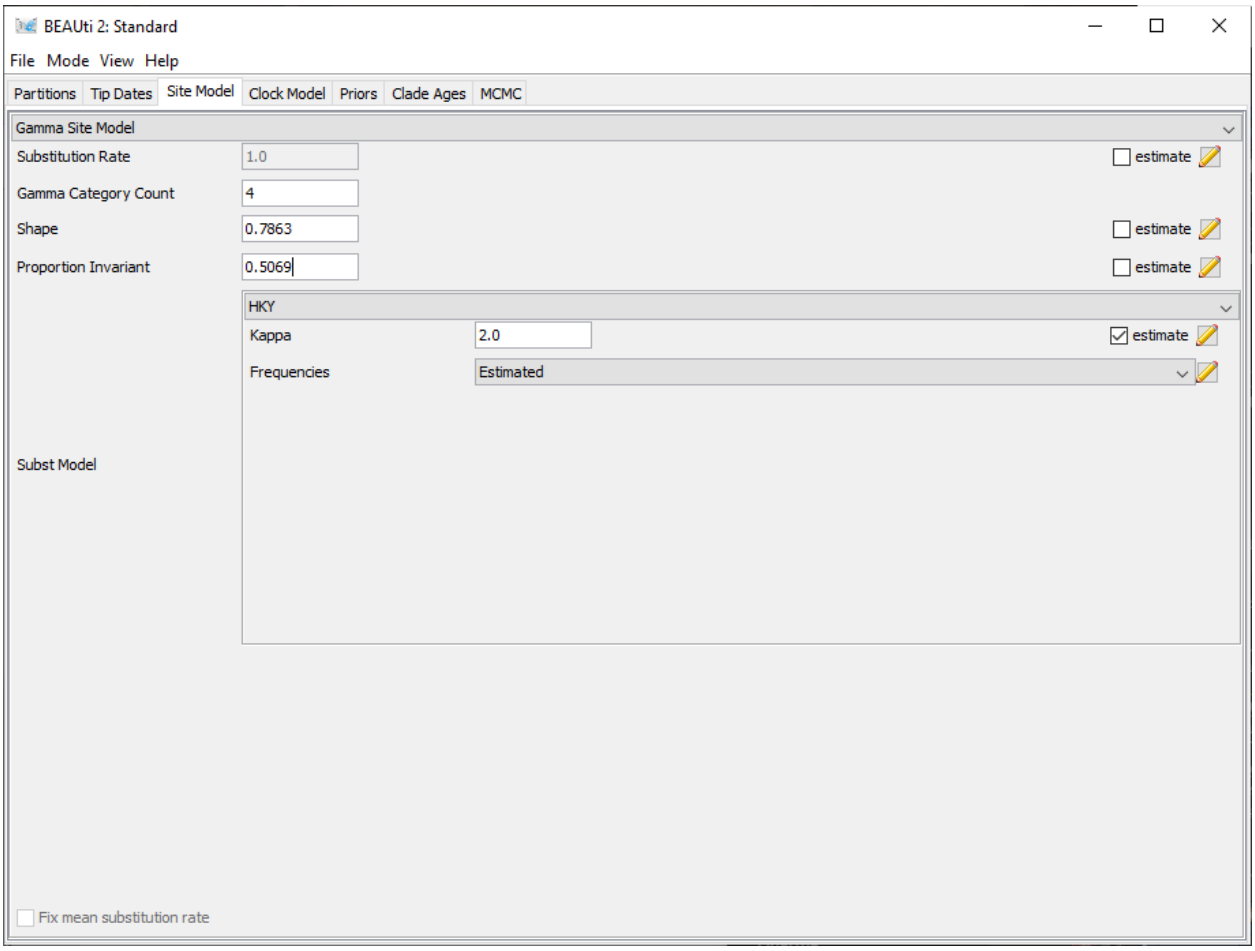

Figure 6: Opción en Beauti para incluir los parámetros del modelo de sustitución

linajes (e.g., nuestro alineamiento está compuesto de secuencias de varias especies de camarones de vega). En cambio, si nuestro alineamiento es de secuencias pertenecientes a una misma especie o población podemos usar un reloj estricto que asume que cada secuencia presenta la misma tasa de sustitución. En nuestro caso elegimos Optimised Relaxed Clock y dejamos los demás valores como están por default (Fig. 7).

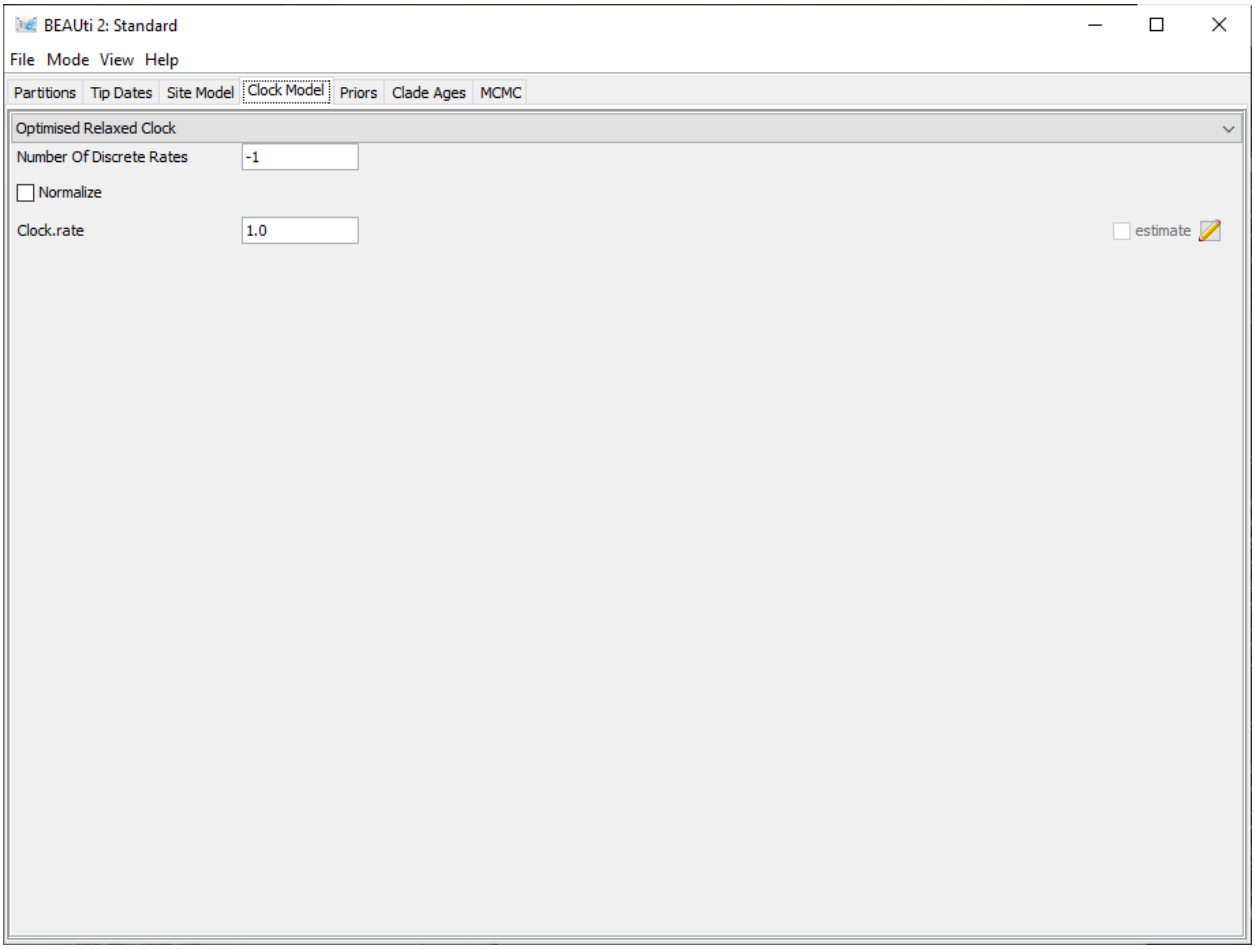

Figure 7: Opción en Beauti para escoger el reloj molecular

Tree Priors

En inferencia Bayesiana debemos especificar distribuciones de probabilidad previa (priors) para los parámetros del modelo, incluyendo el árbol (tree priors). En este ejemplo vamos a utilizar un modelo Birth-Death (Fig. 8) como tree prior, este modelo ampliamente usado para estudiar como las especies (o individuos en una población) cambian através del tiempo (procesos de especiación y extinción). Escogemos este modelo por sobre el modelo Yule (pure birth) que es usado cuando se tiene una única secuencia por especie en nuestro alineamiento y además es utilizado a menudo para casos en los que se espera que las tasas de extinción son insignificantes. Para este ejercicio podemos dejar los demás parámetros igual.

Configuración de las opciones MCMC

La última pestaña en Beauti es MCMC y proporciona ajustes para controlar el funcionamiento de BEAST. Primero está la Longitud de la cadena (Length of chain). Aquí incluimos el número de pasos que la cadena de MCMC hará antes de terminar. Este número va a depender del tamaño de nuestro set de datos, de la complejidad del modelo y de la calidad de la respuesta requerida. El valor por defecto de 10'000.000 y es totalmente arbitrario por lo que debe ajustarse en función del tamaño de su conjunto de datos. Por ejemplo

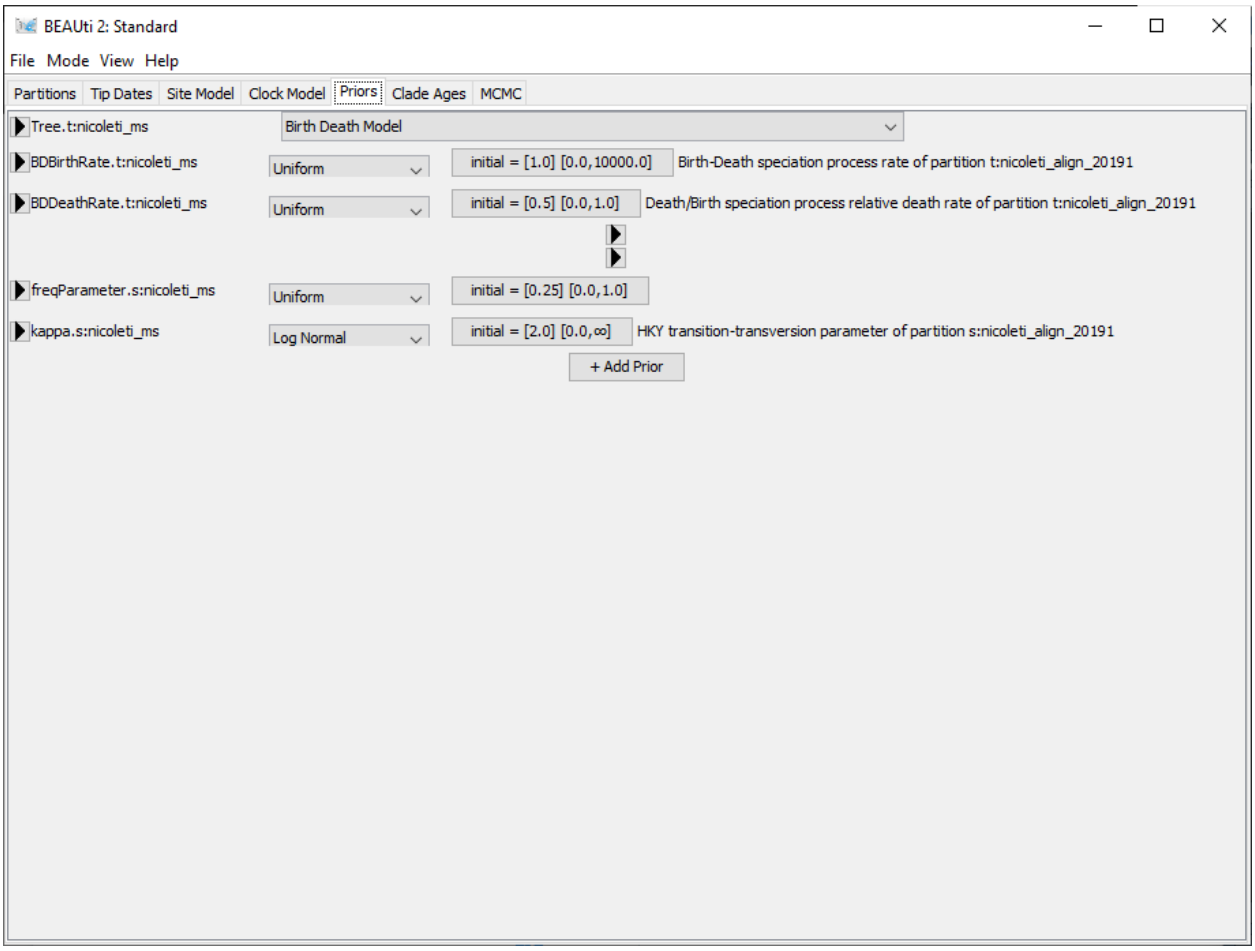

Figure 8: Especificando prior en Beauti

para este conjunto de datos usé entre 50 y 70 millones, pero para el ejemplo lo dejaremos en 1'000.000 ya que es solo para fines explicativos. Dejaremos Store Every, pre Burnin y Num Initialization Attempts con los valores por defecto.

Debajo de estos ajustes generales se encuentran los ajustes de registro. Cada opción particular se puede ver en detalle haciendo clic en la flecha a la izquierda. Se pueden controlar los nombres de los archivos de registro y la frecuencia con la que se almacenarán los valores en cada uno de los archivos.

Comenzaemos por tracelog. El parámetro Log Every para el archivo de registro debe establecerse en relación con la longitud total de la cadena. Un muestreo demasiado frecuente (por ejemplo cada 100) dará lugar a archivos muy grandes con poco beneficio adicional en términos de precisión del análisis. Un muestreo demasiado disperso (e.g., > 1000) significará que el archivo de registro no registrará suficiente información sobre las distribuciones de los parámetros. Para este análisis haremos que BEAST2 escriba en el archivo de registro cada 500 muestras.Dejamos el File Name igual.

A continuación, expandemos las opciones para ver las opciones de screenlog. Dejamos los valores por defecto, es decir la casilla de File Name vacía y Log Every = 1000. Por último podemos cambiar las opciones de treelog haciendo click en la flecha izquierda, aquí si cambiamos el nombre en File Name (camarones.trees) y dejamos las demás opciones por defecto (Fig. 9).

Generación del archivo XML

Ahora estamos listos para crear el archivo xml de BEAST2. Este es el archivo de configuración final que BEAST2 puede utilizar para ejecutar el análisis. Guardaremos el archivo xml con el nombre camarones.xml utilizando File > Save. Ahora estamos listos para ejecutar el archivo a través de BEAST.

Ejecutar BEAST

Ahora ejecutamos BEAST2 y seleccionamos el archivo xml (camarones.xml) desde donde lo hemos guardado. También si se quiere se puede cambiar el 'seed number' - semilla de números aleatorios para la ejecución. Marcamos la casilla 'Use BEAGLE library' si es que está disponible. Si se ha instalado previamente BEAGLE, esto hará que el análisis se ejecute más rápidamente. Corremos BEAST2 haciendo click en el botón Run (Fig. 10).

BEAST2 se ejecutará hasta alcanzar el número especificado de pasos en la cadena. Mientras se ejecuta, imprimirá los valores de screenlog en una consola y almacenará los valores de tracelog y tree log en archivos situados en la misma carpeta que el archivo xml. La salida en pantalla será aproximadamente como se muestra en la Figura 11.

Analizando los resultados

Para examinar si una determinada longitud de cadena es adecuada, el archivo .log resultante del análisis BEAST puede analizarse con el programa Tracer. El objetivo de establecer la longitud de la cadena es conseguir un tamaño de muestra efectivo (ESS) razonable (> 200). Esto lo podemos analizar con más detalle en otro tutorial.

Ahora nos centraremos en obtener un árbol Bayesiano, el maximum clade credibility tree. BEAST produce una muestra posterior de árboles temporales filogenéticos (archivo de salida .trees). Estos árboles deben ser resumidos antes de poder sacar cualquier conclusión sobre la calidad de la estimación filogenética. Una forma de resumir los árboles es utilizando el programa TreeAnnotator que viene junto con BEAST al descargarse. Éste tomará el conjunto de árboles y encontrará el árbol con mejor soporte. También calculará la probabilidad de clado posterior para cada nodo. Este árbol se denomina árbol de máxima credibilidad de clados.

A continuación abrimos TreeAnnotator, establecemos el porcentaje de Burnin en 10% para descartar el primer 10% de los árboles de nuestro archivo .trees. Dejamos el Posterior probability limit igual es decir en cero, esto significa que TreeAnnotator anotará todos los nodos. La siguiente opción la dejaremos igual por default y en Node heights escogemos Mean heights. Por último, tenemos que seleccionar el archivo de

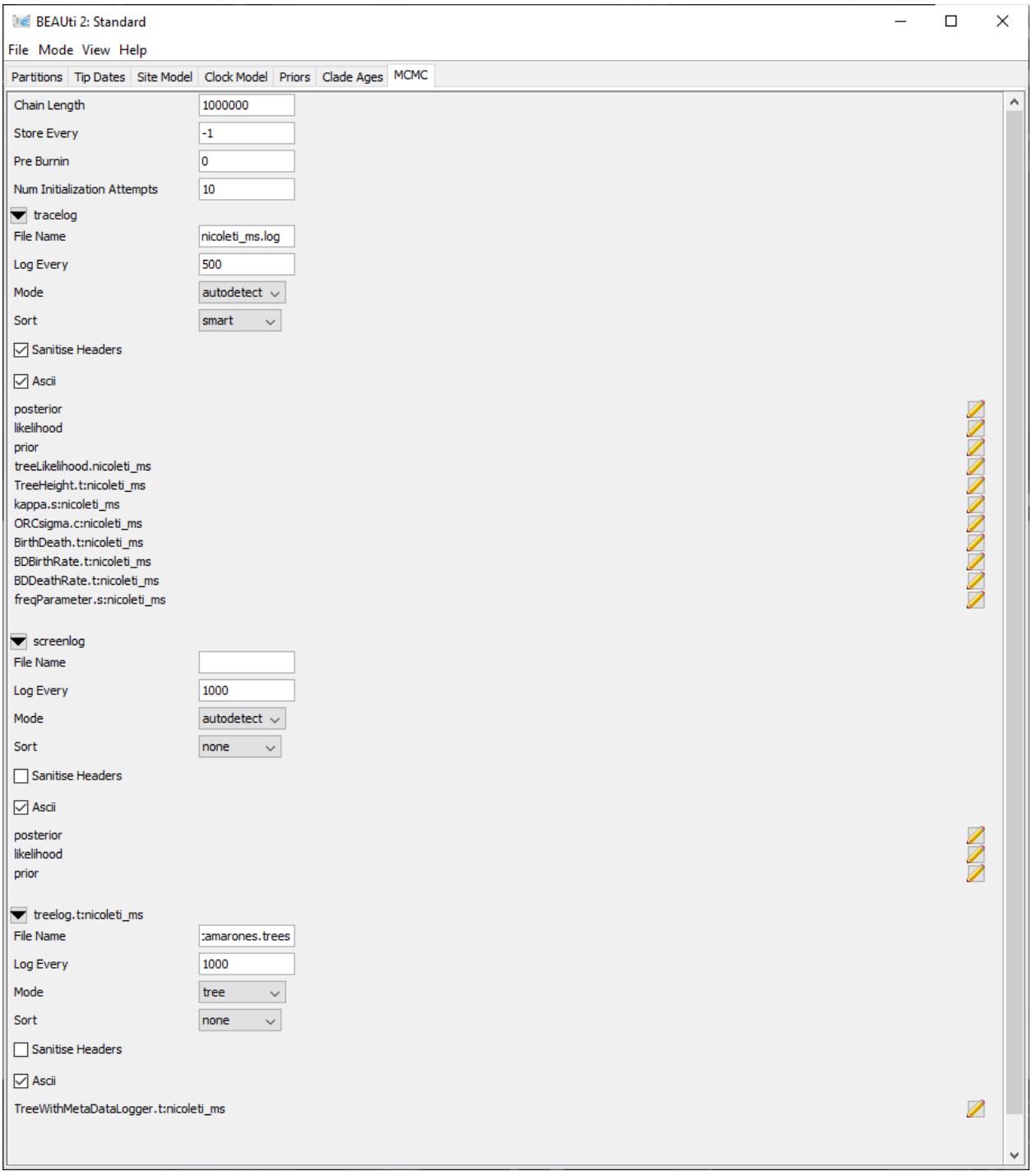

Figure 9: Opciones MCMC

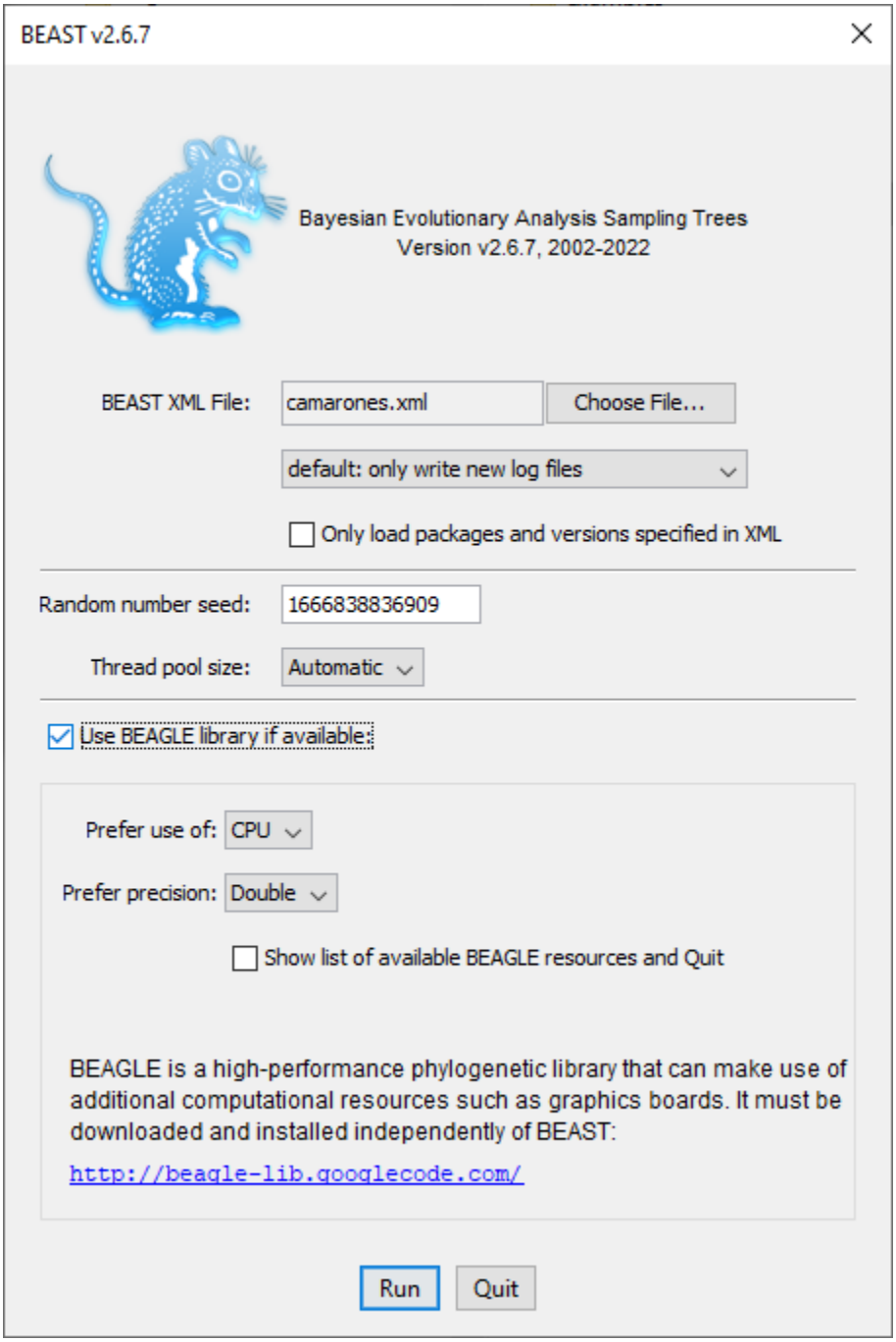

Figure 10: Ejecutando el programa BEAST2

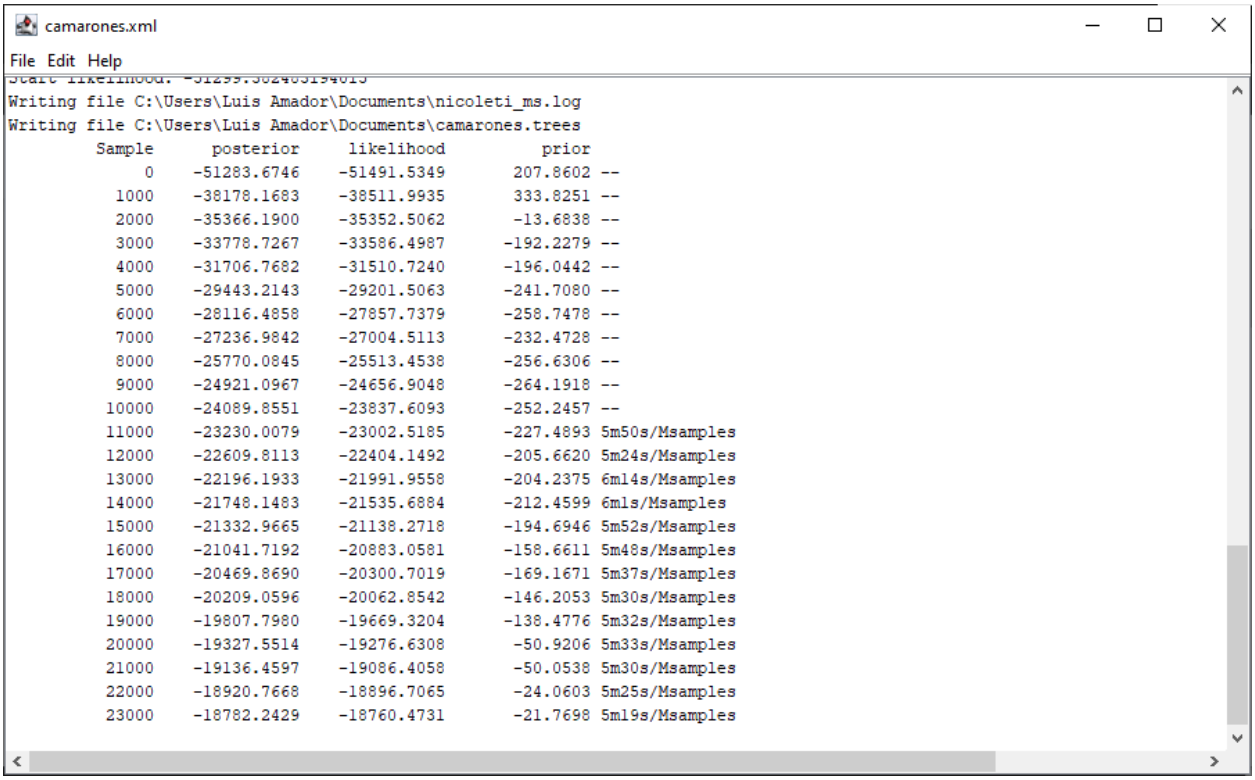

Figure 11: Ejecutando el programa BEAST2

entrada (el archivo camarones.trees de nuestro análisis BEAST) y establecemos un archivo de salida como camarones.tree y ejecutamos el programa. La configuración debería ser como se muestra en la Figura 12.

Visualización del árbol

Podemos visualizar el árbol con uno de los programas informáticos disponibles, como FigTree. Abrimos FigTree y luego localizamos el archivo camarones.tree (Fig. 13).

Es ultramétrico?

Podemos ver nuestro árbol pero no sabemos aún si es ultramétrico o no. Para conocer si nuestro árbol obtenido en BEAST es ultramétrico (que debería serlo después de nuestro ajuste) podemos utilizar el software R y especificamente el paquete ape. Leemos nuestro árbol con la función *read.nexus* y preguntamos a R si nuestro árbol es ultramétrico con la función *is.ultrametric* si la respuesta es TRUE quiere decir que neustro árbol es ultramétrico como podemos ver en esta línea de código.

```
#install.packages("ape")
library(ape)
tree <- read.nexus(file = "camarones.tree")
is.ultrametric(tree)
```

```
## [1] TRUE
```

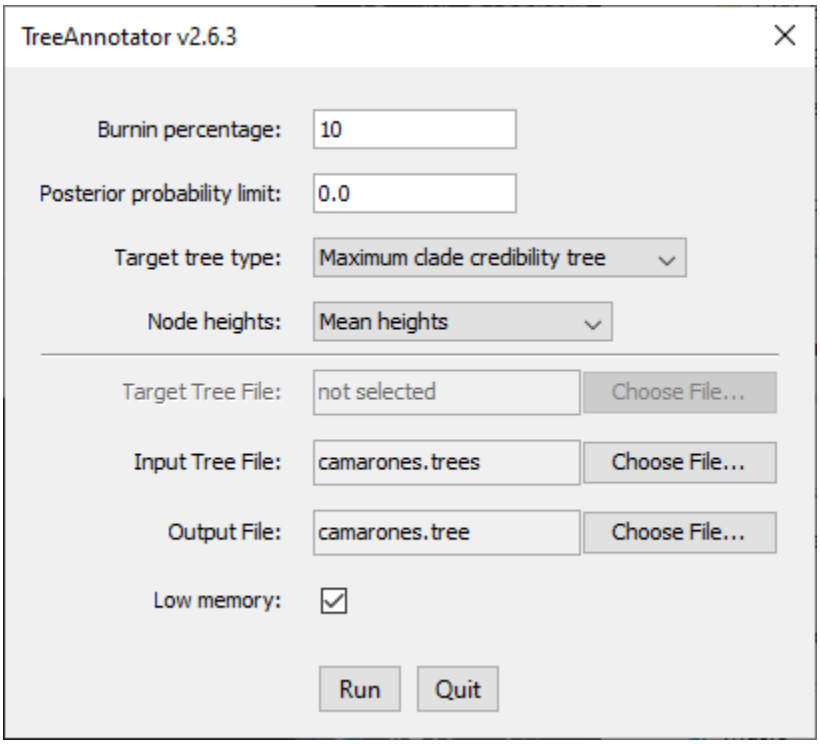

Figure 12: Obteniendo nuestro árbol bayesiano ejecutando TreeAnnotator

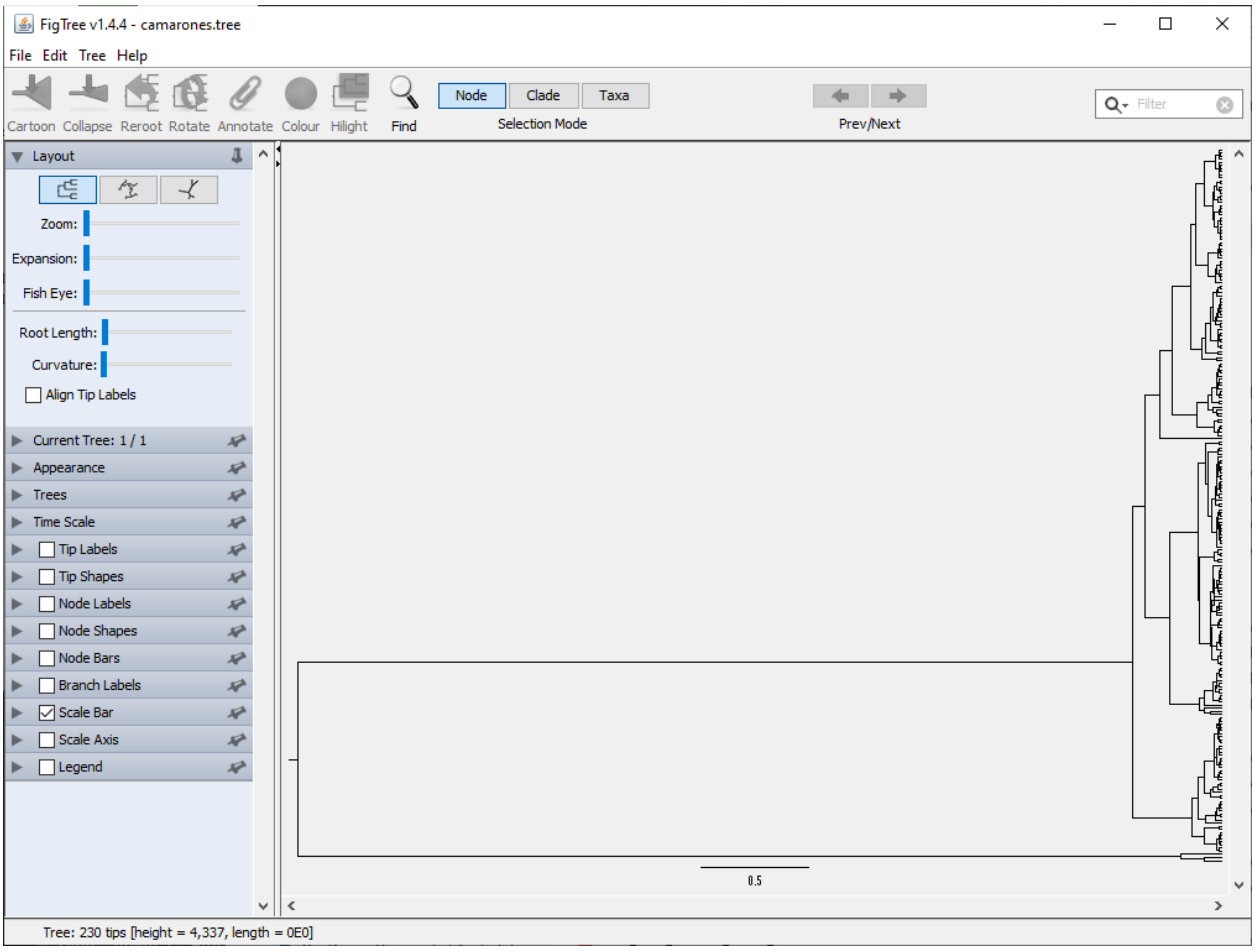

Figure 13: Visualización del árbol con Figtree