

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 los 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

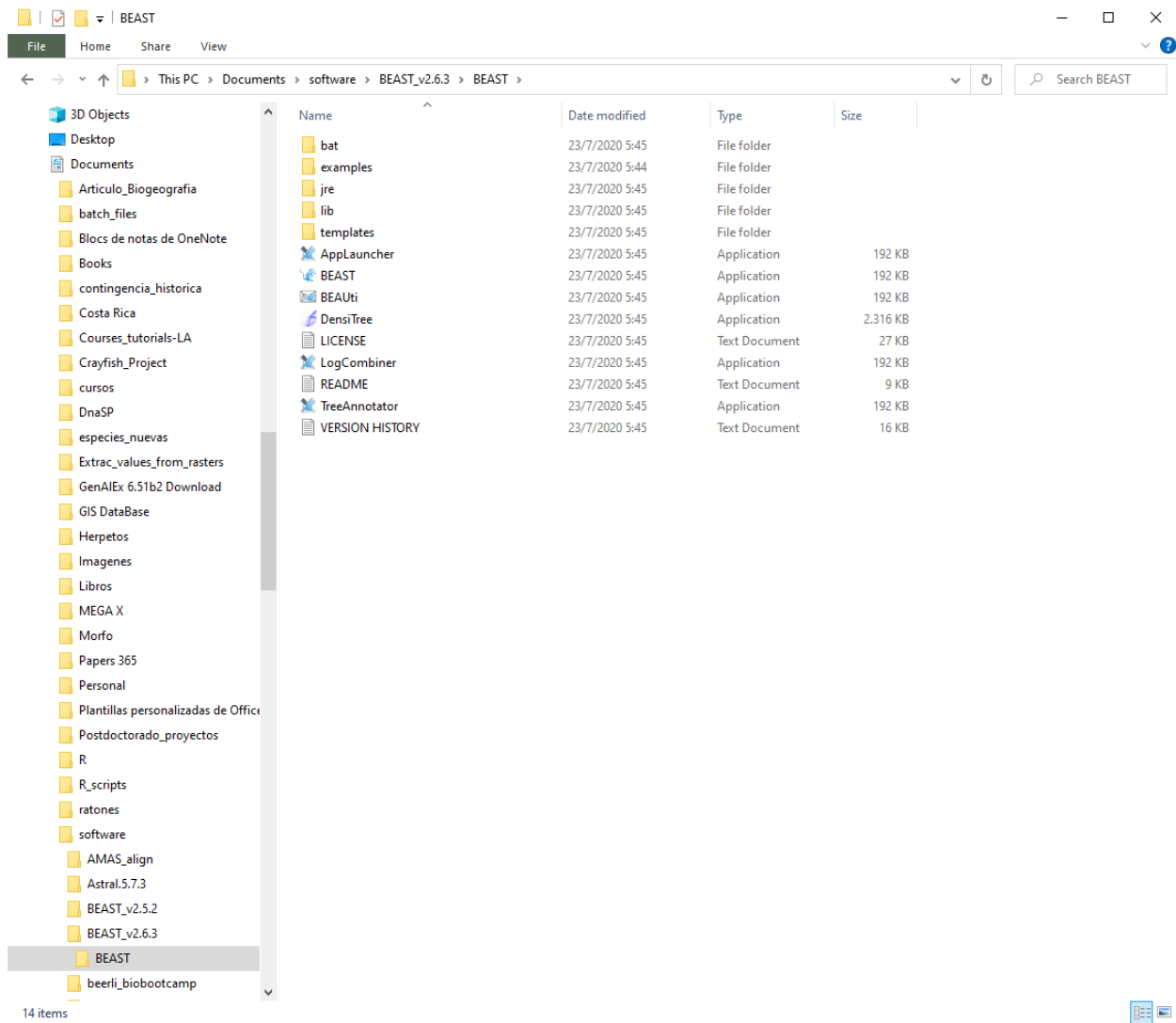


Figure 1: Carpeta con el contenido de los paquetes de BEAST incluido Beauti

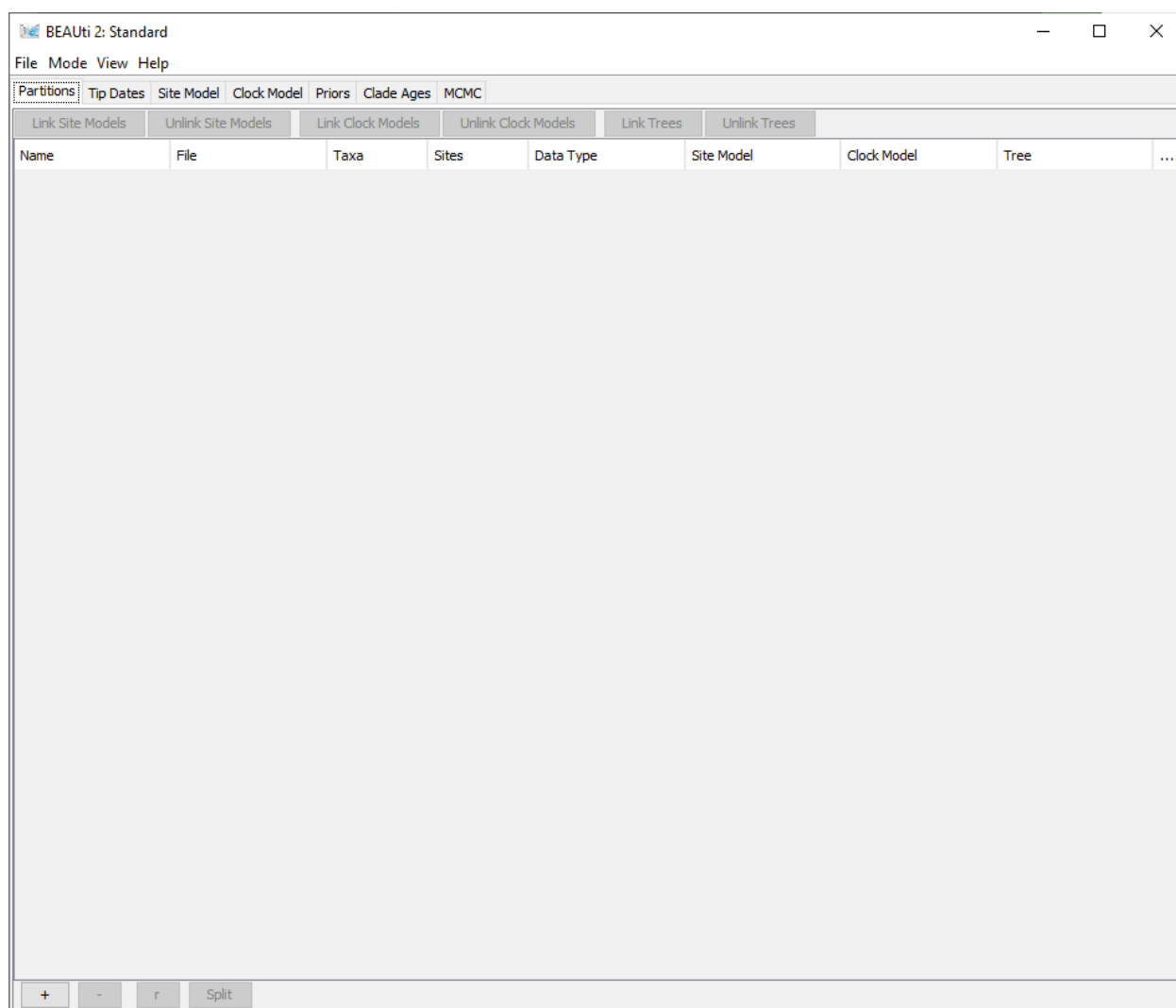


Figure 2: Ventana de inicio de Beauti

BEAST 2 Package Manager					
List of available packages for BEAST v2.6.*					
Name	Inst...	Latest	Dependencies	Link	Detail
BEAST	2.6.7	2.6.7			BEAST package
Babel		0.3.2	BEASTLabs		BABEL = BEAST analysis backing effective linguistics
bacter		2.2.5			Bacterial ARG inference.
BADTRIP		1.0.0			Infer transmission time for non-haplotype data and epi data
BASTA		3.0.1			Bayesian structured coalescent approximation
bdmm		1.0.2	MultiTypeTree, MASTER, SA		Multitype birth-death model (aka birth-death-migration model)
BDSKY	1.4.8	1.4.8			birth death skyline - handles serially sampled tips, piecewise constant rate changes through ...
BEAST_CLASSIC		1.5.0	BEASTLabs		BEAST classes ported from BEAST 1 in wrappers
BEASTLabs	1.9.7	1.9.7			BEAST utilities, such as Script, multi monophyletic constraints
BEASTvnr		0.1.3			Variable Number of Tandem Repeat data, such as microsatellites
Beasy		0.0.2	BEASTLabs		Makes it easier to construct models: Automatic methods text generator, Beasy XML generat...
besp		0.2.0			The Bayesian Epoch Sampling Skyline Plot
BICEPS		1.0.1	BEASTLabs		Bayesian Integrated Coalescent Epoch Plots + Yule Skyline
bModelTest	1.2.1	1.2.1	BEASTLabs		Bayesian model test for nucleotide subst models, gamma rate heterogeneity and invariant si...
BREAK_AWAY		1.0.1	BEASTLabs, GEO_SPHERE		break-away model of phylogeography
CA	2.0.0	2.0.0			Bayesian estimation of clade ages based on probabilities of fossil sampling
CoalRe		0.0.7	feast		Infer viral reassortment networks
CodonSubstModels		1.1.3			Codon substitution models
contactTrees		0.0.1	BEASTLabs, feast		Phylogenetic model with horizontal transfer for linguistics
contraband		1.0.0	SA, starbeast2, MM		Scalable brownian models for continuous trait evolution
CoupledMCMC		1.0.2	BEASTLabs		Adaptive coupled MCMC (adaptive parallel tempering or MC3)
DENIM		1.0.1			Divergence Estimation Notwithstanding ILS and Migration
EpiInf		7.5.2	SA		BD/SIR/SIS epidemic trajectory inference.
FastRelaxedClockLogNormal	1.1.1	1.1.1	BEASTLabs		Relaxed clock that works well with large data
feast		8.1.0			Makes BEAST 2 XML more flexible.
FLC		1.1.0			Flexible local clock model
GEO_SPHERE		1.3.1	BEASTLabs		Whole world phylogeography
Mascot		2.1.2			Marginal approximation of the structured coalescent
MASTER	6.1.2	6.1.2			Stochastic population dynamics simulation
MGSM		0.3.0			Multi-gamma and relaxed gamma site models
MM	1.1.1	1.1.1			Enables models of morphological character evolution
MODEL_SELECTION	1.5.3	1.5.3	BEASTLabs		Select models through path sampling/stepping stone analysis
MSBD		1.2.0			Multi-state birth-death prior with state-specific birth and death rates
MultiTypeTree		7.0.2			Structured coalescent inference
NS		1.1.0	MODEL_SELECTION, BEASTLabs		Nested sampling for model selection and posterior inference
OBAMA		0.2.0	BEASTLabs, bModelTest		OBAMA for Bayesian Amino-acid Model Averaging
ORC	1.0.3	1.0.3	BEASTLabs, FastRelaxedClockLogNormal		Optimised Relaxed Clock model
PhyDyn		1.3.8			PhyDyn: Epidemiological modelling with BEAST
phylodynamics	1.3.0	1.3.0	BDSKY		BDSIR and Stochastic Coalescent
PIQMEE		1.0.2	BDSKY		Birth-death skyline-based method efficiently dealing with duplicate sequences
PoMo		1.0.1			PoMo, a substitution model that separates mutation and drift processes
Recombination		0.0.2			Inference of Recombination networks
SA	2.0.2	2.0.2	BEASTLabs		Sampled ancestor trees
SCOTTI		2.0.1			Structured COalescent Transmission Tree Inference
SNAPP	1.5.2	1.5.2			SNP and AFLP Phylogenies
snapper	1.0.3	1.0.3	SNAPP		Diffusion based SNP and AFLP Phylogenies
SpeciesNetwork		0.13.0			Multispecies network coalescent (MSNC) inference of introgression and hybridization
speedemon		1.0.0	snapper, BEASTLabs, ORC, BICEPS		Fast species delimitation under the multispecies coalescent
SSM		1.1.0			Standard Nucleotide Substitution Models
STACEY	1.2.5	1.2.5			Species delimitation and species tree estimation
StarBEAST2	0.15.13	0.15.13	SA, MM		Multispecies coalescent inference using multi-locus and fossil data
starbeast3	1.0.5	1.0.5	ORC, BEASTLabs, SA		StarBeast3 multispecies coalescent using advanced MCMC operators
substBMA		1.2.3			Substitution Bayesian Model Averaging
timtam		0.3.1			Distribution for birth-death models with occurrence data and population size estimation.
TMA	1.0.0	1.0.0	TreeStat2, BDSKY, phylodynamics, MASTER, BEASTLabs		Tree model adequacy: test whether the tree prior used is adequate for your data
TreeStat2	0.0.2	0.0.2			Utility for calculating tree statistics from tree log file
<input checked="" type="checkbox"/> Latest Install/Upgrade Uninstall Package repositories Close ?					

Figure 3: Ventana de Package Manager en BEAST2

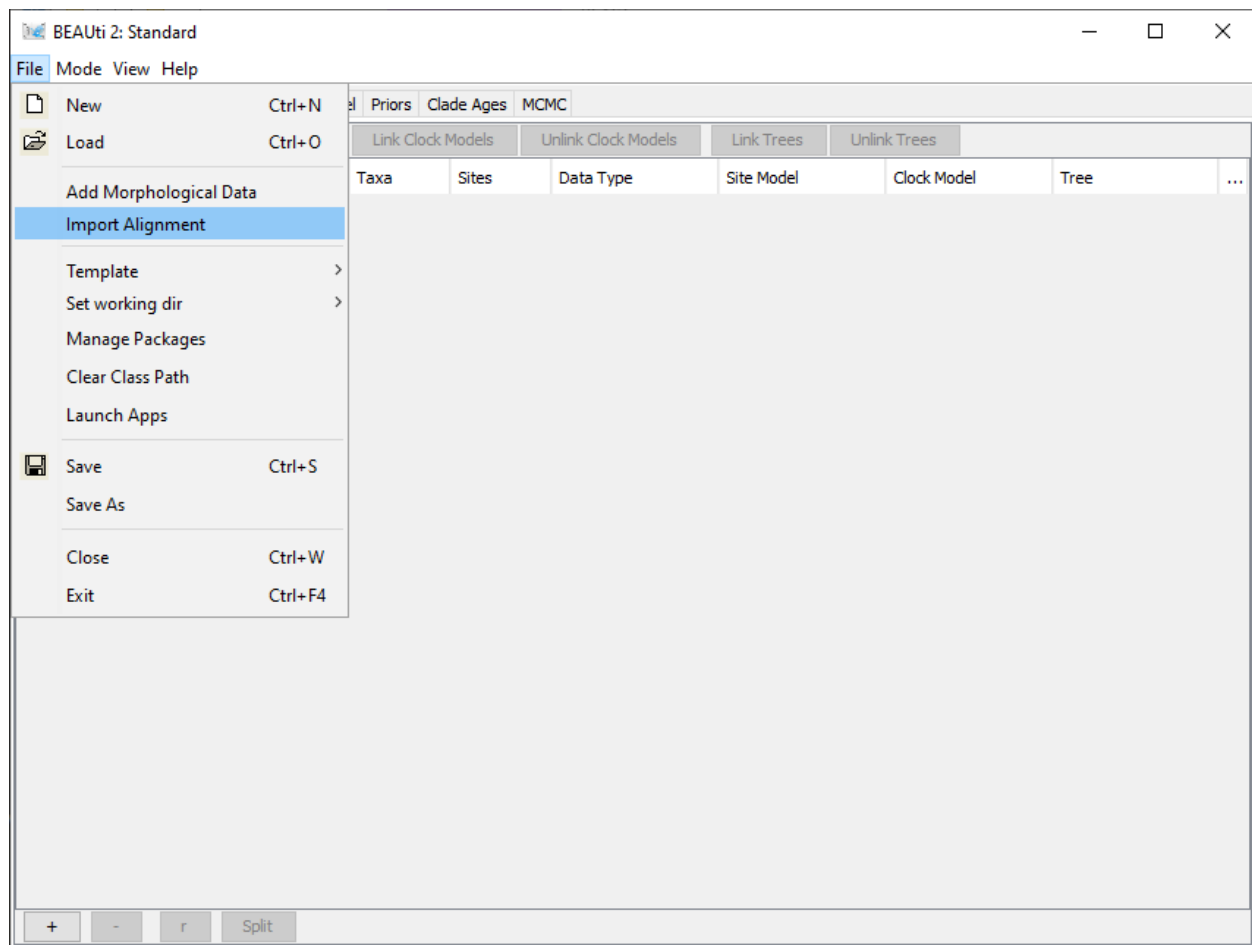


Figure 4: Subir el alineamiento en Beauti

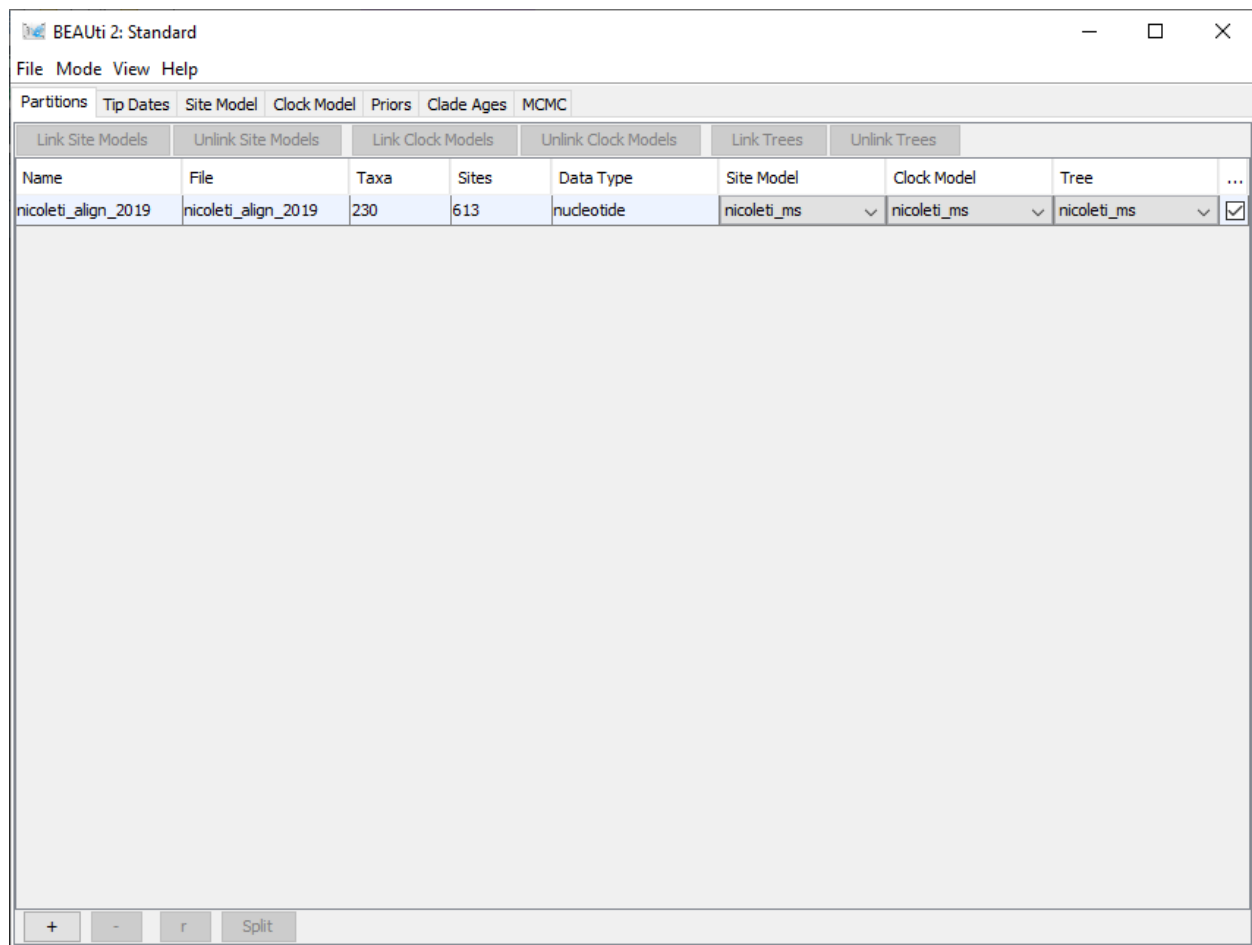


Figure 5: Alineamiento cargado en Beauti

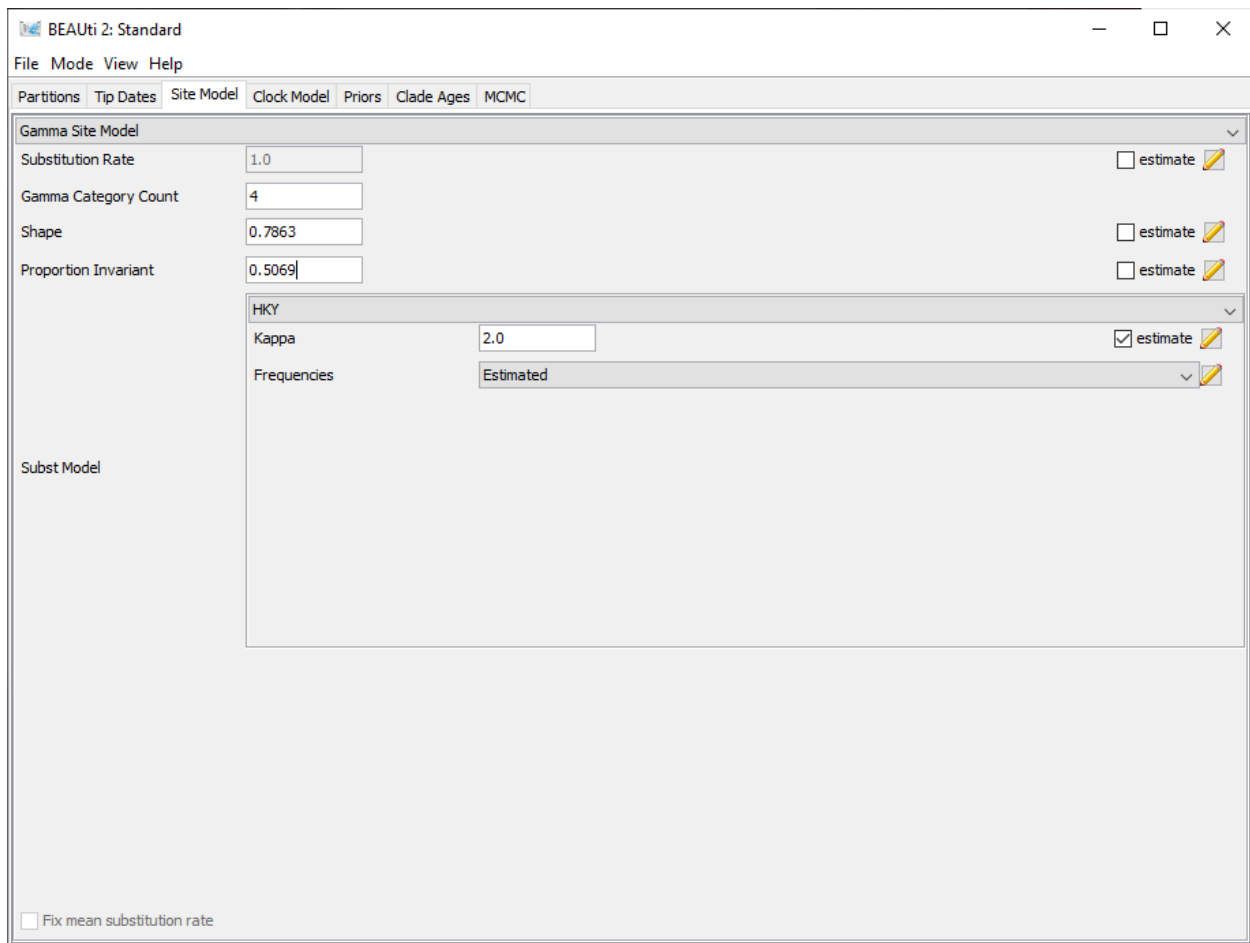


Figure 6: Opción en Beauti para incluir los parámetros del modelo de sustitución

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).

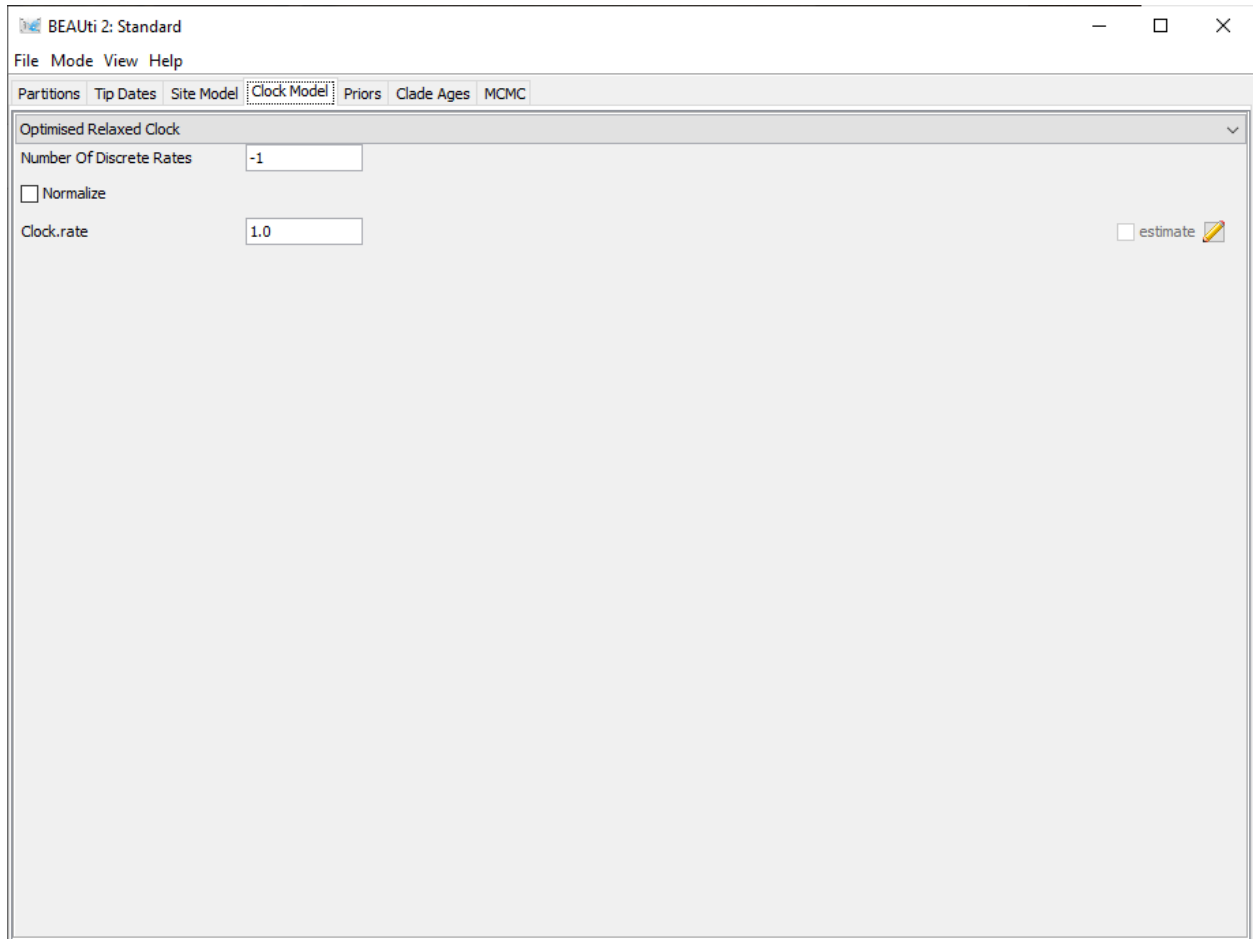


Figure 7: Opción en Beauti para escoger el reloj molecular

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

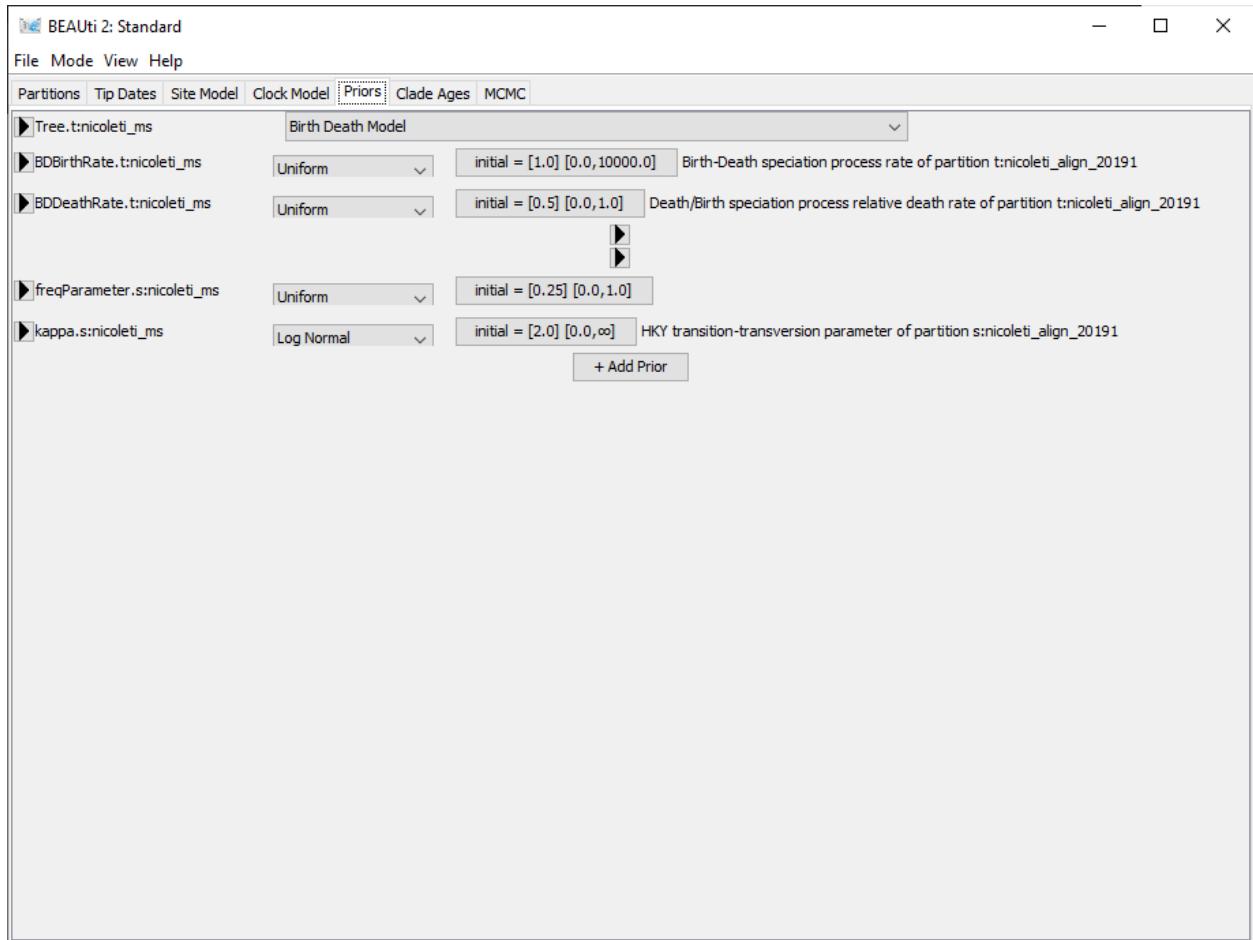


Figure 8: Especificando prior en Beauti

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.

Comenzamos 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 tree log 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

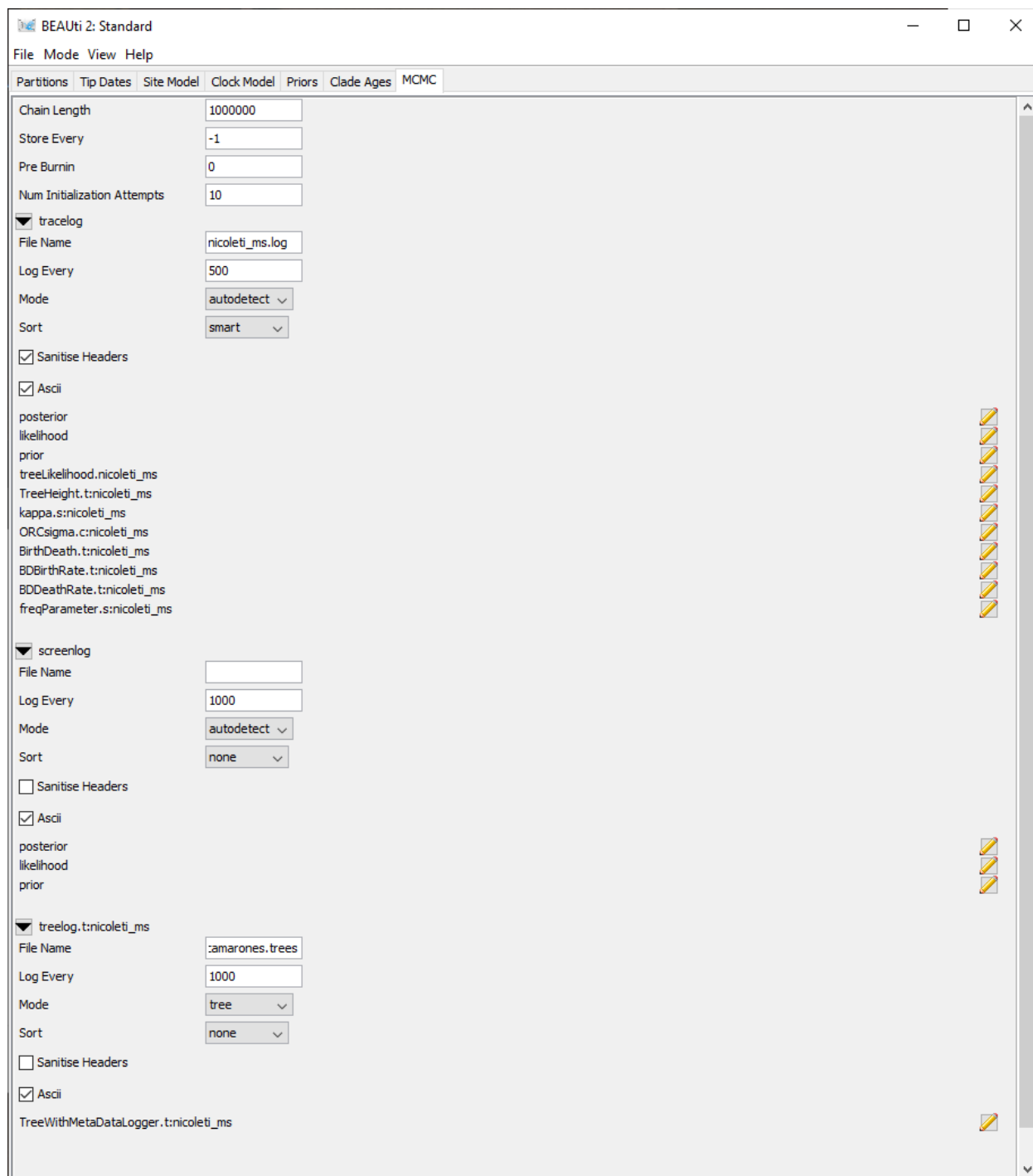


Figure 9: Opciones MCMC

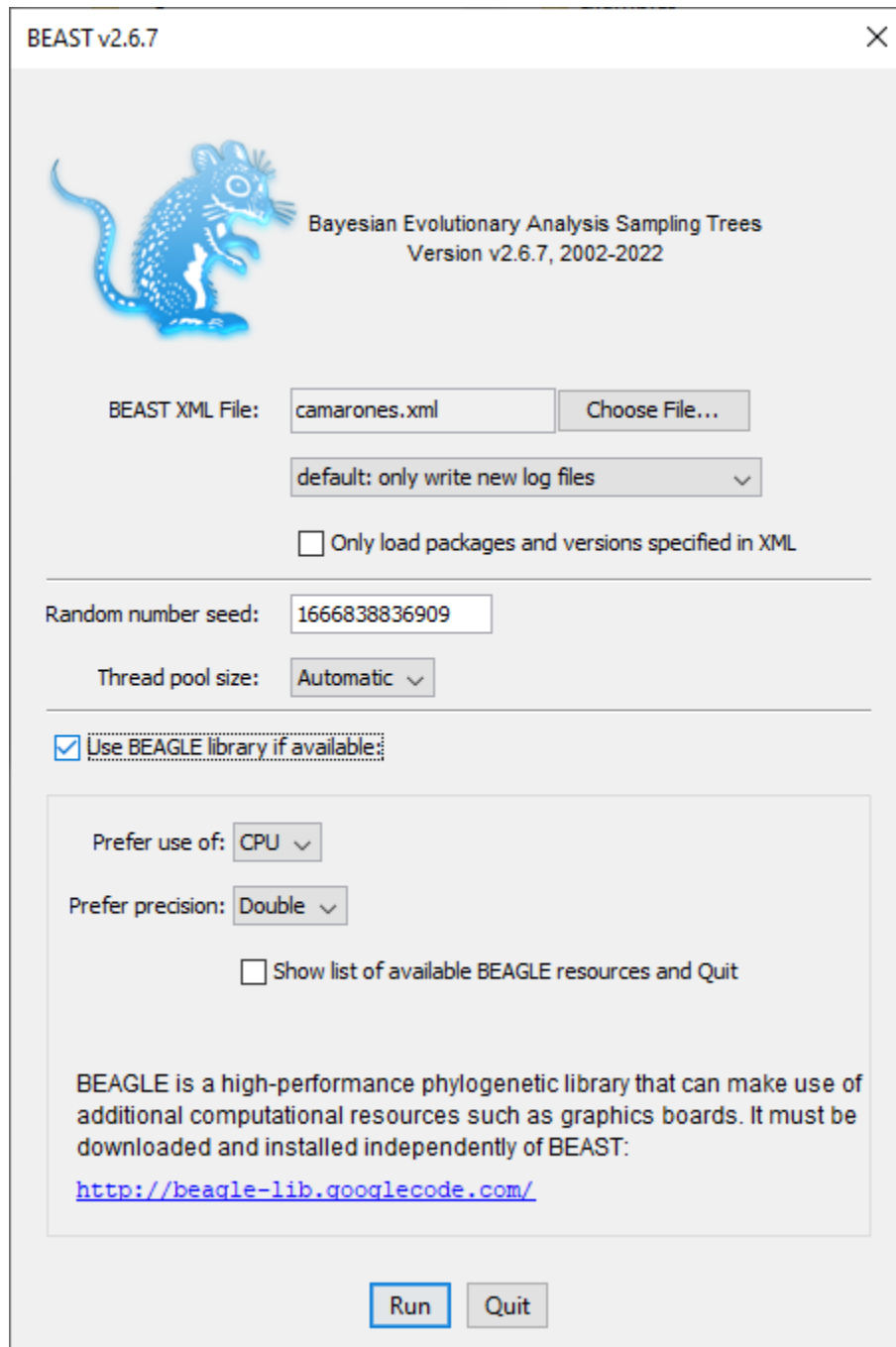


Figure 10: Ejecutando el programa BEAST2

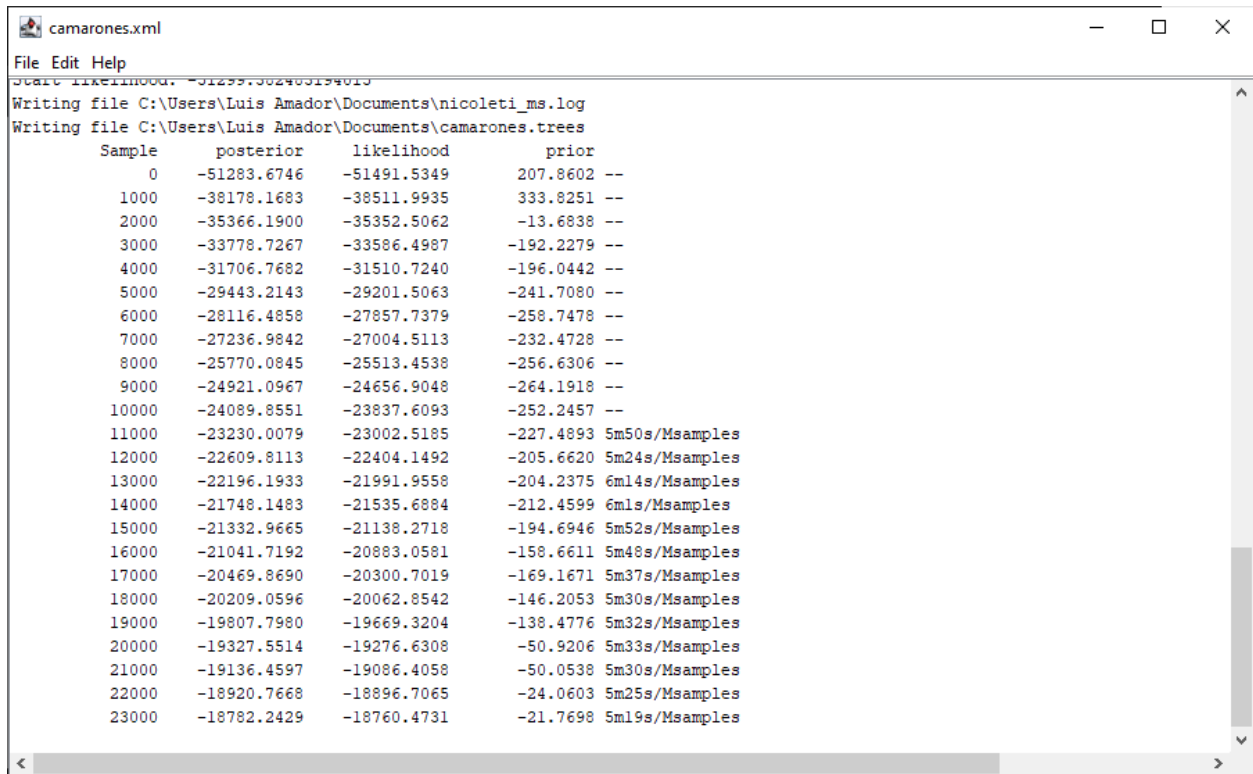


Figure 11: Ejecutando el programa BEAST2

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 específicamente 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 nuestro 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
```

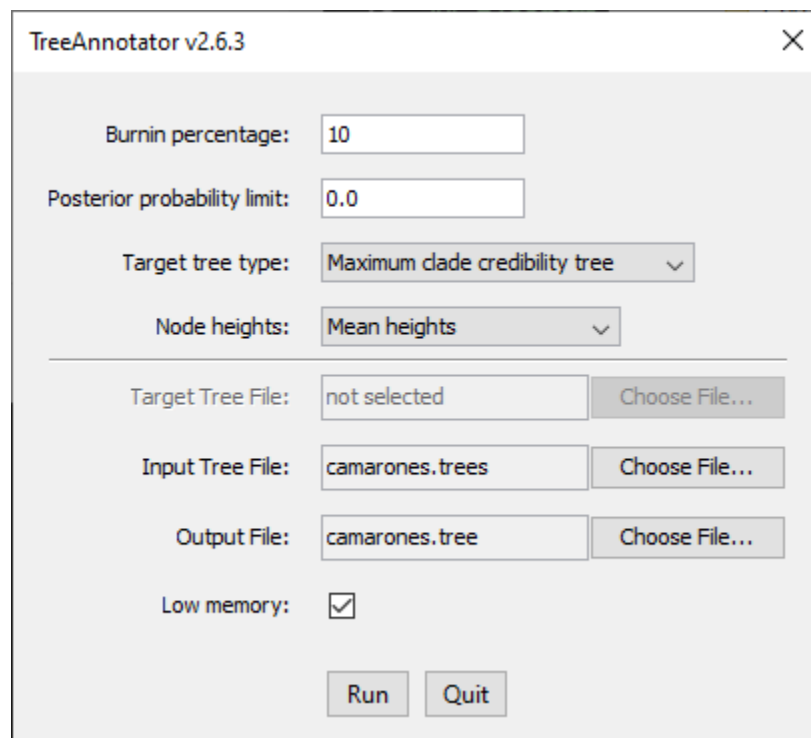


Figure 12: Obteniendo nuestro árbol bayesiano ejecutando TreeAnnotator

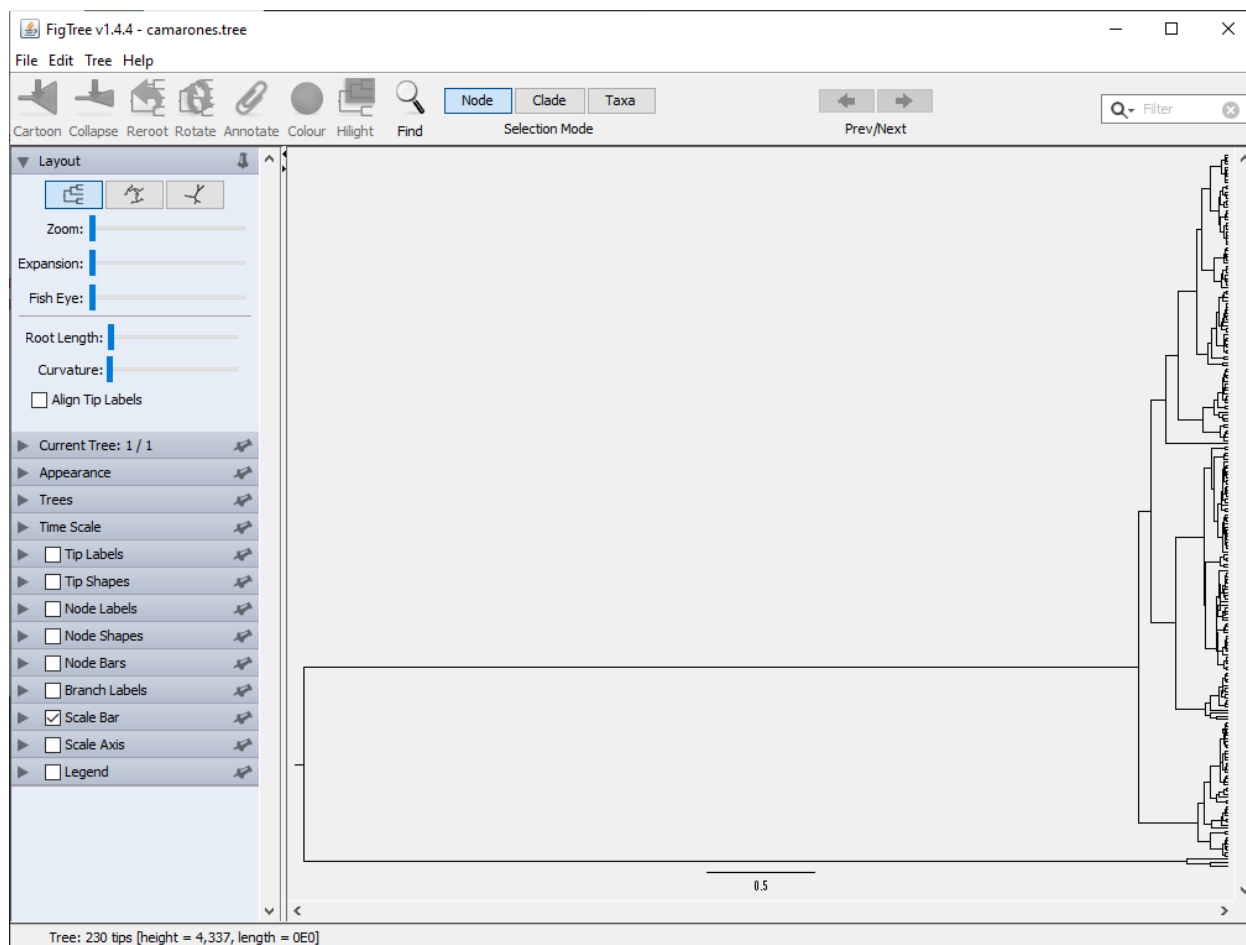


Figure 13: Visualización del árbol con Figtree