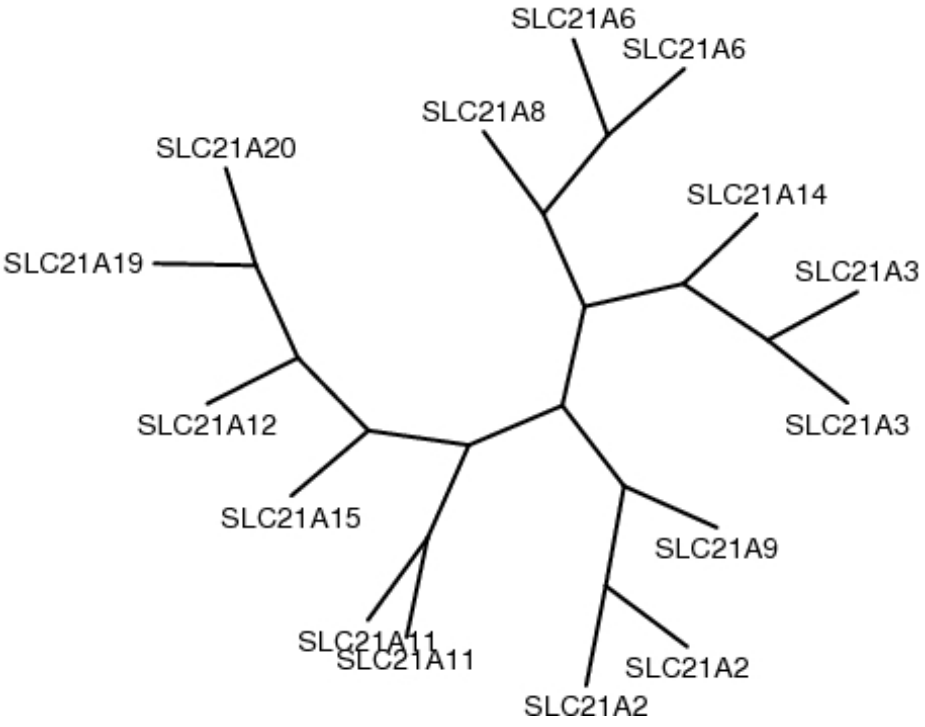
Práctica 2: Reconstrucción filogenética avanzada

Enrique Sapena Ventura

22
Abril
2018



Índice

1. Introducción

Continuando con la práctica anterior, en esta práctica se pretende emplear dos métodos adicionales para la reconstrucción de la historia filogenética de un grupo de bacterias en base a una región de un gen de la β-lactamasa, un gen de resistencia frente a antibióticos como penicilinas y cefalosporinas. Los dos métodos estadísticos de elección son:

1. ***Maximum Likelihood (ML)***→ Modelo estadístico paramétrico basado en la estimación de las distancias de una muestra de n observaciones independientes e idénticamente distribuidas. Existen infinidad de aproximaciones para aplicar este modelo estadístico, pero en esta práctica nos centraremos en el proporcionado por el programa *"PhyML"*, concretamente la versión 3.0, el cual utiliza algoritmos muy eficientes para la medición de las distancias [2].

2. ***Inferencia Bayesiana***→ Este método, para el cual usaremos el programa **MrBayes**, se basa en la probabilidad *a posteriori* de la distribución de árboles.

Se realizará un análisis mediante ambos métodos, y los árboles obtenidos se compararán, decidiendo razonadamente entre uno de los tres árboles.

2. Método de máxima verosimilitud

Primero, convertimos el alineamiento en formato "fasta" de la anterior práctica a formato "Phylips", que es el que acepta el programa. Este formato tiene dos versiones: secuencial e intercalada. En nuestro caso, puesto que los nombres pueden contener más de 10 caracteres, procederemos a formatear nuestro alineamiento en la versión secuencial (La intercalada trunca el nombre)[1]. Una vez obtenido esto, procederemos a emplear el programa para realizar el análisis filogenético y el árbol correspondiente. Lanzamos el programa por línea de comandos; se nos pide el nombre del archivo, se lo proporcionamos y ésto nos llevará a la siguiente pantalla, de ajuste de parámetros. En la primera instancia de esta pantalla se especifican datos del input; cambiamos el tipo de input a secuencial y pasamos a la siguiente pantalla introduciendo "+". Entramos en la pantalla de selección de modelo de sustitución. En la anterior práctica, habíamos determinado que el mejor modelo para nuestros datos era Kimura de dos parámetros con un parámetro $\gamma = 0,27$. Este modelo está representado en el programa como "K80" (Kimura 1980), y es un modelo que distingue entre **transiciones**[2] y **transversiones**[3]. Ajustamos el valor γ al proporcionado, pedimos una estimación de los sitios invariantes y la tasa de transiciones y transversiones, y asignamos el valor 8 al número de categorías de tasas relativas de evolución.

[1] Para el formateo, se ha empleado la siguiente herramienta web: http://phylogeny.lirmm.fr/phylo_cgi/data_converter.cgi

[2] A⇌G — C⇌T

[3] A⇌C/T — G⇌C/T

Se ha escogido este número en base al empleo del programa con un tiempo de ejecución asequible.

Pasamos al siguiente menú, el cual corresponde al menú de búsqueda de árboles. Aquí, escogeremos optimizar la topología del árbol, puesto que es nuestra primera ejecución sobre estos datos. El algoritmo a emplear para la creación del arbol será **"BioNJ"**(Bio-Neighbour-Joining), un algoritmo mejorado de NJ que emplea PhyML, y que aunque emplea el mismo procedimiento básico(Agrupamiento de manera iterativa, creación de un nodo y reducción de la matriz de distancias), pero mejorado mediante la adición de un modelo de varianzas y covarianzas de primer orden de estimadores de la distancia evolutiva.[1]

El siguiente parámetro sirve para escoger entre las operaciones de búsqueda para la topología del árbol; Como usamos una aproximación heurística de búsqueda, hay 3 opciones: NNI (Nearest Neighbour Interchange) y SPR (Subtree Pruning and Regrafting), o combinado. Para nuestro propósito, escogeremos NNI, ya que aunque es un poco menos exacto que SPR, la velocidad de ejecución es mucho mayor.

Por último, entramos al menú de apoyo de las ramas. Para esta práctica, no se realizarán análisis de bootstrap, y la otra opción se dejará por defecto también(Apoyo SH-like). Se corre el programa, y se obtendrán una serie de archivos: uno que contendrá los detalles del análisis ("Stats") y otro que contendrá el arbol, en formato Newick("tree"). Estos archivos serán incluidos con la práctica. El archivo de Stats, entre otros, contiene la semilla del análisis, por si se requiriese una repetición exacta.

3. Inferencia Bayesiana

Primero, convertimos el alineamiento de formato "fasta."[a] formato "nexus", ya que es el formato aceptado por el programa "MrBayes". Una vez convertido a este formato, se añade el siguiente código al final del archivo:

```
begin mrbayes;
lset nst=2  rates=invgamma  ;
set autoclose=no;
mcmc ngen=1000000 printfreq=1000 samplefreq=100 nchains=4;
[sumt outputname=ortologos_sumt.nex burnin=500 contype=halfcompat;]
[sump outputname=ortologos_sump.txt burnin=500 hpd=yes;]
end;
```

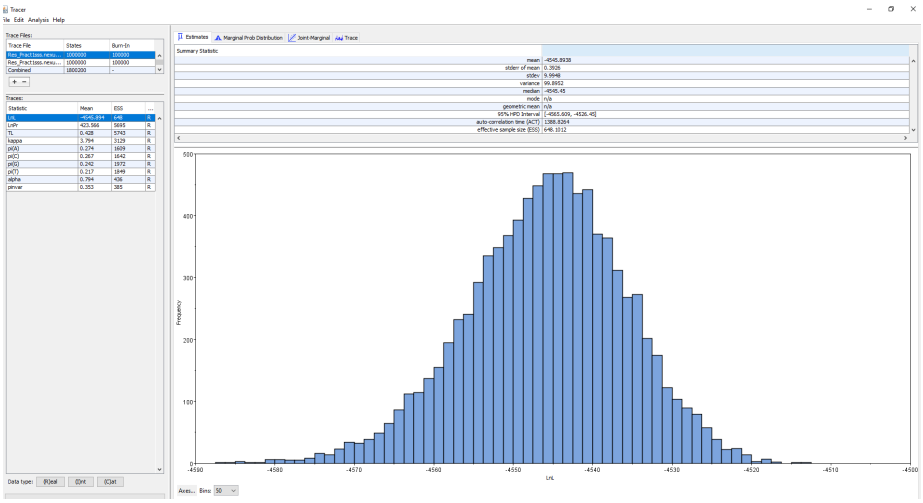
Estas líneas de código al final del archivo le dirán a MrBayes los parámetros a emplear en el análisis. Se emplearán los parámetros del ejemplo proporcionado durante la realización de la práctica 2, con la excepción del parámetro "nst", al cual daremos un valor de 2, ya que corresponde a nuestro modelo de sustitución, Kimura.

El programa realizará el análisis con 1000000 de generaciones. Una vez terminado, nos dará una media para el valor de desviación estándar de las frecuencias

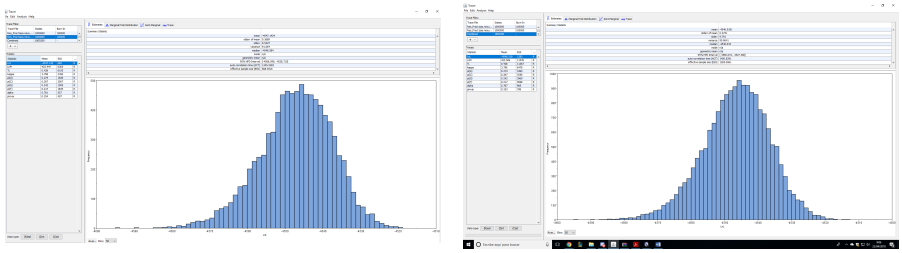
divididas. Este valor, en nuestro caso, es 0.015356. Lo consideramos lo suficientemente bajo, así que le diremos que termine el análisis.

3.1. Tracer

Una vez obtenidos los resultados, nos daremos cuenta de que hay dos carreras, y dos ficheros de árboles. Para decidir qué datos emplear, necesitamos conocer las convergencias de los parámetros de las carreras, tanto por separado como combinadas. Para ello, nos serviremos del programa "Tracer". Cargamos los dos archivos ".p" en el programa, y nos saldrán las siguientes pantallas:



(a)



(b) (c)

Figura 1: Resultados de las carreras con el programa MrBayes, representadas en Tracer. La subfigura A representa la primera carrera, la B la segunda, y la C la combinación de ambas.

De estos valores, nos interesa fijarnos en logaritmo neperiano de la verosimilitud (*LNL*), cuya media es -4545.894. En esta fila, nos fijaremos en el *Effective Sample Size*(ESS)[4], que es el número de valores independientes de la probabilidad posterior. Este valor nos da una idea del balance entre la longitud de la cadena y como de frecuente se ha muestreado la posterior durante la carrera. Por lo general, un valor ideal para este valor debería ser 200, y puesto que en nuestra primera carrera hemos obtenido un valor de 648, emplearemos esta carrera para representar el árbol. Para ello, emplearemos el programa "TreeAnnotator", dentro del paquete de programas "Beast". Corremos el programa, indicándole el número de "Burn-In" (100000). Esto nos dará un árbol, que visualizaremos mediante el programa "FigTree", y exportaremos en formato png.

[4]Definido como→ $n_{efectiva} = \frac{n}{1+(n-1)\rho}$

4. Arboles, resultados y discusión

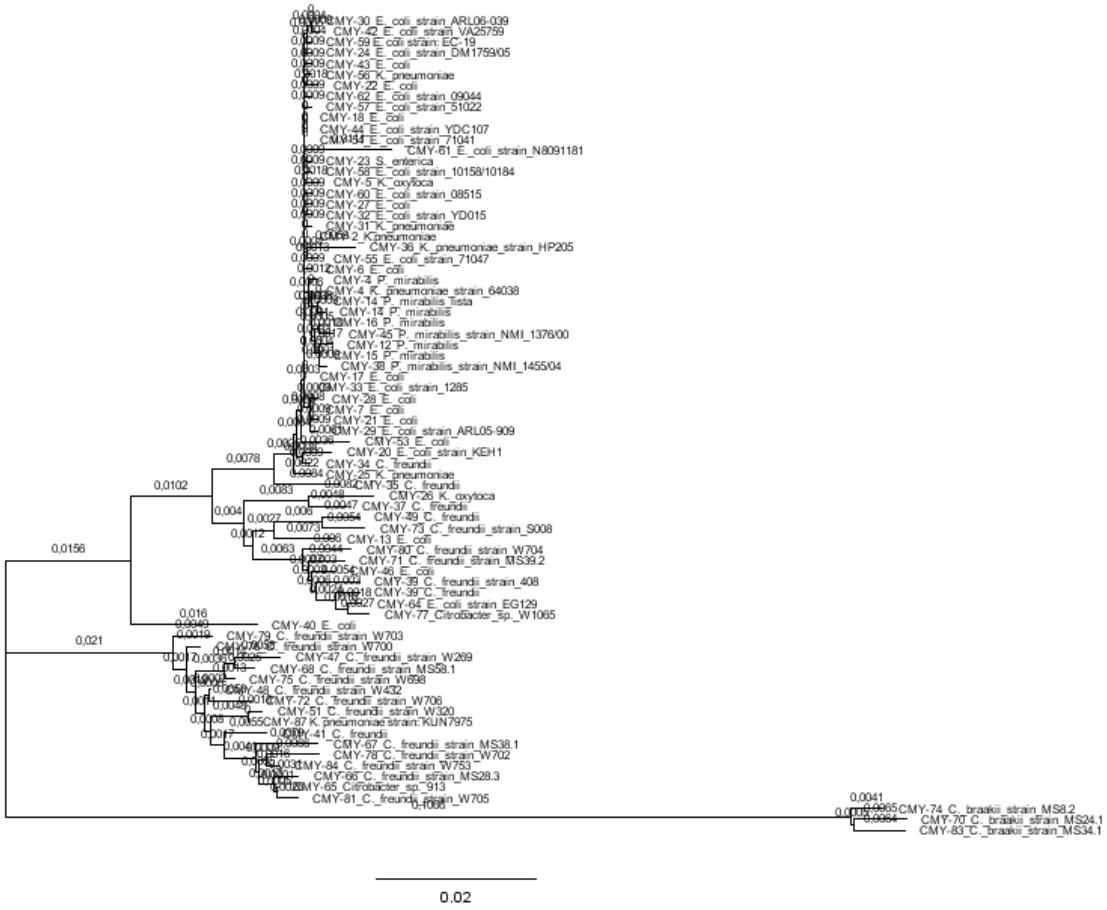


Figura 2: Arbol obtenido mediante N-J en MEGA

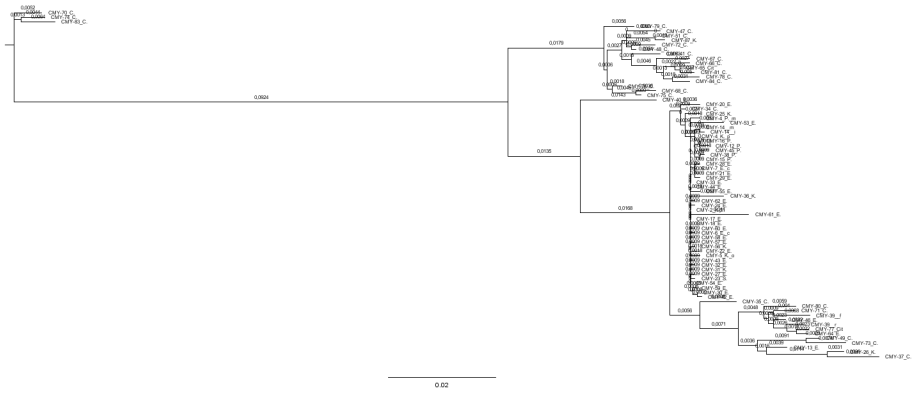


Figura 3: Arbol obtenido mediante PhyML

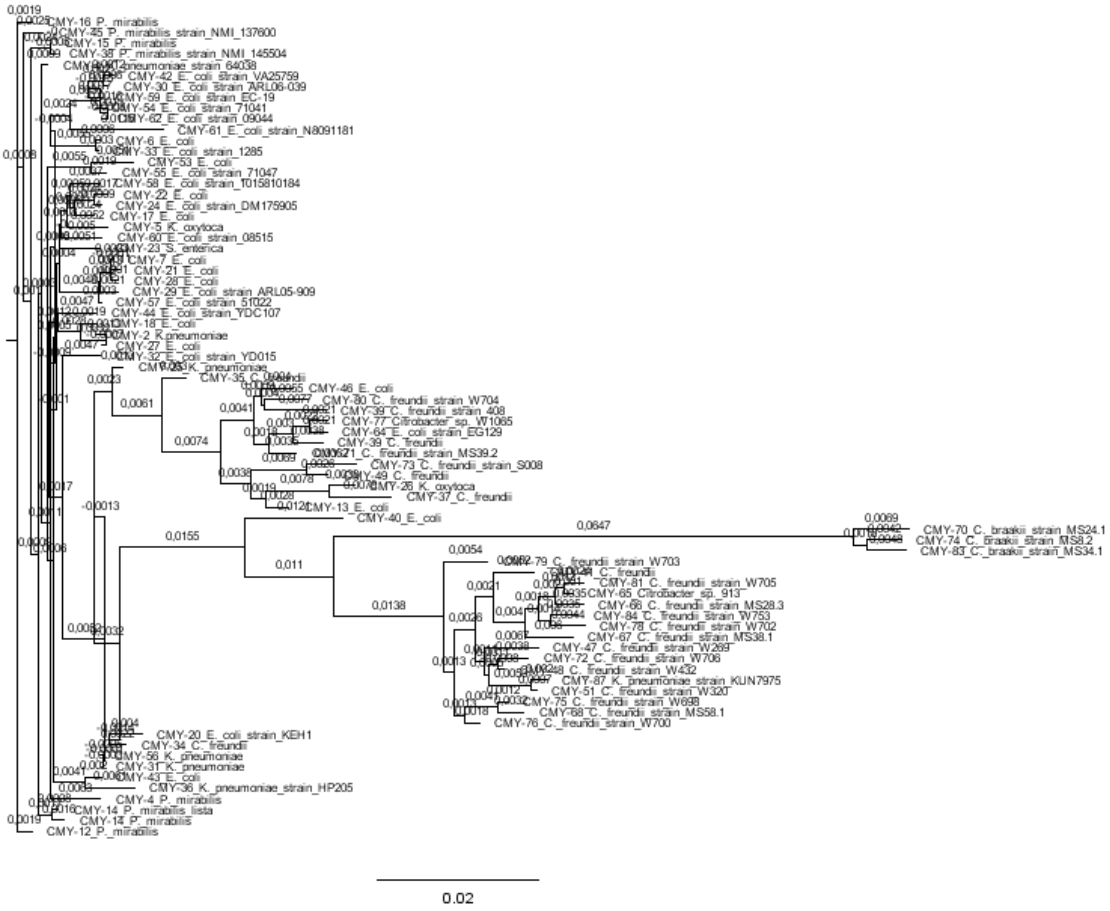


Figura 4: Arbol obtenido mediante el modelo de Cadenas de Markov-MonteCarlo (MCMC) en MrBayes

En los resultados, aunque no se puedan observar claramente (Se han adjuntado tanto los archivos para la visualización en FigTree como una versión en imagen, exportada a png), se puede ver que *C. brakii* es el grupo outlier en 2 de los tres árboles. En el tercero (El obtenido mediante inferencia Bayesiana) se puede observar una rama con una distancia considerable con respecto a su cluster "vecino", el cual corresponde al cluster de *C. brakii*. Sin embargo, basándonos en los datos, descartaremos este árbol, ya que hay una cierta evidencia de que *C. brakii* sea un grupo outlier, basándonos en los otros 2 árboles.

Basándonos en el tipo de gen que es (β-Lactamasa, un gen de resistencia que se puede presentar tanto en el genoma bacteriano como en plásmidos), y en el tipo de algoritmo, yo escogería el árbol obtenido mediante el método de máxima verosimilitud proporcionado por PhyML. Es el que más se adecúa a la agrupación en nodos por especies, con valores decrecientes de bootstrap conforme nos acercamos a las hojas de los nodos (Tal y como es de esperar, puesto que secuencias cercanas compartirán una mayor similitud de secuencia). Sin embargo, como ya se ha comentado, este gen es un gen complicado para la reconstrucción de relaciones filogenéticas entre taxones puesto que, al estar presente en algunas ocasiones dentro de plasmidos, hay que tener en cuenta tanto eventos de transferencia horizontal entre especies, como eventos de transformación, además de una presión evolutiva muy distinta entre las cepas, dependiendo de si han sido expuestas a antibióticos con anillos β-lactámicos (Entre otros eventos).

En conclusión, no es fácil escoger el mejor método, pero es necesario, puesto que las características de nuestras secuencias, el algoritmo/programa de elección y los parámetros que escojamos para nuestro análisis determinarán, en última instancia, si nuestros resultados son fiables o un "Byproduct" de no seguir una metodología adecuada.

Referencias

[1] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.*, 14(7):685–695, Jul 1997.

[2] S. Guindon, J. F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.*, 59(3):307–321, May 2010.