ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

SIMULACIÓN COMPUTACIONAL DE MODELOS PROBABILÍSTICOS DE EVOLUCIÓN DEL ADN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA MATEMÁTICA

PROYECTO DE INVESTIGACIÓN

RAQUEL ESTHELA VARGAS VIVANCO

raquel.vargas@epn.edu.ec

DIRECTORA: ADRIANA UQUILLAS ANDRADE PhD.

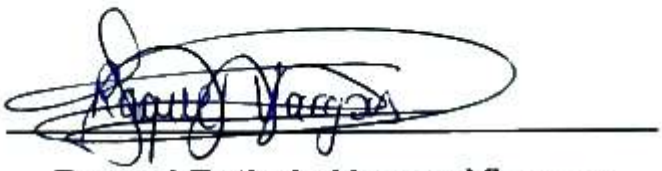
adriana.uquillas@epn.edu.ec

Quito, junio 2018

DECLARACIÓN

Yo, Raquel Esthela Vargas Vivanco, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por su normativa institucional vigente.



Raquel Esthela Vargas Vivanco

CERTIFICACIÓN

Certifico que el presente trabajo de titulación fue desarrollado por Raquel Esthela Vargas Vivanco, bajo mi supervisión.

Adriana Uquillas Andrade

DIRECTORA

Agradecimientos

Quiero agradecer a mis padres por brindarme el apoyo moral y económico durante mis estudios de pregrado y a mis tutores por guiarme, corregirme y aconsejarme durante el desarrollo de este trabajo.

Además, quiero agradecer de manera muy especial al Centro de Modelización Matemática MODEMAT, sobre todo al Dr. Luis Miguel Torres por proporcionarme las facilidades para realizar este trabajo de titulación.

A mi madre.

Contenido

Lista de figuras

Lista de tablas

1 Introducción

Los *árboles filogenéticos* o *filogenias* son diagramas que representan las relaciones evolutivas entre organismos vivos. Esta es precisamente su principal utilidad, determinar la historia evolutiva entre los seres vivos. Por eso, también son ampliamente utilizados en distintas ramas de la Biología y Medicina.

A manera de ejemplo, citemos el descubrimiento de un compuesto que se utiliza en el tratamiento de algunos tipos de cáncer y que se expende con el nombre comercial de *Taxol*.

Científicos descubrieron que los árboles de tejo del Pacífico producen Taxol, sin embargo resultaba difícil y costoso obtener una cantidad suficiente de este compuesto a partir de este tipo de árbol como para que pueda usarse ampliamente. No obstante, basados en las relaciones evolutivas, los biólogos esperaban que especies cercanamente relacionadas con el tejo de Pacífico produzcan compuestos similarmente efectivos.

Afortunadamente estaban en lo correcto, se descubrió que las hojas del tejo europeo podían usarse para producir Taxol de manera más eficiente. [14]

El proceso de obtener filogenias a partir de información obtenida de un grupo de seres vivos es llamado *Reconstrucción de Árboles Filogenéticos* o *Reconstrucción Filogenética* y la información con la que se trabaja son principalmente secuencias de ADN o de proteínas.

Existen varios métodos para hacer Reconstrucción Filogenética, por ejemplo: *Máxima Parsimonia, Neighboor Joining, Máxima Verosimilitud, Inferencia Bayesiana, etc.*

Por otro lado, el presente trabajo se centra en el estudio de los *Modelos Probabilísticos de Evolución del ADN* o Modelos Evolutivos, los cuales son la base de los métodos estadísticos de Reconstrucción Filogenética como la inferencia por Máxima Verosimilitud y la inferencia Bayesiana.

Bajo ciertas suposiciones, como se explicará en el Capítulo (2), la evolución del ADN puede modelarse por medio de una Cadena de Markov homogénea en tiempo continuo; procesos estocásticos que se estudiarán en la sección (1.4).

El propósito de este estudio es crear una herramienta computacional con la cual, a partir de un árbol filogenético y un modelo evolutivo dados,
simular secuencias de ADN. Esta tarea puede ser vista como el proceso inverso a la Reconstrucción, en donde se parte de un grupo de secuencias de ADN para obtener el árbol y los parámetros numéricos que conforman el modelo evolutivo.

Entonces, se va a simular datos para distintos árboles y distintos modelos evolutivos. Posteriormente, como se indica en el Capítulo 6, sobre estos

datos simulados se realizarán reconstrucciones con el método de Máxima Verosimilitud para analizar los parámetros y topologías estimados a partir de los datos simulados.

Para alcanzar este objetivo, se estudiará las estructuras matemáticas en las que se sustentan estos Modelos Evolutivos del ADN y sus propiedades.

Además, se revisará las propiedades de los Estimadores de Máxima Verosimilitud para poder definir el método de inferencia de árboles filogenéticos por Máxima Verosimilitud.

Para desarrollar la herramienta computacional de simulación de secuencias de ADN, se hará uso de las facilidades que ofrece el lenguaje de programación R y como un aporte extra, se presentará una aplicación web que permite graficar, modificar y descargar las imágenes de árboles filogenéticos en distintos formatos. Aplicación que también ha sido creada utilizando librerías de R.

1.1 Descripción del Problema

Primero veamos en qué consiste el problema de la reconstrucción de árboles filogenéticos. Este puede describirse de la siguiente manera:

Se tiene información de un conjunto de seres vivos: animales, bacterias, plantas, etc. Información representada por medio de una matriz de secuencias de ADN.
La tarea consiste en encontrar las *relaciones evolutivas* entre estos seres vivos, utilizando algún método de inferencia.

Estas relaciones evolutivas se representan por medio de un diagrama parecido a un árbol cuyas hojas representan las especies de seres vivos de los cuales se extrajo la información.

Los nodos padre o *internos*, representan a los ancestros extintos que tenían en común los seres vivos estudiados.

Para ilustrar esta idea, considérense a las especies del género *Canis*:

- **Perro doméstico** *Canis lupus familiaris*
- **Lobo común** *Canis lupus lupus*
- **Coyote** *Canis latrans*

Las relaciones evolutivas entre estas tres especies de caninos se representan en el árbol de la figura (1.1).

El nodo (1), en este ejemplo, representa al ancestro que el perro y el lobo tenían en común, mientras que el nodo (2) representa el ancestro que tenían en común el coyote y el nodo (1).

La pregunta ahora es:

¿ Cómo podemos reconstruir este árbol a partir de la información que se tiene de las especies actuales?

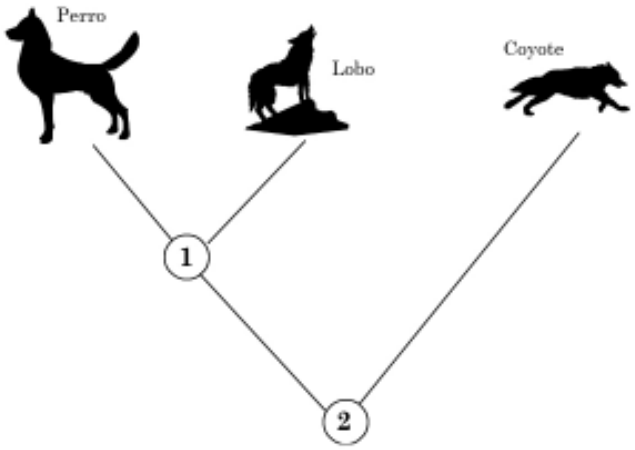


FIGURA 1.1: Relaciones evolutivas entre el perro, lobo y coyote

Existen muchos enfoques para resolver este problema, como se indicó en la parte introductoria de este capítulo, pero en el presente trabajo se utilizarán métodos de inferencia estadística como los métodos de Máxima Verosimilitud o Inferencia Bayesiana.

Por otro lado, el problema que se aborda en este trabajo es el opuesto al de la reconstrucción de árboles filogenéticos. En este caso partimos de un árbol filogenético fijo y lo que queremos es recuperar la matriz de secuencias de ADN cuya evolución ha ocurrido de acuerdo al árbol filogenético que tenemos como dato.

Como se explica en el capítulo 2, la evolución en las ramas de un árbol filogenético se modela por medio de una cadena de Markov homogénea en tiempo continuo y como se detalla en la sección (1.4), estas cadenas de Markov pueden representarse por medio de una matriz generador infinitesimal Q y de un vector de frecuencias estacionarias π.

Por lo tanto, utilizando técnicas de simulación estocástica, se puede obtener un conjunto de secuencias de ADN a partir de un árbol filogenético cuyas ramas han evolucionado de acuerdo a una cadena de Markov homogénea en tiempo continuo con generador Q y vector de distribución estacionaria π.

En la siguiente sección se explica brevemente qué es el ADN y como se lo representa.

1.2 Ácido Desoxirribonucleico

El ADN es un ácido que se encuentra en el núcleo de la mayoría de organismos vivos y se compone de las siguientes moléculas:

- Deoxyribosa
- Fosfato

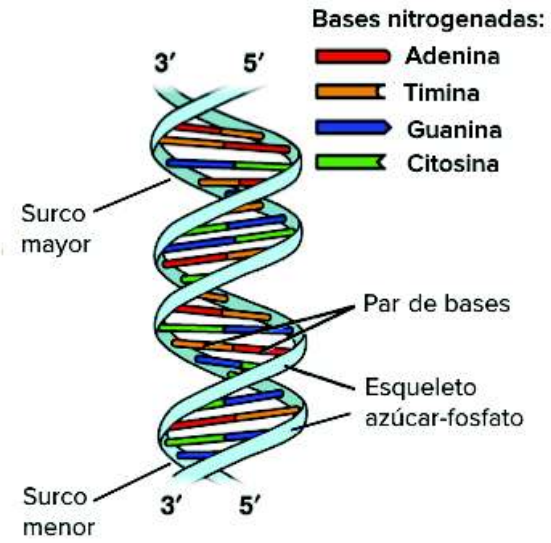


FIGURA 1.2: Estructura de doble hélice
Fuente: es.khanacademy.org

- Cuatro bases nitrogenadas:
  - Adenina, abreviado como A
  - Guanina, abreviado como G
  - Citocina, abreviado como C
  - Timina, abreviado como T

Según la *U.S. National Library of Medicine*, el ADN guarda información en forma de código compuesto de las cuatro bases nitrogenadas similar al orden en el que aparecen las letras del alfabeto para formar palabras.

Las bases nitrogenadas se juntan en parejas de la siguiente forma:

A con **T**
G con **C**

En otras palabras, cada una de las bases tiene una base complementaria a la cual se junta por medio de enlaces de puentes de hidrógeno

Estos emparejamientos hacen que la estructura del ADN tenga forma de escalera en espiral o también conocida como doble hélice.

Estas cuatro bases pueden ser agrupadas en:

1. **Purinas**: Adenina y Guanina.
2. **Pirimidinas**: Citocina y Timina.

1.2.1 Mutaciones

Durante el proceso de multiplicación del ADN, la estructura de doble hélice del ADN se rompe y se genera dos hebras "sueltas" las cuales, posteriormente se convertirán en dos estructuras de doble hélice simplemente reuniendo las

bases nitrogenadas complementarias como se explicó en la sección anterior. Este proceso natural de multiplicación del ADN, se ejecuta de tal forma que pocos errores se cometan en el producto [3]; es decir, cuando los padres heredan su material genético a sus descendientes, el proceso de multiplicación celular se ejecuta de tal forma que se comentan el menor número de errores que puedan afectar a las crías. Considerando lo anterior, se puede decir que una *mutación* es simplemente un cambio en el ADN.

Para que exista evolución biológica, debe haber variación genética. Por lo tanto, se puede definir a la *evolución* como el cambio en al composición genética de una población luego de sucesivas generaciones, la cual puede se causada por selección natural, hibridación, endogamia o mutación.

Existen diferentes tipos de mutaciones así como causas para estas mutaciones. En el presente trabajo se considerará únicamente las mutaciones conocidas como *sustituciones*. Estas mutaciones son simplemente el reemplazo de una base por otra en un determinado sitio de la secuencia de ADN.

Por ejemplo, supongamos que las siguientes son partes de las secuencias de ADN de un padre y un hijo.

$$\begin{array}{rc} \text{Padre:} & \ldots \text{CTGGAG} \ldots \\ \text{Hijo:} & \ldots \text{CTGGGG} \ldots \\ & \quad\; i \end{array}$$

Entonces, se dice que ocurrió un sustitución en la posición i. Esta sustitución fue de la base A a la base G.

A su vez, existen dos tipos de sustituciones:

1. **Transiciones.-** Cuando una purina es reemplazada por otra purina o una pirimidina es reemplazada por otra pirimidina, es decir, los cambios del tipo:
$$A \iff G \text{ ó } C \iff T$$

2. **Transversiones.-** Cuando una purina es reemplazada por una pirimidina y viceversa. Es decir, los siguientes cambios:
$$A \iff C,\ G \iff C,\ A \iff T \text{ ó } G \iff T$$

Como ya se mencionó, existen otros tipos de mutaciones, por ejemplo inserciones, eliminaciones, etcétera; pero solo se considerarán las sustituciones con el fin de lograr que la modelación del proceso evolutivo sea más clara y matemáticamente tratable. [3]

1.3 Topología de árboles filogenéticos y árboles métricos

En la sección anterior se planteó la idea de árbol filogenético. En esta sección se introducirá la terminología que va a utilizarse en el presente trabajo.

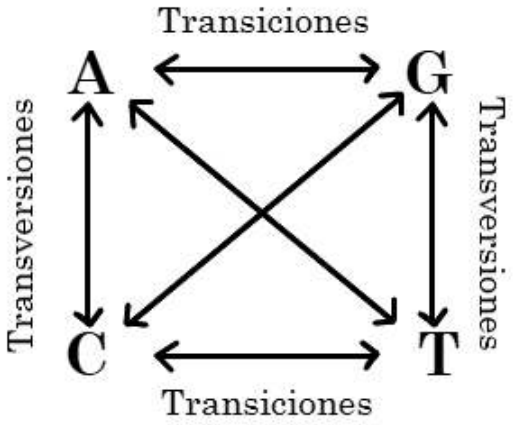


FIGURA 1.3: Tipos de sustituciones

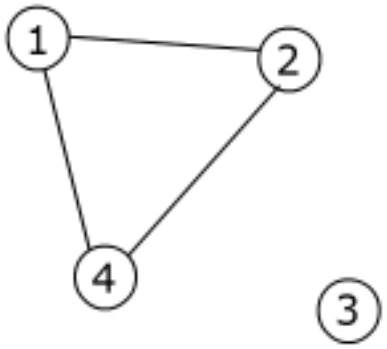


FIGURA 1.4: Ejemplo de grafo

La definición de "árbol" en Análisis Filogenético es la misma que se le asigna a este concepto en Teoría de Grafos, siendo el árbol un tipo particular de grafo que tiene ciertas características.

***Definición* 1.1** (Grafo)**.** *Un* grafo G *es un par ordenado* $G = (V, E)$*, compuesto por un conjunto finito* V *cuyos elementos son llamados* vértices *o* nodos *y un conjunto* E *de pares ordenados o no ordenados de* V*. Si los elementos de* E *son pares ordenados, estos son llamados arcos y el grafo* G *es llamado* grafo dirigido*; por otro lado, si los elementos de* E *son pares no ordenados, estos son llamados aristas y* G *es llamado* grafo no dirigido*.*

Ejemplo 1.1.- El grafo $G = (V, E)$ con $V = \{1, 2, 3, 4\}$ y $E = \{\{1, 2\}, \{1, 4\}\{2, 4\}\}$, está representado gráficamente en la figura (1.4)

***Definición* 1.2** (Camino)**.** *Dado un grafo* $G = (V, E)$*, un* camino *del vértice* s *al vértice* t *con* $s, t \in V$*, es una sucesión de aristas* $e_1, e_2, e_3, \ldots, e_n \in E$ *tales que:*

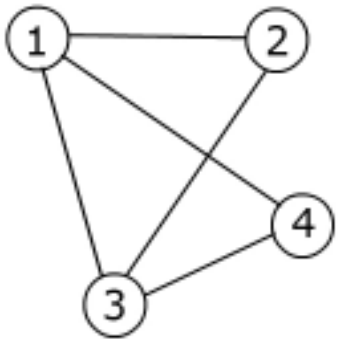


FIGURA 1.5: Ejemplo de grafo conexo.

1. *el extremo inicial de la primera arista e_1 es s,*
2. *el extremo final de la última arista e_n es t y*
3. *el extremo final de la arista e_i es el extremo inicial de la arista e_{i+1}, para todo $i \in \{1, 2, \ldots, n\}$*

***Definición* 1.3** (Grafo conexo)**.** *Un grafo $G = (V, E)$ se dice* conexo*, si para todo par de vértices $i, j \in V$ con $i \neq j$, existe un camino que conecta i con j que no repite vértices.*

***Definición* 1.4** (Ciclo)**.** *Dado un grafo $G = (V, E)$, un* ciclo *sobre G es una sucesión de aristas $e_1, e_2, e_3, \ldots, e_n \in E$ donde el extremo final de e_i es el extremo inicial de e_{i+1}, con $i \in \{1, 2, \ldots, n\}$ y el extremo inicial de e_1 coincide con el extremo final de e_n.*

Ejemplo 1.2.- Considerando el grafo del ejemplo (1.1):

$$\{1, 2\}, \{2, 4\} \quad \text{es un camino.}$$

$$\{1, 2\}, \{2, 4\}, \{4, 1\} \quad \text{es un ciclo.}$$

El grafo $G = (V, E)$, con $V = \{1, 2, 3, 4\}$ y $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 4\}, \{3, 4\}\}$ es un grafo conexo y en la figura (1.5) está su representación gráfica.

***Definición* 1.5** (Árbol)**.** *Un* árbol *$T = (V, E)$ es un grafo conexo que no contiene ciclos.*

La definición (1.5) es la que se le asigna al concepto de árbol en Matemáticas, específicamente en Teoría de Grafos. Antes de dar la definición de *árbol filogenético*, se presentará la clasificación de los nodos en un árbol y la definición de *árbol binario.*

***Definición* 1.6** (Nodos vecinos)**.** *Sean $G = (V, E)$ un grafo $s \in V$ y $v \in V$ dos nodos de G. Se dice que s y v son* vecinos*, si existe una arista $e \in E$ que los conecta; es decir, si $s, v \in e$.*

FIGURA 1.6: Ejemplo de árbol

***Definición* 1.7** (Grado de un nodo)**.** *Sea $G = (V, E)$ un grafo. El* grado de un nodo *$s \in V$, denotado $\delta(s)$, es la cardinalidad del conjunto de vecinos del nodo s, denotado N_s, donde $N_s = \{v \in V : \text{ } v \text{ es vecino de } s\}$.*

Dadas estas dos definiciones, se puede clasificar a los nodos de un árbol en nodos hoja y nodos internos.

***Definición* 1.8.** *Sea $T = (V, E)$ un árbol y $v \in V$ un nodo de T. Se dice que v es un* nodo interno *si el grado de v, $\delta(v) \geq 2$. Si el grado de v es $\delta(v) = 1$, entonces v el llamado un* nodo hoja*.*

En el caso de los árboles filogenéticos, las "hojas" de árbol representan los diferentes individuos, especies, géneros, poblaciones, etcétera, generalmente vivos y de los cuales se puede extraer información genética que permite hacer la inferencia de árbol filogenético.

Se utilizará la palabra *taxón* o *taxon* (pl. taxones o táxones) para denominar de manera general a los organismo o grupos de organismos de los cuales se extrae la información genética. También se suele utilizar el término "unidad taxonómica operacional", abreviado OTU por sus siglas en inglés [*Operational Taxonomic Unit*]. En el presente texto se utilizarán los términos "taxón" en singular y "taxones" en plural para referirse a los organismos que representan las hojas del árbol filogenético.

***Definición* 1.9** (Árbol binario)**.** *Un árbol $T = (V, E)$ se dice* binario*, si cada nodo interno de T tiene grado 3.*

En Análisis Filogenético, a los árboles binarios también se los suele llamar *estrictamente bifurcados*. Esto significa que durante el proceso evolutivo cualquier nodo interno del árbol dio lugar a únicamente dos *linajes* u hojas.

Si en un árbol filogenético existen nodos internos con grado mayor que 3, entonces se dice que este árbol tiene *politomías*.

***Definición* 1.10.** *Sea $\mathcal{X}$ un conjunto finito de taxones o etiquetas. Entonces un $\mathcal{X}$–*árbol filogenético *está compuesto de un árbol $T = (V, E)$ y una función biyectiva $\phi : \mathcal{X} \longrightarrow L \subset V$, donde L denota el conjunto de hojas de T. La función ϕ es llamada* función etiquetadora *y el $\mathcal{X}$–árbol también es conocido como* árbol con hojas etiquetadas*.*

En palabras más simples, la función ϕ asigna una etiqueta o un nombre a cada uno de los taxones u hojas del árbol T.

***Observación* 1.1.** Es importante indicar que funciones etiquetadoras diferentes, usualmente producen árboles distintos. Esto permite distinguir dos árboles por medio de su función ϕ en lugar de simplemente observar su forma.

Las siguientes definiciones de grado entrante y saliente de un nodo son importantes para luego poder definir "árbol enraizado" y "árbol no enraizado".

***Definición* 1.11.** *Sea $D = (V, A)$ un grafo dirigido y $s \in V$ un nodo de D.*

- *El* grado entrante *de s, denotado por $\delta^+(s)$, es el número de arcos incidentes a s; esto es, el número de arcos que tienen como extremo final al nodo s.*
- *El* grado saliente *de s, denotado por $\delta^-(s)$, es el número de arcos salientes del nodo s; esto es, el número de arcos que tienen como extremo inicial al nodo s.*

Antes de definir lo que es un árbol enraizado, se dará la definición de *árbol métrico*.

La definición que se presentan a continuación está relacionada con el concepto de "largos de ramas", los cuales usualmente representan una medida de cuánto cambio ha ocurrido entre las secuencias de los nodos (internos y hojas) del árbol. [3].

***Definición* 1.12.** *Un* árbol métrico *(T, w) está compuesto por un árbol T (enraizado o no enraizado) y una función $w : E(T) \longrightarrow [0, \infty)$, que le asigna números no negativos a las ramas de T. El número $w(e)$ es llamado* largo *o* peso *de la rama $e \in E(T)$.*

Si no se especifican largos de ramas para un árbol T, entonces este se conoce como árbol topológico*.*

Usualmente, a la forma del árbol se le denomina la topología *del árbol.*

La función w o más específicamente, los largos de ramas, son verdaderamente importantes en el estudio de árboles filogenéticos. Estos largos representan el número esperado de mutaciones en un intervalo de tiempo, en otras palabras.

La aclaración anterior es necesaria porque en general es incorrecto asumir que los largos de las ramas de un árbol filogenético representan intervalos de tiempo. Dos ramas pueden reflejar el mismo intervalo de tiempo, pero tener diferentes cantidades esperadas de cambio evolutivo. En otras palabras, el largo de una rama puede estar compuesto por un múltiplo r_i del intervalo de tiempo t_i, donde el producto $r_i t_i$ es el largo de la rama. [5]

***Definición* 1.13** (Árbol enraizado)**.** *Un árbol $T^\rho = (V, E)$ se dice* enraizado *si cada nodo de V tiene grado entrante igual a 1, exceptuando la raíz cuyo grado entrante es 0.*

Un árbol se dice no enraizado *si no es enraizado.*

Un árbol enraizado es un árbol dirigido en el cual existe un nodo ρ a partir del cual se puede dirigir las ramas en dirección a las hojas. Biológicamente esta característica le asigna un sentido a la evolución del árbol, en donde se indica que las secuencias de los taxones han evolucionado de un mismo ancestro común representado por el nodo ρ.

1.4 Cadenas de Markov en tiempo continuo

Esta sección es necesaria para comprender mejor el capítulo siguiente que trata sobre modelos evolutivos, pues se va a modelar la evolución de un grupo de taxones por medio de una cadena de Markov en tiempo continuo (homogénea en el tiempo).

***Definición* 1.14.** *Una* cadena de Markov en tiempo continuo con conjunto finito de estados*, es un proceso estocástico $\mathbf{X} = \{X(t) : t \geq 0\}$ que toma valores en un conjunto finito S y tal que posee la propiedad de Markov; esto es, para todo $s, t \geq 0$ y para todo $i, j, \chi(u)$ elementos de S, con $0 \leq u \leq s$ se cumple que:*

$$\mathbb{P}\{X(t+s) = j | X(s) = i, X(u) = \chi(u), 0 \leq u \leq s\} = \mathbb{P}\{X(t+s) = j | X(s) = i\} \tag{1.1}$$

Los elementos de S son llamados *estados* y la propiedad de Markov se interpreta de la siguiente forma, la probabilidad condicional del futuro $X(t+s)$, dado el presente $X(s)$ y el pasado $X(u) = \chi(u), 0 \leq u \leq s$, depende solamente del presente y es independiente del pasado. [17]

***Definición* 1.15.** *Una cadena de Markov en tiempo continuo $\{X(t) : t \geq 0\}$, se dice* homogénea en el tiempo *si satisface la siguiente propiedad, para todo $s, t \geq 0$ con $t > s$*

$$\mathbb{P}\{X(t) = j | X(s) = i\} = \mathbb{P}\{X(t-s) = j | X(0) = i\} \quad \forall\, i, j \in S \tag{1.2}$$

En otras palabras, una cadena de Markov es homogénea si las probabilidades de cambiar de un estado a otro son homogéneas en el tiempo. En el presente trabajo, se estudiarán únicamente cadenas con esta propiedad.

1.4.1 Tiempos exponenciales

Sea T_i la cantidad de tiempo que la cadena pasa en el estado i antes de cambiar a otro estado. Una característica de las cadenas de Markov en tiempo continuo es que la cantidad de tiempo que el proceso pasa en el estado i está exponencialmente distribuida con media, digamos $1/v_i$, en la siguiente sección se explicará lo que representa v_i.

La propiedad de *falta de memoria* es una de las características de las variables aleatorias exponenciales.[1]

Para demostrar que T_i debe tener la propiedad de falta de memoria, supongamos que al tiempo $t \geq 0$, la cadena está en el estado i. Entonces, el evento $\{T_i \geq t\}$ es equivalente al evento $\{X(u) = i, 0 \leq u \leq t\}$ y el evento $\{T_i \geq t+\epsilon\}$ es equivalente al evento $\{X(u) = i, 0 \leq u \leq t + \epsilon\}$, con $\epsilon \geq 0$. Luego,

$$\mathbb{P}(T_i \geq t + \epsilon | T_i \geq t) = \mathbb{P}\{X(u) = i, 0 \leq u \leq t + \epsilon | X(u) = i, 0 \leq u \leq t\}$$

ahora, por la propiedad $P(A \cap B|A) = P(B|A)$, la expresión anterior es equivalente a

$$\mathbb{P}\{X(u) = i, t \leq u \leq t + \epsilon | X(u) = i, 0 \leq u \leq t\}$$

por la propiedad de Markov, la expresión anterior es igual a

$$\mathbb{P}\{X(u) = i, t \leq u \leq t + \epsilon | X(t) = i\}$$

finalmente, por homogeneidad en el tiempo se sigue que

$$\mathbb{P}(T_i \geq t + \epsilon | T_i \geq t) = \mathbb{P}\{X(u) = i, 0 \leq u \leq \epsilon | X(0) = i\} = \mathbb{P}(T_i > \epsilon)$$

Luego, T_i es una variable aleatoria continua con la propiedad de falta de memoria y por lo tanto T_i debe estar exponencialmente distribuida. [21]

Finalmente, notar que la cantidad de tiempo que el proceso pasa en el estado i y el siguiente estado visitado, digamos j, deben ser variables aleatorias independientes. Pues, si el siguiente estado visitado dependiera de T_i, entonces la información de cuanto tiempo ha pasado el proceso en i sería relevante para la predicción del siguiente estado, pero esto contradice la propiedad de Markov. [17]

1.4.2 Generador infinitesimal

Para introducir la definición de matriz generadora infinitesimal, sea

$$p_{ij}(t) = \mathbb{P}\{X(t + s) = j | X(s) = i\}$$

la *probabilidad de transición* del estado i al estado j, es decir, la probabilidad de que el proceso, presente en el estado i al tiempo s, pase al estado j luego de un tiempo t.

Para cada $i, j \in S$ se asigna un número no negativo q_{ij}, el cual puede verse como la *tasa instantánea* a la cual la cadena cambia del estado i al estado j.

Sea v_i la tasa a la cual el proceso está "saliendo" del estado i, es decir,

$$v_i = \sum_{i \neq j} q_{ij}. \tag{1.3}$$

[1] Una variable aleatoria X posee la propiedad de *falta de memoria* si para todo $s, t \geq 0$,

$$\mathbb{P}(X > s + t | X > t) = \mathbb{P}(X > s)$$

Por un lado, $1/v_i$ es precisamente la media de la variable aleatoria T_i de la sección (1.4.1).

Por otro lado, como afirma LAWLER [13, p. 69-70], las probabilidades de transición de una cadena de Markov en tiempo continuo y homogénea, pueden escribirse de la siguiente forma:

$$p_{ii}(\Delta t) = \mathbb{P}\{X(t+\Delta t) = i | X(t) = i\} = 1 - v_i \Delta t + o(\Delta t) \tag{1.4}$$

$$p_{ij}(\Delta t) = \mathbb{P}\{X(t+\Delta t) = j | X(t) = i\} = q_{ij}\Delta t + o(\Delta t) \quad i \neq j \tag{1.5}$$

donde Δt es un intervalo pequeño de tiempo y $o(\Delta t)$ es una función que es mucho más pequeña que Δt para un Δt pequeño, es decir,

$$\lim_{\Delta t \longrightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0.$$

Notar que q_{ij} también puede verse como: la tasa a la cual el proceso sale de i por la probabilidad de que ocurra una transición al estado j, esto es

$$q_{ij} = v_i p_{ij} \tag{1.6}$$

***Definición* 1.16** (Matriz de Markov)**.** *La matriz $M(t)$ de dimensión $|S|\times|S|$ cuyas entradas (i,j) son las probabilidades de transición $p_{ij}(t)$ es llamada* matriz de transición o matriz de Markov*, la cual además es una función (matricial) diferenciable en t.*

***Definición* 1.17** (Generador infinitesimal)**.** *La matriz Q de dimensión $|S| \times |S|$ cuyas entradas $[Q]_{ij}$ son tales que:*

$$[Q]_{ij} = \begin{cases} q_{ij} & si \quad i \neq j \\ -v_i = -\sum_{i \neq j} q_{ij} & caso\ contrario \end{cases} \tag{1.7}$$

es llamada generador infinitesimal *o simplemente* generador *de la cadena de Markov en tiempo continuo. Esta matriz contiene la información de las tasas de cambio de la cadena.*

Notar que, gracias a la definición de q_{ij} y v_i, la matriz Q tiene las siguientes propiedades:

GI1) $[Q]_{ij} \geq 0$, si $i \neq j$. Es decir, las entradas fuera de la diagonal son no negativas.

GI2) $[Q]_{ii} \leq 0$, para todo $i \in S$. Las entradas de la diagonal son no positivas.

GI3) $\sum_j [Q]_{ij} = 0$. Las filas de Q suman cero.

El siguiente lema será útil para demostrar las ecuaciones Backward de Kolmogorov.

***Lema* 1.1** (Ecuaciones de Chapman–Kolmogorov)**.** *Para todo $s \geq 0$ y $t \geq 0$*

$$p_{ij}(t+s) = \sum_{k=0}^{\infty} p_{ik}(t) p_{kj}(s) \tag{1.8}$$

Demostración.-. Por definición,

$$p_{ij}(t+s) = \mathbb{P}\{X(t+s)|X(0) = i\}$$

entonces, para que el proceso vaya del estado i al estado j en el tiempo $t+s$, este debe haber estado en algún estado k al tiempo t.

Se sigue que, sumando por todos los posibles k

$$p_{ij} = \sum_{k=0}^{\infty} \mathbb{P}\{X(t+s) = j, X(t) = k|X(0) = i\}$$

Ahora, condicionando respecto a todos los posibles $X(t) = k$, la expresión anterior es igual a la siguiente:

$$= \sum_{k=0}^{\infty} \mathbb{P}\{X(t+s) = j|X(t) = k, X(0) = i\} * \mathbb{P}\{X(t) = k|X(0) = i\}$$

por la propiedad de Markov se sigue que

$$p_{ij} = \sum_{k=0}^{\infty} \mathbb{P}\{X(t+s) = j|X(t) = k\} * \mathbb{P}\{X(t) = k|X(0) = i\}$$

Finalmente, por homogeneidad en el tiempo se obtiene el resultado,

$$p_{ij}(t+s) = \sum_{k=0}^{\infty} p_{kj}(s)p_{ik}(t) = \sum_{k=0}^{\infty} p_{ik}(t)p_{kj}(s)$$

□

En el siguiente teorema se presentan las *ecuaciones Backward de Kolmogorov*. La importancia de las ecuaciones Backward y Forward de Kolmogorov radica en que a través del generador infinitesimal, estas ecuaciones definen, cada una, un sistema de ecuaciones diferenciales ordinarias. Bajo las mismas condiciones iniciales, estos sistemas de ecuaciones tienen la misma solución y esta puede usarse para estimar las probabilidades de transición p_{ij}.

***Teorema* 1.1** (Ecuaciones Backward de Kolmogorov)**.** *Para todo par de estados $i, j \in S$ y tiempos $t \geq 0$,*

$$p'_{ij}(t) = \sum_{k \neq i} q_{ik}p_{kj}(t) - v_i p_{ij}(t)$$

Demostración.-. Para demostrar este teorema se va a desarrollar la expresión $p_{ij}(\Delta t+t)$, con $\Delta t > 0$ pequeño. De las ecuaciones de Chapman–Kolmogorov, se tiene

$$p_{ij}(\Delta t + t) = \sum_{k=0}^{\infty} p_{ik}(\Delta t)p_{kj}(t)$$

Separando el término con $k = i$, se sigue

$$p_{ij}(\Delta t + t) = \sum_{k \neq i} p_{ik}(\Delta t) p_{kj}(t) + p_{ii}(\Delta t) p_{ij}(t)$$

$$= \sum_{k \neq i} [q_{ik}\Delta t + o(\Delta t)] p_{kj}(t) + [1 - v_i \Delta t + o(\Delta t)] p_{ij}(t)$$

$$= \sum_{k \neq i} q_{ik}\Delta t p_{kj}(t) + \sum_{k \neq i} p_{kj}(t) o(\Delta t) + p_{ij}(t) - v_i \Delta t p_{ij}(t) + o(\Delta t) p_{ij}(t)$$

Notar que $\sum_{k \neq i} p_{kj}(t) + p_{ij}(t) = 1$, entonces la expresión anterior es equivalente a:

$$p_{ij}(\Delta t + t) - p_{ij}(t) = \sum_{k \neq i} q_{ik}\Delta t p_{kj}(t) - v_i \Delta t p_{ij}(t) + o(\Delta t)$$

dividiendo para $\Delta t > 0$, se sigue que

$$\frac{p_{ij}(\Delta t + t) - p_{ij}(t)}{\Delta t} = \sum_{k \neq i} q_{ik} p_{kj}(t) - v_i p_{ij}(t) + \frac{o(\Delta t)}{\Delta t}$$

tomando el límite cuando $\Delta t \longrightarrow 0$, se obtiene el resultado:

$$p'_{ij}(t) = \lim_{\Delta t \longrightarrow 0} \frac{p_{ij}(\Delta t + t) - p_{ij}(t)}{\Delta t} = \sum_{k \neq i} q_{ik} p_{kj}(t) - v_i p_{ij}(t).$$

□

Como afirma TAKAHARA [21], se puede ver que el lado derecho de las ecuaciones Backward de Kolmogorov es simplemente el producto escalar de la i–ésima fila de Q por la j–ésima columna de $M(t)$, es decir:

$$[M'(t)]_{ij} = [QM(t)]_{ij}$$

luego, las ecuaciones Backward pueden escribirse como:

$$M'(t) = QM(t) \tag{1.9}$$

Haciendo un razonamiento similar al hecho en la demostración de las ecuaciones Backward se pueden deducir las ecuaciones Forward de Kolmogorov, desarrollando $p_{ij}(t + \Delta t)$. Estas están dadas por:

$$p'_{ij}(t) = \sum_{k \neq i} p_{ik} q_{kj}(t) - v_j p_{ij}(t) \tag{1.10}$$

y en forma matricial tienen la forma,

$$M'(t) = M(t)Q \tag{1.11}$$

Notar que las ecuaciones (1.9) y (1.11) definen sendos sistemas de ecuaciones diferenciales ordinarias, los cuales tienen la misma condición inicial,

porque

$$p_{ii}(0) = \mathbb{P}\{X(0) = i | X(0) = i\} = 1$$
$$p_{ij}(0) = \mathbb{P}\{X(0) = j | X(0) = i\} = 0 \quad i \neq j$$

es decir, la condición inicial es:

$$M(0) = I$$

donde I es la matriz identidad de dimensión $|S| \times |S|$.

Ambos sistemas de ecuaciones diferenciales con la condición inicial anterior, tienen la misma solución y está dada por:

$$M(t) = e^{Qt} \tag{1.12}$$

donde la expresión e^{Qt} es llamada *exponencial de una matriz* y se define por,

$$e^{Qt} = \sum_{n=0}^{\infty} \frac{(tQ)^n}{n!} = I + tQ + \frac{(tQ)^2}{2!} + \frac{(tQ)^3}{3!} + \dots \tag{1.13}$$

El exponencial de matrices es una función definida sobre las matrices cuadradas que es análoga a la función exponencial para números reales. Además esta es una generalización de expansión en serie de Taylor para la función exponencial estándar. La serie en (1.13) converge absolutamente para toda $Q \in \mathbb{C}^{n\times n}$, por lo tanto, el exponencial de Q está bien definido. [18]

La serie en (1.13) es convergente y además diferenciable como lo indica WAHLÉN [22] en la siguiente proposición.

***Proposición* 1.1.** *La serie $\sum_{k=0}^{\infty} \frac{Q^k t^k}{k!}$, que define e^{Qt}, converge puntualmente en $\mathbb{R}$. Además, la función $t \mapsto e^{Qt}$ es diferenciable con derivada Qe^{Qt}.*

Demostración. Cada elemento de $\sum_{k=0}^{\infty} \frac{Q^k t^k}{k!}$ es una serie de potencias en t con coeficientes $\frac{[Q_{ij}]}{k!}$. El radio de convergencia es finito, pues

$$\left| \frac{[Q_{ij}]}{k!} \right| r^k \leq \frac{||Q_{ij}||r^k}{k!} \longrightarrow 0$$

cuando $k \longrightarrow \infty$ para todo $r \geq 0$. Por lo tanto, la serie converge puntualmente en $\mathbb{R}$.

Para encontrar la derivada de e^{Qt} diferenciamos término a término.

$$\frac{d}{dt} \sum_{k=0}^{\infty} \frac{Q^k t^k}{k!} = Q + Q^2 t + \frac{Q^3 t^2}{2!} + \dots + \frac{Q^k t^{k-1}}{(k-1)!} \dots$$
$$= Q \left(I + Qt + \frac{Q^2 t^2}{2!} + \frac{Q^3 t^3}{3!} + \dots + \frac{Q^{k-1} t^{k-1}}{(k-1)!} \dots \right)$$
$$= \sum_{k=1}^{\infty} \frac{t^{k-1} Q^k}{(k-1)!} = Q \sum_{j=0}^{\infty} \frac{t^j Q^j}{j!}, \quad \text{con } j = k-1$$

□

En la proposición (1.1) se ha demostrado que la serie que define e^Q siempre converge, por esta razón una forma de aproximar e^Q para una matriz Q fija, es calcular una serie de Taylor truncada con k términos. [18]

Si además Q es diagonalizable, entonces se puede escribir $Q = S\Lambda S^{-1}$, donde Λ es una matriz diagonal compuesta por los valores propios de Q y S es una matriz cuyas columnas corresponden a los vectores propios de Q. Luego, se puede calcular el exponencial de Q como:

$$e^Q = \sum_{k=0}^{\infty} \frac{Q^k}{k!} = I + S\Lambda S^{-1}t + \frac{1}{2}S\Lambda^2 S^{-1}t^2 + \dots$$

$$= \sum_{k=0}^{\infty} \frac{1}{k!}(S\Lambda S^{-1})^k t^k = S\left(\sum_{k=0}^{\infty} \frac{\Lambda^k t^k}{k!}\right) S^{-1}$$

$$= Se^{\Lambda t}S^{-1}$$

$$= S\begin{pmatrix} e^{\lambda_1 t} & 0 & \dots \\ 0 & e^{\lambda_2 t} & \dots \\ & \ddots & \\ \dots & 0 & e^{\lambda_n t} \end{pmatrix} S^{-1}$$

1.4.3 Distribuciones estacionarias

Una par de definiciones importantes para el estudio de modelos evolutivos son las de distribución estacionaria y reversibilidad en el tiempo.

***Definición* 1.18.** *Sea* $\mathbf{X} = \{X(t), t \geq 0\}$ *una cadena de Markov en tiempo continuo con espacio de estados finito* S*, generador* Q *y matriz de transición* $M(t)$*. Un vector de dimensión* $|S|$*,* $\pi = (\pi_i)_{i\in S}$ *con* $\pi_i \geq 0$ *para todo* $i \in S$ *y* $\sum_{i\in S} \pi_i = 1$*, se dice* distribución estacionaria *de* $\mathbf{X}$ *si,*

$$\pi = \pi M(t) \quad \forall t \geq 0 \tag{1.14}$$

La relación entre π y Q está dada por las llamadas ecuaciones de balance, las cuales indican que

$$0 = \pi Q \tag{1.15}$$

En efecto, π es un vector de distribución estacionaria si y solo si,

$$\pi = \pi M(t) \quad \forall t \geq 0$$

$$\iff \pi = \pi e^{Qt} \quad \forall t \geq 0$$

$$\iff \pi = \pi\left(I + tQ + \frac{(tQ)^2}{2!} + \frac{(tQ)^3}{3!} + \dots\right) \quad \forall t \geq 0$$

$$\iff \pi - \pi = \pi\left(tQ + \frac{(tQ)^2}{2!} + \frac{(tQ)^3}{3!} + \dots\right) \quad \forall t \geq 0$$

$$\iff 0 = \pi \sum_{n=1}^{\infty} \frac{(tQ)^n}{n!} \quad \forall t \geq 0$$

$$\iff 0 = \pi \sum_{n=1}^{\infty} \frac{t^n}{n!} \pi Q \quad \forall t \geq 0$$

$$\iff 0 = \pi Q^n \quad \forall n \geq 1$$

$$\iff 0 = \pi Q$$

La j–ésima ecuación de balance viene dada por

$$0 = -v_j \pi_j + \sum_{i \neq j} q_{ij} \pi_i$$

$$v_j \pi_j = \sum_{i \neq j} q_{ij} \pi_i \tag{1.16}$$

la cual se puede interpretar de la siguiente manera:

> "La tasa a largo plazo de dejar el estado j debe ser igual a la tasa a largo plazo de entrar al estado j." [21].

Notar que π_j es la proporción de tiempo que el proceso pasa en j a largo plazo y v_j es la tasa a la que el proceso está saliendo del estado j.

Por otro lado, q_{ij} es la tasa a la cual el proceso pasa del estado i a j, entonces el producto $\pi_i q_{ij}$, es la tasa a largo plazo a la cual el proceso está saliendo del estado i para entrar en el estado j. Para encontrar la tasa total a largo plazo a la que el proceso entra a j sumamos sobre todos los $i \neq j$.

Finalmente, la distribución π hace que el proceso sea estacionario, es decir, si la distribución de $X(0)$ es π, entonces la distribución de $X(t)$ también es π para todo $t \geq 0$. En efecto, supongamos que y $X(t) = j$, entonces condicionando para $X(0) = i$ se tiene,

$$\mathbb{P}\{X(t) = j\} = \sum_{i \in S} \mathbb{P}\{X(t) = j | X(0) = i\} \mathbb{P}\{X(0) = i\}$$

pero $\mathbb{P}\{X(0) = i\} = \pi_i$ porque la distribución de $X(0)$ es π, luego

$$\mathbb{P}\{X(t) = j\} = \sum_{i \in S} \mathbb{P}\{X(t) = j | X(0) = i\} \pi_i = \sum_{i \in S} \pi_i p_{ij}$$

$$= [\pi M(t)]_j$$

$$= \pi_j$$

1.4.4 Reversibilidad en el tiempo

Para terminar la sección de cadenas de Markov en tiempo continuo presentamos la propiedad de *reversibilidad en el tiempo*.

En el estudio de Árboles Filogenéticos, la cualidad de reversibilidad en el tiempo de una cadena de Markov ayuda a definir Modelos Evolutivos más simples y con propiedades útiles.

***Definición* 1.19.** *Una cadena de Markov en tiempo continuo* $\mathbf{X} = \{X(t), t \geq 0\}$ *es* reversible en el tiempo *si* $\mathbf{X}$ *tiene una distribución estacionaria* π *y cuando el proceso es extendido a todos los* $t \in \mathbb{R}$ *para obtener el proceso estacionario* $\{X(t), t \in \mathbb{R}\}$*, entonces el proceso revertido* $\mathbf{Y} = \{Y(t) = X(-t), t \in \mathbb{R}\}$*, tiene la misma estructura probabilística que* $\mathbf{X}$*.*

1.5 Hipótesis del Reloj Molecular

En la sección anterior se definió el concepto de cadena de Markov continua y homogénea en el tiempo. Además, los lemas y teoremas presentados tenían como supuesto que la cadena sea homogénea.

Como la evolución de un grupo de taxones puede ser modelada por medio de una cadena de Markov en tiempo continuo, es necesario determinar si este proceso es homogéneo en el tiempo. Para lograr esto es necesario introducir la *hipótesis del reloj molecular*.

En pocas palabras, un *reloj molecular* es una hipótesis que predice una tasa constante de evolución molecular entre especies. [2]

Esta hipótesis fue propuesta por Emile Zuckerkandl y Linus Pauling en 1965 luego de haber observado que la tasa de evolución para proteínas como la hemoglobina, era relativamente constante entre diferentes órdenes de mamíferos. [24]

La suposición de una tasa constante evolución de proteínas y ADN, en el tiempo y entre linajes, puede usarse para reconstruir relaciones filogenéticas entre especies utilizando su información genética y métodos de inferencia estadística como Máxima Verosimilitud o inferencia Bayesiana.

Además, esta tasa constante de evolución permite tener probabilidades homogéneas de mutación para los nucleótidos y de esta forma, suponer que la cadena de Markov que gobierna el proceso evolutivo es homogénea en el tiempo y que posee las propiedades presentadas en la sección anterior.

1.5.1 Controversia

A pesar de la utilidad del reloj molecular, en especial para hacer reconstrucción filogenética, esta hipótesis causó mucha controversia desde que se propuso a principios de los años 60's.

> En ese entonces lo generalmente aceptado entre evolucionistas era lo que planteaba el Neo–Darwinismo: que la tasa de evolución es determinada por cambios ambientales y selección natural.
>
> A finales de los años 60's, la *Teoría Neutral de Evolución Molecular* fue propuesta por Motoo Kimura. Esta teoría afirma que la evolución molecular no está dominada por la selección natural, sino por una fijación aleatoria de mutaciones neutrales; en otras palabras, está dominada por mutaciones cuyos efectos en la adaptación son muy pequeños para que la selección natural juegue un papel importante en su destino. De esta forma, la tasa de evolución molecular es igual a la tasa de mutación neutral y es

independiente de factores como cambios ambientales o tamaños de las poblaciones.

Sin embargo es necesario hacer ciertas aclaraciones:

1. El reloj molecular se imagina que es estocástico y los cambios moleculares se acumulan aleatoriamente de acuerdo a un proceso de Poisson, de tal forma que se esperan fluctuaciones aleatorias aunque la subyacente tasa es constante en el tiempo.
2. Diferentes genes, diferentes proteínas o diferentes regiones del mismo gen pueden tener diferentes tasas evolutivas y sus relojes pueden avanzar a diferentes tasas. La teoría neutral explica esto de la siguiente manera, diferentes genes se encuentra bajo diferentes restricciones evolutivas y aquellos bajo restricciones fuertes tendrán una proporción más pequeña de mutaciones neutrales y por consiguiente una tasa de evolución menor.
3. No se espera que el reloj molecular sea universal y usualmente es aplicado a un grupo de especies. Por ejemplo, podemos decir que el reloj se cumple para un gen dentro de los mamíferos.

[24]

Para finalizar, dado que el objetivo del presente trabajo de titulación es simular modelos evolutivos a partir de un árboles dados, se supone que los mismos satisfacen la hipótesis del reloj molecular.

Bajo este supuesto, los largos de las ramas deben verse como mediadas de cuánta mutación ha ocurrido y para obtener los largos de ramas, simplemente escalamos el intervalo de tiempo por una tasa constante de mutación. [3]

2 Modelos Probabilísticos de Mutación del ADN

Como ya se mencionó en la sección (1.4) del Capítulo 1, existen varias formas de hacer inferencia de árboles filogenéticos además se hará uso de los métodos que utilicen información proveniente del ADN de los organismos. Más específicamente se hará uso de los métodos basados en *Modelos Evolutivos*.

En el presente capítulo se explicará cómo el proceso evolutivo en una rama de un árbol filogenético puede modelarse por medio de una cadena de Markov en tiempo continuo y homogénea.

2.1 Modelo de Markov para una rama

Para modelar la evolución en un árbol filogenético, debemos enfocarnos en el proceso de sustitución de cada rama del árbol. Por esta razón, primero se modelará la evolución para rama.

Supongamos que tenemos dos nodos: uno ancestral y su descendiente. Sean S_0 la secuencia de ADN del ancestro y S_1 la del descendiente que ha evolucionado de S_0. Entonces los sitios en las secuencias S_0 y S_1 pueden tomar los siguientes estados:

$$\mathcal{E} = \{A, G, C, T\}.$$

Como afirman Strimmer y von Haeseler [20], para modelar el proceso de sustitución en la rama que conecta S_0 con S_1 es necesario hacer las siguientes suposiciones:

1. En cualquier sitio de una secuencia de ADN, el cambio de la base i a la base j ocurre de manera aleatoria e independiente de la base que haya ocupado ese sitio antes de i.

2. Las tasas de sustitución no cambian en el tiempo.

3. Las frecuencias relativas de A, G, C y T están en equilibrio.

De esta forma, se puede tratar a cada sitio en la secuencia de ADN como una variable aleatoria que toma uno de estos cuatro posibles estados: A, G, C o T. Entonces, la evolución en la rama que conecta el ancestro con el descendiente, en otras palabras, la probabilidad de que un estado en la secuencia ancestral sea reemplazado por otro en la secuencia del descendiente, puede modelarse por medio de una cadena de Markov homogénea en tiempo continuo.

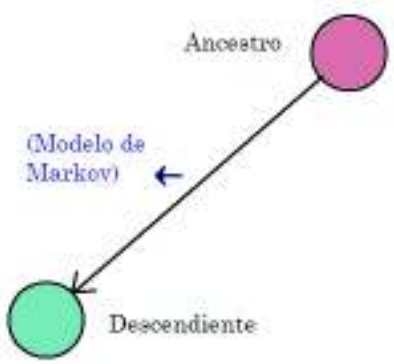


FIGURA 2.1: Modelo de Markov para una rama.

***Definición* 2.1.** *Se define a la matriz de* tasas instantáneas de cambio Q*, como el generador infinitesimal del proceso estocástico que modela la evolución de la secuencia S_0 en S_1, de acuerdo a la definición (1.17) de la sección (1.4).*

Para secuencias de ADN, la matriz Q es de dimensión 4×4 y contiene las tasas de cambio entre nucleótidos. Esta tiene la siguiente forma:

$$Q = \begin{pmatrix} q_{AA} & q_{AG} & q_{AC} & q_{AT} \\ q_{GA} & q_{GG} & q_{GC} & q_{GT} \\ q_{CA} & q_{CG} & q_{CC} & q_{CT} \\ q_{TA} & q_{TG} & q_{TC} & q_{TT} \end{pmatrix}$$

Sea $M(t)$ la matriz de transición que modela el proceso en la rama que conecta la secuencia ancestral S_0 con su descendiente S_1 y que tiene a Q como matriz de tasas. Supongamos que $p_t = (p_A(t), p_G(t), p_C(t), p_T(t))$ es el vector de distribución de los estados al tiempo t, con $t = 0$ para la secuencia del ancestro, entonces la relación entre $M(t)$ y Q se obtiene al resolver el siguiente sistema de ecuaciones diferenciales ordinarias.

$$\begin{cases} \frac{d}{dt} p_t = p_t Q \\ p(0) = p_0 \end{cases} \tag{2.1}$$

La solución del sistema de ecuaciones (2.1) es:

$$p_t = p_0 e^{Qt} = p_0 M(t) \tag{2.2}$$

con $M(t) = e^{Qt}$ y $p_0 = p(0)$ es el vector de distribución de las bases nitrogenadas para la secuencia del ancestro S_0.

En efecto, por la proposición (1.1), la solución del sistema de ecuaciones diferenciales ordinarias de primer orden en (2.1) tiene la forma,

$$p_t = \mathbf{c} e^{Qt}, \quad \text{con } \mathbf{c} \in \mathbb{R}^4. \tag{2.3}$$

Utilizando la condición inicial y debido a que $e^{Q0} = e^{\mathbf{0}} = I$, se tiene:

$$p(0) = p_0 = \mathbf{c}e^{Q0}\mathbf{c}I$$

$$\Longrightarrow \mathbf{c} = p_0.$$

Por lo tanto,

$$p_t = p_0 e^{Qt}$$

2.2 El modelo general de sustitución de nucleótidos

Luego de haber definido el modelo de Markov para una rama, se puede generalizar esta definición para un árbol filogenético $\mathcal{T}^\rho$, asignando matrices de Markov a cada una de sus ramas.

Como indican Allman y Rhodes [3], la definición siguiente corresponde al *modelo general de sustitución de nucleótidos*.

***Definición* 2.2.** *Dado un árbol filogenético enraizado $\mathcal{T}^\rho$ con raíz ρ, el* Modelo General de Markov *de evolución del ADN sobre $\mathcal{T}^\rho$ consiste de los siguientes parámetros:*

1. *Un* vector de distribución de estados en la raíz $\mathbf{p}_\rho = (p_A, p_G, p_C, p_T)$. *Tal que:*

$$p_A, p_G, p_C, p_T \geq 0 \quad y \quad p_A + p_G + p_C + p_T = 1 \tag{2.4}$$

   Asumiendo que cada sitio en la cadena de ADN se comporta de manera independiente y con idéntica distribución, las entradas del vector $\mathbf{p}_\rho$ se interpretan como las frecuencias con las que se observan las distintas bases nitrogenadas en el nodo raíz ρ.

2. *Para cada arco $\mathbf{e} = (u, v)$ de $\mathcal{T}^\rho$, se tiene una* matriz de Markov $M_e(t)$ *con matriz de tasas Q_e*

Notar que la matriz M está totalmente definida por la matriz de tasas Q.

En la práctica, las matrices de Markov de un modelo evolutivo para un árbol $\mathcal{T}^\rho$ están relacionadas entre sí y más comúnmente se suele asumir que todas las ramas (aristas) del árbol comparten el mismo modelo evolutivo. Como indican Allman y Rhodes [3], el punto (2) de la definición anterior se puede modificar de la siguiente manera:

2a) Una matriz Q cuadrada de dimensión 4, cuyas entradas fuera de la diagonal son no negativas y sus filas suman 0.

$$Q = \begin{pmatrix} q_{AA} & q_{AG} & q_{AC} & q_{AT} \\ q_{GA} & q_{GG} & q_{GC} & q_{GT} \\ q_{CA} & q_{CG} & q_{CC} & q_{CT} \\ q_{TA} & q_{TG} & q_{TC} & q_{TT} \end{pmatrix} \tag{2.5}$$

las entradas q_{ij} de Q se interpretan como la tasa instantánea de cambio (en sustituciones en un sitio por unidad de tiempo) en la que la base i es reemplazada por la base j, para $i, j \in \mathcal{E}$. Las entradas q_{ii} pueden ser vistas como la tasa con la que se pierden las bases i, por esta razón las entradas de la diagonal de Q son negativas.

2b Para cada arista $\mathbf{e} \in \mathcal{T}^{\rho}$ de largo t_e, se define la matriz de Markov

$$M_e = M(t_e) = e^{Qt_e} \tag{2.6}$$

2.3 Suposiciones de los modelos

1. Para que se satisfaga la propiedad de Markov, se supone que el tiempo transcurre desde la raíz hacia las hojas y que las probabilidades de cambio de estado en una arista dada, dependen únicamente del estado del ancestro inmediatamente anterior a ese nodo.

2. Se supone que el proceso en una rama es independiente de los procesos en las ramas restantes. Cualquier correlación entre estados de las hojas proviene únicamente del estado de su más reciente ancestro común.

3. Cada sitio en las secuencias de ADN de las hojas se comporta de manera independiente e idénticamente distribuida.

2.3.1 Distribución de los estados en las hojas

Si se desea conocer el vector que contiene las probabilidades de observar las cuatro bases en cualquier sitio de la secuencia del descendiente, el procedimiento es bastante directo.

Supongamos que se conoce el vector de distribución de la raíz y las matrices de Markov que describen el proceso evolutivo en cada rama del árbol. Entonces, como indican Allman y Rhodes [3], se tiene tres casos,

1. **Árbol con una sola rama**

   Si $\mathbf{p}_{\rho}$ y M_e son los de la rama que conecta a la raíz ρ con el descendiente S_1, entonces el vector de distribución de bases de S_1 es

   $$\mathbf{p}_1 = \mathbf{p}_{\rho} M_e \tag{2.7}$$

2. **Árbol con dos ramas sin nodos interiores**

   Conociendo el vector de distribución de la raíz ρ y las respectivas matrices de Markov relacionadas a cada rama d y e, el cálculo de los vectores de distribución de bases en las hojas es también directo.

   Basta calcular el producto $\mathbf{p}_1 = \mathbf{p}_{\rho} M_d$ para el descendiente de la rama d y $\mathbf{p}_2 = \mathbf{p}_{\rho} M_e$ para el descendiente de la rama e.

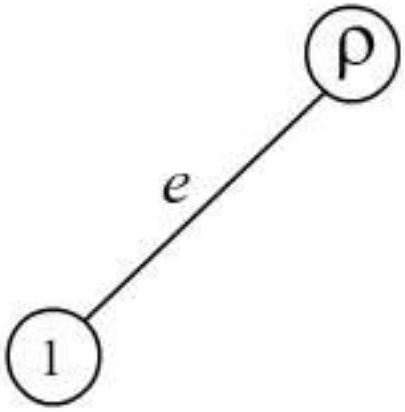


FIGURA 2.2: Árbol con una rama

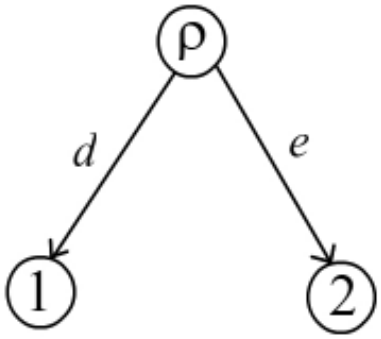


FIGURA 2.3: Árbol con dos ramas sin nodos internos

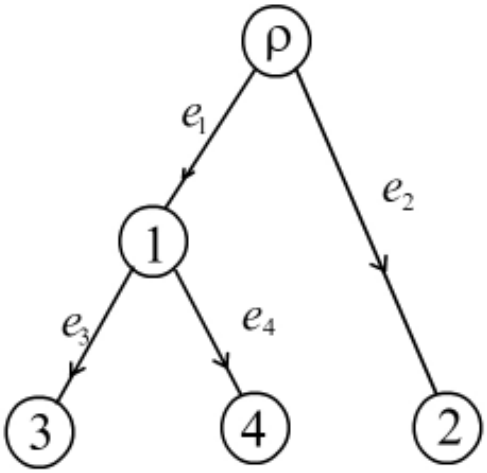


FIGURA 2.4: Árbol con nodos internos

3. **Árbol con más de dos ramas y con nodos internos**

   En este caso, se combinan los casos anteriores, es decir, se realizan las multiplicaciones partiendo desde la raíz ρ hacia "abajo" en el árbol como se muestra en la figura (2.4) .

2.4 Ejemplos de modelos evolutivos

En las secciones anteriores se dedujo el Modelo General de Markov (MGM). Añadiendo ciertas restricciones a los parámetros de este modelo, se pueden definir casos especiales del MGM que son muy conocidos, utilizados en la práctica y que han sido nombrados en honor a aquellas personas que los han desarrollado.

A continuación se presentan algunos ejemplos.

2.4.1 El Modelo de Jukes-Cantor

Este modelo fue introducido en 1969 por Thomas Jukes y Charles P. Cantor. Es el modelo más restrictivo y será abreviado por JC69.

***Definición* 2.3.** *El modelo JC69 consta de los siguientes parámetros:*

1. *Un vector de distribución de la raíz*

$$p_\rho = (1/4, 1/4, 1/4, 1/4)$$

2. *El modelo de Markov en tiempo continuo tiene a la matriz de tasas Q de la forma,*

$$Q = \begin{pmatrix} -\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha \end{pmatrix} \tag{2.8}$$

***Observación* 2.1.** El modelo JC69 supone que la distribución de las bases en la raíz es equiprobable y además la tasa con la que ocurren los cambios es la misma $\alpha/3$.

***Teorema* 2.1.** *La matriz de Markov asociada a la matriz Q en la ecuación (2.8) del modelo JC69 es*

$$M(t) = e^{Qt} = \begin{pmatrix} 1-a & a/3 & a/3 & a/3 \\ a/3 & 1-a & a/3 & a/3 \\ a/3 & a/3 & 1-a & a/3 \\ a/3 & a/3 & a/3 & 1-a \end{pmatrix} \tag{2.9}$$

con $a = a(t) = \frac{3}{4}\left(1 - \exp\left\{-\frac{4\alpha t}{3}\right\}\right)$

Demostración. Para calcular e^{Qt} se utilizará el método de la diagonalización de Q. Entonces $Q = V\Lambda V^{-1}$.

$$M(t) = e^{Qt} = Ve^{\Lambda t}V^{-1}$$

Primero, se calculará los valores propios de Q.

$$p(\lambda) = |Q - \lambda I| = \begin{vmatrix} -\alpha-\lambda & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha-\lambda & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha-\lambda & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha-\lambda \end{vmatrix}$$

Sea $\alpha' = \alpha/3$ y sea $x = -3\alpha' - \lambda$.

$$|Q - \lambda I| = \begin{vmatrix} x & \alpha' & \alpha' & \alpha' \\ \alpha' & x & \alpha' & \alpha' \\ \alpha' & \alpha' & x & \alpha' \\ \alpha' & \alpha' & \alpha' & x \end{vmatrix}$$

$$= x\begin{vmatrix} x & \alpha' & \alpha' \\ \alpha' & x & \alpha' \\ \alpha' & \alpha' & x \end{vmatrix} - \alpha'\begin{vmatrix} \alpha' & \alpha' & \alpha' \\ \alpha' & x & \alpha' \\ \alpha' & \alpha' & x \end{vmatrix} + \alpha'\begin{vmatrix} \alpha' & x & \alpha' \\ \alpha' & x & \alpha' \\ \alpha' & \alpha' & x \end{vmatrix} - \alpha'\begin{vmatrix} \alpha' & x & \alpha' \\ \alpha' & \alpha' & x \\ \alpha' & \alpha' & \alpha' \end{vmatrix}$$

$$= p(\lambda) = |Q - \lambda I| = a + b + c + d$$

$$a = x(x^3 + 2\alpha'^3 - 3\alpha'^2 x)$$

$$b = -\alpha'(\alpha' x^2 + \alpha'^3 - 2\alpha'^2 x)$$

$$c = \alpha'(2\alpha'^2 x - \alpha'^3 - \alpha' x^2)$$

$$d = -\alpha'(\alpha'^3 + \alpha' x^2 - 2\alpha'^2 x)$$

$$p(\lambda) = x^4 + 2\alpha'^3 x - 3\alpha'^2 x^2 - \alpha'^2 x^2 - \alpha'^4 + 2\alpha'^3 x$$

$$+2\alpha'^3 x - \alpha'^4 - \alpha'^2 x^2 - \alpha'^4 - \alpha'^2 x^2 + 2\alpha'^3 x$$

$$p(\lambda) = x^4 + 8\alpha'^3 x - 6\alpha'^2 x^2 - 3\alpha'^4$$

Se reemplaza $x = 3\alpha' - \lambda$

$$p(\lambda) = (3\alpha' - \lambda)^4 + 8\alpha'^3(3\alpha' - \lambda) - 6\alpha'^2(3\alpha' - \lambda)^2 - 3\alpha'^4$$

$$p(\lambda) = \lambda^4 + 12\alpha'\lambda^3 + 12\alpha'^2\lambda^2 + 64\alpha'^3\lambda$$

$$p(\lambda) = \lambda(\lambda^3 + 12\alpha'\lambda^2 + 12\alpha'^2\lambda + 64\alpha'^3$$

$$p(\lambda) = \lambda(\lambda + 4\alpha')^3$$

$$\lambda_1 = 0$$

$$\lambda_{2,3,4} = -4\alpha' = -\frac{4\alpha}{3}$$

Ahora se calcula los vectores propios asociados a cada valor propio. Valor propio asociado a $\lambda_1 = 0$:

$$\begin{pmatrix} -\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\implies v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Valor propio asociado a $\lambda_2 = -\frac{4\alpha}{3}$:

$$\begin{pmatrix} -\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -\frac{4\alpha}{3}x_1 \\ -\frac{4\alpha}{3}x_2 \\ -\frac{4\alpha}{3}x_3 \\ -\frac{4\alpha}{3}x_4 \end{pmatrix}$$

$$\begin{pmatrix} \alpha & \alpha & \alpha & \alpha \\ \alpha & \alpha & \alpha & \alpha \\ \alpha & \alpha & \alpha & \alpha \\ \alpha & \alpha & \alpha & \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\implies x_1 + x_2 + x_3 + x_4 = 0 \implies v_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

Valor propio asociado a $\lambda_3 = -\frac{4\alpha}{3}$

$$v_3 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

Valor propio asociado a $\lambda_4 = -\frac{4\alpha}{3}$

$$v_4 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

Luego,

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \quad \Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{4\alpha}{3} & 0 & 0 \\ 0 & 0 & -\frac{4\alpha}{3} & 0 \\ 0 & 0 & 0 & -\frac{4\alpha}{3} \end{pmatrix}$$

Sea $x = -\frac{4\alpha t}{3}$, se tiene

$$M(t) = e^{Qt} = Ve^{\Lambda t}V^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^x & 0 & 0 \\ 0 & 0 & e^x & 0 \\ 0 & 0 & 0 & e^x \end{pmatrix} \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & 1/4 & -1/4 & -1/4 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{pmatrix}$$

$$= \begin{pmatrix} 1/4 + 3e^x/4 & 1/4 - e^x/4 & 1/4 - e^x/4 & 1/4 - e^x/4 \\ 1/4 - e^x/4 & 1/4 + 3e^x/4 & 1/4 - e^x/4 & 1/4 - e^x/4 \\ 1/4 - e^x/4 & 1/4 - e^x/4 & 1/4 + 3e^x/4 & 1/4 - e^x/4 \\ 1/4 - e^x/4 & 1/4 - e^x/4 & 1/4 - e^x/4 & 1/4 + 3e^x/4 \end{pmatrix}$$

Sea $e^x = 1 - \frac{4}{3}a(t)$, entonces

$$1/4 + 3e^x/4 = \frac{1}{4} + \frac{3}{4} - \frac{3}{4} \times \frac{4}{3}a(t) = 1 - a(t)$$

$$1/4 - e^x/4 = \frac{1}{4} - \frac{1}{4} + \frac{4}{3} \times \frac{1}{4}a(t) = a(t)/3$$

Por lo tanto,

$$M(t) = \begin{pmatrix} 1 - a(t) & a(t)/3 & a(t)/3 & a(t)/3 \\ a(t)/3 & 1 - a(t) & a(t)/3 & a(t)/3 \\ a(t)/3 & a(t)/3 & 1 - a(t) & a(t)/3 \\ a(t)/3 & a(t)/3 & a(t)/3 & 1 - a(t) \end{pmatrix}$$

con $a(t) = \frac{3}{4}(1 - e^{4\alpha t/3})$. □

***Observación* 2.2.** $a(t)$ es la probabilidad de que cualquier base i haya cambiado a cualquier otra j, $i, j \in \mathcal{E}'$.

Notar que el modelo JC69 asigna distribución uniforme a las bases nitrogenadas para todos los nodos del árbol, en efecto:

$$p_\rho(t) = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) \begin{pmatrix} 1-a(t) & a(t)/3 & a(t)/3 & a(t)/3 \\ a(t)/3 & 1-a(t) & a(t)/3 & a(t)/3 \\ a(t)/3 & a(t)/3 & 1-a(t) & a(t)/3 \\ a(t)/3 & a(t)/3 & a(t)/3 & 1-a(t) \end{pmatrix} = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) \tag{2.10}$$

para cualquier nodo ρ.

2.4.2 El Modelo Kimura de 2 parámetros

Si se añade un grado más de libertad β al MGM, se obtiene el llamado Modelo de Kimura de dos parámetros o Kimura–2 o simplemente K80, este modelo fue introducido en 1980 por Motoo Kimura [9]. En este caso, se asume que las transiciones y las transversiones ocurren con probabilidades distintas.

La matriz de tasas de cambio tiene la siguiente forma.
Sea α la tasa de sustituciones de tipo transición y
sea β la tasa de sustituciones de tipo transversión

$$Q = \begin{pmatrix} y & \alpha & \beta & \beta \\ \alpha & y & \beta & \beta \\ \beta & \beta & y & \alpha \\ \beta & \beta & \alpha & y \end{pmatrix} \tag{2.11}$$

con $y = -\alpha - 2\beta$. La tasa total de cambio es por tanto, $\alpha + 2\beta$
La matriz de Markov tiene la siguiente forma

$$M(t) = e^{Qt} = \begin{pmatrix} x & a(t) & b(t) & b(t) \\ a(t) & x & b(t) & b(t) \\ b(t) & b(t) & x & a(t) \\ b(t) & b(t) & a(t) & x \end{pmatrix} \tag{2.12}$$

con $x = 1 - a(t) - 2b(t)$.

***Teorema* 2.2.** *Las probabilidades de transición y transversión para el modelo Kimura–2 son:*

$$\mathcal{P}(\textit{transición}|t) = \mathbf{a}(t) = \frac{1}{4}\left(1 - 2\exp\{-2(\alpha+\beta)t\} + \exp\{-4\beta t\}\right)$$

$$\mathcal{P}(\textit{transversión}|t) = \mathbf{b}(t) = \frac{1}{4}\left(1 - \exp\{-4\beta t\}\right)$$

Demostración. Por definición $M(t) = e^{Qt}$, luego por la proposición (1.1) la derivada de $M(t)$ es,

$$M'(t) = Qe^{Qt} = M(t)Q$$

$$= \begin{pmatrix} x & a(t) & b(t) & b(t) \\ a(t) & x & b(t) & b(t) \\ b(t) & b(t) & x & a(t) \\ b(t) & b(t) & a(t) & x \end{pmatrix} \begin{pmatrix} y & \alpha & \beta & \beta \\ \alpha & y & \beta & \beta \\ \beta & \beta & y & \alpha \\ \beta & \beta & \alpha & y \end{pmatrix}$$

El producto de $M(t)Q$ es la siguiente matriz simétrica:

$$M'(t) = \begin{pmatrix} xy + \alpha\mathbf{a} + 2\beta\mathbf{b} & y\mathbf{a} + \alpha x + 2\beta\mathbf{b} & y\mathbf{b} + \alpha\mathbf{b} + \beta\mathbf{a} + \beta x & y\mathbf{b} + \alpha\mathbf{b} + \beta\mathbf{a} + \beta x \\ - & & y\mathbf{b} + \alpha\mathbf{b} + \beta\mathbf{a} + \beta x & y\mathbf{b} + \alpha\mathbf{b} + \beta\mathbf{a} + \beta x \\ & & - & y\mathbf{a} + \alpha x + 2\beta\mathbf{b} \\ & & & - \end{pmatrix}$$

De la expresión anterior se puede definir un sistema de ecuaciones diferenciales ordinarias para calcular las expresiones de las probabilidades de transición y transversión.

$$\begin{cases} \frac{d}{dt}\mathbf{a}(t) = \frac{d}{dt}P_{AG}(t) = \frac{d}{dt}P_{CT}(t) & (1) \\ \frac{d}{dt}\mathbf{b}(t) = \frac{d}{dt}P_{AC}(t) = \frac{d}{dt}P_{GC}(t) = \frac{d}{dt}P_{AT}(t) = \frac{d}{dt}P_{GT}(t) & (2) \end{cases}$$

$$\begin{cases} \frac{d}{dt}\mathbf{a}(t) = y\mathbf{a}(t) + \alpha x + 2\beta\mathbf{b}(t) & (1) \\ \frac{d}{dt}\mathbf{b}(t) = y\mathbf{b}(t) + \alpha\mathbf{b}(t) + \beta\mathbf{a}(t) + \beta x & (2) \end{cases}$$

Resolviendo (2).

$$\frac{d}{dt}b(t) = yb(t) + \alpha b(t) + \beta a(t) + \beta x$$

$$\frac{d}{dt}b(t) = (-\alpha - 2\beta)b(t) + \alpha b(t) + \beta a(t) + \beta(1 - a(t) - 2b(t))$$

$$\frac{d}{dt}b(t) = \beta - 4\beta b(t)$$

$$\int \frac{db(t)}{\beta - 4\beta b(t)} = \int dt$$

$$\frac{\log(\beta - 4\beta b(t))}{-4\beta} = t + c$$

$$\log(\beta - 4\beta b(t)) = -4\beta(t + c)$$

$$\beta - 4\beta b(t) = ce^{-4\beta t}$$

$$b(t) = \frac{\beta - ce^{-4\beta t}}{4\beta} = \frac{1}{4} - \frac{c}{4\beta}e^{-4\beta t}$$

La condición inicial es $b(0) = 0$ porque no existen transversiones al tiempo $t = 0$. Entonces,

$$b(0) = \frac{1}{4} - \frac{c}{4\beta}e^{0} = 0 \quad \Longrightarrow \quad c = \frac{1}{4} \times 4\beta = \beta$$

Por tanto la probabilidad de transversión es

$$b(t) = \frac{1}{4}\left(1 - e^{-4\beta t}\right) \tag{2.13}$$

Ahora se procede a resolver la ecuación (1).

$$\frac{d}{dt}a(t) = \alpha - 2(\alpha+\beta)a(t) + 2(\beta-\alpha)b(t) = \alpha - 2(\alpha+\beta)a(t) + 2(\beta-\alpha)\times\frac{1}{4}(1-e^{-4\beta t})$$

$$\frac{d}{dt}a(t) + 2(\alpha+\beta)a(t) = \frac{1}{2}(\alpha+\beta) - \frac{1}{2}(\beta-\alpha)e^{-4\beta t}$$

Se utilizará el método del factor integrante para resolver la ecuación diferencial (1) con:

$$p(t) = 2(\alpha+\beta) \quad \text{y} \quad q(t) = \frac{1}{2}(\alpha+\beta) - \frac{1}{2}(\beta-\alpha)e^{-4\beta t}$$

$$a(t)\exp\left(\int 2(\alpha+\beta)dt\right) = \int\left(\frac{1}{2}(\alpha+\beta) - \frac{1}{2}(\beta-\alpha)e^{-4\beta t}\right)\exp\left(\int 2(\alpha+\beta)dt\right)dt$$

$$a(t)e^{2(\alpha+\beta)t} = \frac{1}{2}(\alpha+\beta)\int e^{2(\alpha+\beta)t}dt + \frac{1}{2}(\alpha-\beta)\int e^{2(\alpha-\beta)t}dt$$

$$a(t)e^{2(\alpha+\beta)t} = \frac{(\alpha+\beta)}{2}\times\frac{e^{2(\alpha+\beta)t}}{2(\alpha+\beta)} + \frac{\alpha-\beta}{2}\times\frac{e^{2(\alpha-\beta)t}}{2(\alpha-\beta)} + c$$

$$a(t)e^{2(\alpha+\beta)t} = \frac{1}{4}e^{2(\alpha+\beta)t} + \frac{1}{4}e^{2(\alpha-\beta)t} + c$$

Nuevamente, la condición inicial es $a(t) = 0$ porque al tiempo $t = 0$ no existen transiciones. Entonces,

$$0 = \frac{1}{4}e^{0} + \frac{1}{4}e^{0} + c \quad \Longrightarrow \quad c = -\frac{1}{2}$$

Por lo tanto, la probabilidad de transición es

$$a(t) = \frac{1}{e^{2(\alpha+\beta)t}}\left(\frac{1}{4}e^{2(\alpha+\beta)t} + \frac{1}{4}e^{2(\alpha-\beta)t} - \frac{1}{2}\right)$$

$$a(t) = \frac{1}{4} + \frac{1}{4}e^{-4\beta t} - \frac{1}{2}e^{-2(\alpha+\beta)t}$$

$$a(t) = \frac{1}{4}\left(1 - 2e^{-2(\alpha+\beta)t} + e^{-4\beta t}\right) \tag{2.14}$$

□

El modelo Kimura–2 asume como vector de distribución de bases en la raíz al vector,

$$\mathbf{p}_\rho = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$$

En efecto,

$$\lim_{t\longrightarrow\infty} a(t) = \lim_{t\longrightarrow\infty} \frac{1}{4}\left(1 - 2e^{-2(\alpha+\beta)t} + e^{-4\beta t}\right) = \frac{1}{4}$$

$$\lim_{t\longrightarrow\infty} b(t) = \lim_{t\longrightarrow\infty} \frac{1}{4}\left(1 - e^{-4\beta t}\right) = \frac{1}{4}$$

entonces,

$$\lim_{t\longrightarrow\infty} M(t) = \lim_{t\longrightarrow\infty} \begin{pmatrix} x & a(t) & b(t) & b(t) \\ a(t) & x & b(t) & b(t) \\ b(t) & b(t) & x & a(t) \\ b(t) & b(t) & a(t) & x \end{pmatrix} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

porque $x = 1 - a(t) - 2b(t)$.

2.4.3 Representación alternativa del modelo K2P

Se presenta una forma alternativa de representar al modelo K2P, de tal manera que su generador Q esté expresado en función de un único parámetro κ (kappa), en lugar de los dos parámetros α y β.

Si definimos $\kappa = \frac{\alpha}{\beta}$, donde α es la tasa de transiciones y β es la tasa de transversiones ($\beta > 0$), podemos multiplicar al generador Q del modelo K2P por $1/\beta$ para obtener

$$Q' = \frac{1}{\beta}Q = \begin{pmatrix} -\frac{\alpha+2\beta}{\beta} & \frac{\alpha}{\beta} & 1 & 1 \\ \frac{\alpha}{\beta} & -\frac{\alpha+2\beta}{\beta} & 1 & 1 \\ 1 & 1 & -\frac{\alpha+2\beta}{\beta} & \frac{\alpha}{\beta} \\ 1 & 1 & \frac{\alpha}{\beta} & -\frac{\alpha+2\beta}{\beta} \end{pmatrix}$$

$$Q' = \begin{pmatrix} -(\kappa+2) & \kappa & 1 & 1 \\ \kappa & -(\kappa+2) & 1 & 1 \\ 1 & 1 & -(\kappa+2) & \kappa \\ 1 & 1 & \kappa & -(\kappa+2) \end{pmatrix} \tag{2.15}$$

La representación de la ecuación (2.15) es muy utilizada en la práctica porque le quita un grado de libertad al modelo K2P.

Como se indica en la sección (2.4.2), la tasa total de cambio del modelo K2P es $\alpha + 2\beta$. Entonces, si se desea recuperar los valores de α y β, podemos fijar la tasa total de cambio a 1[1] y de esta forma obtener el sistema de ecuaciones

$$\begin{cases} \frac{\alpha}{\beta} = \kappa \\ \alpha + 2\beta = 1 \end{cases}$$

de donde se obtiene $\alpha = \kappa\beta$ y $\beta = 1/(2+\kappa)$.

[1]Esto significa que esperamos observar 1 cambio por unidad de tiempo.

2.4.4 Modelo Kimura–3

También se puede definir el modelo Kimura de 3 parámetros o Kimura–3 de la siguiente manera.

La matriz de tasas Q tiene la siguiente forma,

$$Q = \begin{pmatrix} x & \beta & \gamma & \delta \\ \beta & x & \delta & \gamma \\ \gamma & \delta & x & \beta \\ \delta & \gamma & \beta & x \end{pmatrix} \tag{2.16}$$

con $y = -\beta - \gamma - \delta$. La tasa total de cambio en este caso es $\beta + \gamma + \delta$.

La matriz de Markov tiene la siguiente forma

$$M(t) = e^{Qt} = \begin{pmatrix} y & b(t) & c(t) & d(t) \\ b(t) & y & d(t) & c(t) \\ c(t) & d(t) & y & b(t) \\ d(t) & c(t) & b(t) & y \end{pmatrix} \tag{2.17}$$

con $y = 1 - b(t) - c(t) - d(t)$.

Notar que se hace una discriminación en las transversiones de tipo $A \Longleftrightarrow T$ y $G \Longleftrightarrow C$ asignándoles probabilidad $c(t)$.

Su vector de distribución de bases en la raíz al igual del modelo Kimura–2 es:

$$\mathbf{p}_\rho = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$$

En el siguiente teorema se deducen las fórmulas para los parámetros del modelo Kimura–3.

***Teorema* 2.3.** *El modelo Kimura de tres parámetros tiene como matriz de Markov a la matriz de la ecuación (2.17), donde*

$$b(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} + e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$

$$c(t) = \frac{1}{4}\left(1 + e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$

$$d(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} + e^{-2(\beta+\gamma)t}\right)$$

Demostración. Se utilizará el método de la diagonalización,

$$M(t) = e^{Qt} = V e^{\Lambda t} V^{-1}$$

donde Λ es una matriz diagonal que contiene los valores propios de la matriz Q de la ecuación (2.16) y V es una matriz que contiene sus vectores propios.

Primero se calculan los valores propios de Q.

$$p(\lambda) = |Q - \lambda I| = \begin{vmatrix} x-\lambda & \beta & \gamma & \delta \\ \beta & x-\lambda & \delta & \gamma \\ \gamma & \delta & x-\lambda & \beta \\ \delta & \gamma & \beta & x-\lambda \end{vmatrix}$$

Sea $w = x - \lambda = -(\beta + \gamma + \delta + \lambda)$.

$$p(\lambda) = |Q - \lambda I| = \begin{vmatrix} w & \beta & \gamma & \delta \\ \beta & w & \delta & \gamma \\ \gamma & \delta & w & \beta \\ \delta & \gamma & \beta & w \end{vmatrix}$$

El cálculo del anterior determinante nos ofrece la siguiente expresión para el polinomio característico de la matriz Q.

$$p(\lambda) = (\beta + \gamma - \delta - w)(\beta - \gamma + \delta - w)(\beta - \gamma - \delta + w)(\beta + \gamma + \delta + w)$$

Por lo tanto, los valores propios son:

1. $$\beta + \gamma - \delta - w = 0 \iff \beta + \gamma - \delta + (\beta + \gamma + \delta + \lambda) = 0$$
$$2\beta + 2\gamma + \lambda = 0$$
$$\lambda_1 = -2(\beta + \gamma)$$

2. $$\beta - \gamma + \delta - w = 0 \iff \beta - \gamma + \delta + (\beta + \gamma + \delta + \lambda)$$
$$2\beta + 2\delta + \lambda = 0$$
$$\lambda_2 = -2(\beta + \delta)$$

3. $$\beta - \gamma - \delta + w = 0 \iff \beta - \gamma - \delta - (\beta + \gamma + \delta + \lambda)$$
$$-2\gamma - 2\delta - \lambda = 0$$
$$\lambda_3 = -2(\gamma + \delta)$$

4. $$\beta + \gamma + \delta + w = 0 \iff \beta + \gamma + \delta - (\beta + \gamma + \delta + \lambda)$$
$$\lambda_4 = 0$$

Ahora calculamos los vectores propios.

$$Qv = \lambda v \iff (Q - \lambda I)v = 0$$

1. Asociado a $\lambda_1 = -2(\beta + \gamma)$. Primero calculamos $x - \lambda_1$

$$x - \lambda = -\beta - \gamma - \delta + 2\beta + 2\gamma = \beta + \gamma - \delta$$

$$\begin{pmatrix} \beta+\gamma-\delta & \beta & \gamma & \delta \\ \beta & \beta+\gamma-\delta & \delta & \gamma \\ \gamma & \delta & \beta+\gamma-\delta & \beta \\ \delta & \gamma & \beta & \beta+\gamma-\delta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

De donde se obtiene la ecuación,

$$(\beta+\gamma-\delta)x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = 0$$

$$\beta(x_1+x_2) + \gamma(x_1+x_3) + \delta(x_4-x_1) = 0$$

Por lo tanto, $x_1 = 1$, $x_2 = -1$, $x_3 = -1$, $x_1 = 1$ y

$$v_1 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

2. Asociado a $\lambda_2 = -2(\beta+\delta)$. Calculamos $x - \lambda_2$

$$x - \lambda_2 = -\beta-\gamma-\delta+2\beta+2\delta = \beta-\gamma+\delta$$

$$\begin{pmatrix} \beta-\gamma+\delta & \beta & \gamma & \delta \\ \beta & \beta-\gamma+\delta & \delta & \gamma \\ \gamma & \delta & \beta-\gamma+\delta & \beta \\ \delta & \gamma & \beta & \beta-\gamma+\delta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

De donde se obtiene la ecuación,

$$(\beta-\gamma+\delta)x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = 0$$

$$\beta(x_1+x_2) + \gamma(x_3-x_1) + \delta(x_1+x_4) = 0$$

Por lo tanto, $x_1 = 1$, $x_2 = -1$, $x_3 = 1$, $x_1 = -1$ y

$$v_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

3. Asociado a $\lambda_3 = -2(\gamma+\delta)$. Calculamos $x - \lambda_3$

$$x - \lambda_2 = -\beta-\gamma-\delta+2\gamma+2\delta = -\beta+\gamma+\delta$$

$$\begin{pmatrix} -\beta+\gamma+\delta & \beta & \gamma & \delta \\ \beta & -\beta+\gamma+\delta & \delta & \gamma \\ \gamma & \delta & -\beta+\gamma+\delta & \beta \\ \delta & \gamma & \beta & -\beta+\gamma+\delta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

De donde se obtiene la ecuación,

$$(-\beta+\gamma+\delta)x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = 0$$

$$\beta(x_2 - x_1) + \gamma(x_1 + x_3) + \delta(x_1 + x_4) = 0$$

Por lo tanto, $x_1 = 1$, $x_2 = 1$, $x_3 = -1$, $x_1 = -1$ y

$$v_3 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

4. Asociado a $\lambda_4 = 0$

$$v_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Luego de ordenar los vectores de tal forma que la matriz V sea simétrica, se obtiene:

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad \Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -2(\beta+\delta) & 0 & 0 \\ 0 & 0 & -2(\gamma+\delta) & 0 \\ 0 & 0 & 0 & -2(\beta+\gamma) \end{pmatrix}$$

Ahora se puede calcular $M(t) = e^{Qt} = Ve^{\Lambda t}V^{-1}$, donde

$$e^{\Lambda t} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-2(\beta+\delta)t} & 0 & 0 \\ 0 & 0 & e^{-2(\gamma+\delta)t} & 0 \\ 0 & 0 & 0 & e^{-2(\beta+\gamma)t} \end{pmatrix}$$

$$V^{-1} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & 1/4 & -1/4 & -1/4 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{pmatrix} = \frac{1}{4}V$$

Luego,

$$M(t) = \frac{1}{4}\left(Ve^{\Lambda t}V\right)$$

Sea $x = -2(\beta+\delta)t$, $y = -2(\gamma+\delta)t$ y $z = -2(\beta+\gamma)t$

$$M(t) = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-2(\beta+\delta)t} & 0 & 0 \\ 0 & 0 & e^{-2(\gamma+\delta)t} & 0 \\ 0 & 0 & 0 & e^{-2(\beta+\gamma)t} \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$M(t) = \frac{1}{4}\begin{pmatrix} 1+e^x+e^y+e^z & 1-e^x+e^y-e^z & 1+e^x-e^y-e^z & 1-e^x-e^y+e^z \\ 1-e^x+e^y-e^z & 1+e^x+e^y+e^z & 1-e^x-e^y+e^z & 1+e^x-e^y-e^z \\ 1+e^x-e^y-e^z & 1-e^x-e^y+e^z & 1+e^x+e^y+e^z & 1-e^x+e^y-e^z \\ 1-e^x-e^y+e^z & 1+e^x-e^y-e^z & 1-e^x+e^y-e^z & 1+e^x+e^y+e^z \end{pmatrix}$$

Por lo tanto, los parámetros del modelo Kimura–3 son:

$$b(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} + e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$

$$c(t) = \frac{1}{4}\left(1 + e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$

$$d(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} + e^{-2(\beta+\gamma)t}\right)$$

□

El modelo Kimura–3 tiene más motivaciones matemáticas que biológicas, pues biológicamente, no tiene mucho sentido hacer distinción entre los cuatro tipos de transversión debido a su estructura molecular.

2.4.5 El modelo GTR

El *General Time–Reversible Model* (GTR) es el modelo más general que tiene la cualidad de *reversibilidad en el tiempo*. Esta cualidad es muy útil para hacer la reconstrucción del árboles filogenéticos.

Reversibilidad en el tiempo para modelos evolutivos está acorde a la definición (1.19) de la sección (1.4). Como indica Felsenstein [5], la reversibilidad en el tiempo se puede interpretar de la siguiente manera:

> "La probabilidad de empezar con un nucleótido i en un extremo de la rama y que este evolucione en la base j en el otro extremo es la misma probabilidad de empezar con j y evolucionar en i."

La propiedad de Markov indica que "el futuro es independiente del pasado dado el estado presente", entonces la propiedad de *reversibilidad en el tiempo* permite intercambiar el pasado con el futuro y declarar que el "pasado es independiente del futuro dado el estado presente." Esta propiedad también significa que el proceso que va "hacia atrás" en el tiempo también es una cadena de Markov (estacionaria).

Extendiendo este concepto a la formulación continua de un modelo, para que la reversibilidad en el tiempo de cumpla, se necesita tener

$$\text{diag}(p)Q = Q^T\text{diag}(p) \tag{2.18}$$

Notar que los modelos JC69, Kimura–2 y Kimura–3 tiene la propiedad de reversibilidad, pues los tres modelos poseen matrices de tasas y matrices de Markov simétricas.

A continuación se define el Modelo General Reversible *GTR*, el cual fue introducido por primera vez en 1984 por Lanave et al.

***Definición* 2.4.** *Dado un árbol $\mathcal{T}^\rho$ con largos de ramas, se dice que el proceso evolutivo de una rama $e \in E(\mathcal{T}^\rho)$ está descrito por un modelo GTR si su modelo de Markov consta de los siguientes parámetros.*

1. *Un vector de distribución de estados del ancestro, denotado por*

$$\pi = (\pi_1, \pi_2, \pi_3, \pi_4) \tag{2.19}$$

tal que, valores arbitrarios de $\pi_1, \pi_2, \pi_3, \pi_4$ satisfacen la condición siguiente:

$$\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1 \tag{2.20}$$

2. *Una matriz de tasas Q tal que,*

$$Q = \begin{pmatrix} * & \pi_G\alpha & \pi_C\beta & \pi_T\gamma \\ \pi_G\alpha & * & \pi_C\delta & \pi_T\epsilon \\ \pi_C\beta & \pi_G\delta & * & \pi_T\eta \\ \pi_T\gamma & \pi_G\epsilon & \pi_C\eta & * \end{pmatrix} \tag{2.21}$$

donde $\alpha, \beta, \gamma, \delta, \epsilon, \eta$ son números reales mayores o iguales a cero y las entradas en la diagonal están escogidas de tal forma que las filas sumen cero.

2.4.6 Modelo F81

El modelo de Jukes–Cantor es el más simple, pero a la vez, el más restrictivo de todos los posibles modelos evolutivos.

Una de sus restricciones es la que fija las frecuencias estacionarias de las bases a $\frac{1}{4}$. Sin embargo, como se indica en la sección (2.4.5) en la definición del Modelo General Reversible en el tiempo (GTR), estas frecuencias no tienen que ser necesariamente iguales.

Entonces, si relajamos la restricción del modelo JC donde se requiere que $\pi_A = \pi_G = \pi_C = \pi_T = 0.25$ y permitimos que estas frecuencias tomen valores arbitrarios, es decir, $\pi_A \neq \pi_G \neq \pi_C \neq \pi_T$; se obtiene un modelo conocido como F81 (o F84). Este modelo ha sido usado en el programa PHYLIP de Felsenstein desde 1984 [5]. De manera similar, si relajamos la restricción de frecuencias estacionarias iguales para el modelo K2P, obtenemos el modelo HKY el cual fue introducido por HASEGAWA, KISHINO, and YANO [7].

3 Distancias basadas en modelos

El objetivo de este capítulo es determinar medidas de distancia evolutiva, se quiere determinar qué tan similares son dos taxones. Dada una colección de taxones y sus respectivas secuencias de ADN (alineadas), la manera más inmediata de determinar cuán diferentes son dos secuencias es contando la cantidad de sitios que tienen nucleótidos diferentes; sin embargo, teniendo un modelo evolutivo que ajusta bien los datos, se puede definir medidas de distancia más sofisticadas.

3.1 Distancia de Hamming

La forma más simple de calcular la distancia entre dos secuencias S_0 y S_1 está dada por la *distancia de Hamming*, la cual se define a continuación.

***Definición* 3.1** (Distancia de Hamming)**.** *Sean x y y dos secuencias de caracteres $\chi_1(x), \chi_2(x), \ldots, \chi_n(x)$ y $\chi_1(y), \chi_2(y), \ldots, \chi_n(y)$ respectivamente. La* distancia de Hamming *entre x y y se define por:*

$$\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta_{\chi_i(x), \chi_i(y)} \tag{3.1}$$

donde

$$\delta_{a,b} = \begin{cases} 0, & si \;\; a = b \\ 1, & si \;\; a \neq b \end{cases}$$

Entonces, la distancia de Hamming simplemente cuenta la cantidad de sitios en las secuencias x, y que tienen nucleótidos diferentes y esa cantidad es dividida para el número total de sitios en las secuencias.[1]

La distancia de Hamming también es conocida como *p–distancia* porque su valor es usualmente denotado por p o *distancia no corregida*. Se denomina "no corregida" porque esta distancia no toma en cuenta los cambios de base ocultos que pudieron ocurrir durante la evolución.

Por otro lado, teniendo un modelo probabilístico de evolución de ADN, se pueden definir distancias que tomen en cuenta estos cambios ocultos de base; a continuación se deducen fórmulas de distancias para determinados modelos evolutivos. Estas distancias también son llamadas "distancias corregidas" y son una pieza clave para poder hacer la reconstrucción de árboles filogenéticos.

[1] Siempre se trabaja con secuencias que tiene la misma cantidad de sitios o nucleótidos. Por esta razón tiene una matriz como dato de entrada, cuyas filas corresponden a los taxones y en las columnas se encuentran los nucleótidos que componen sus respectivas secuencias de ADN.

3.2 Distancia de Jukes–Cantor

En el capítulo anterior se introdujo el modelo de Jukes–Cantor, el cual tenía la siguiente matriz de tasas:

$$Q = \begin{pmatrix} -\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha \end{pmatrix}$$

y su respectiva matriz de Markov,

$$M(t) = e^{Qt} = \begin{pmatrix} 1-a & a/3 & a/3 & a/3 \\ a/3 & 1-a & a/3 & a/3 \\ a/3 & a/3 & 1-a & a/3 \\ a/3 & a/3 & a/3 & 1-a \end{pmatrix}$$

donde,

$$a = a(t) = \frac{3}{4}\left(1 - \exp\left\{-\frac{4\alpha t}{3}\right\}\right) \tag{3.2}$$

Para deducir la fórmula de la distancia del modelo de Jukes–Cantor, supongamos que se cuenta con una secuencia ancestral a la cual denotaremos por S_0 y una secuencia descendiente, denotada por S_1. Se utiliza la convención en la que las filas de $M(t)$ representan a la secuencia S_0 y las columnas a la secuencia S_1.

Se requiere encontrar una expresión para la tasa total de cambio αt, la cual se puede desarrollar despejando esta variable de la ecuación (3.2). Se tiene, entonces

$$\begin{aligned} a(t) &= \frac{3}{4}\left(1 - e^{-\frac{4\alpha t}{3}}\right) \\ \frac{4}{3}a(t) &= 1 - e^{-\frac{4\alpha t}{3}} \\ e^{-\frac{4\alpha t}{3}} &= 1 - \frac{4}{3}a(t) \\ \ln\left(^{-\frac{4\alpha t}{3}}\right) &= \ln\left(1 - \frac{4}{3}a(t)\right) \\ -\frac{4\alpha t}{3} &= \ln\left(1 - \frac{4}{3}a(t)\right) \\ \alpha t &= -\frac{3}{4}\ln\left(1 - \frac{4}{3}a(t)\right) \end{aligned}$$

Si S_0 y S_1 son secuencias de ADN y el modelo de Jukes–Cantor describe adecuadamente la evolución de S_0 en S_1, $a(t)$ puede estimarse utilizando la distancia de Hamming entre S_0 y S_1, sea $\hat{a}$ esta distancia. Entonces, la

distancia de Jukes–Cantor entre S_0 t S_1 se calcula por:

$$d_{JC}(S_0, S_1) = -\frac{3}{4}\ln\left(1 - \frac{4}{3}\hat{a}\right) \tag{3.3}$$

La distancia $d_{JC}(S_0, S_1)$ es una estimación del número total de sustituciones por sitio que han ocurrido durante la evolución.

3.3 Distancias Kimura

Se puede imitar los pasos de la deducción de la distancia de Jukes–Cantor con los otros modelos evolutivos y encontrar sus respectivas distancias.

***Lema* 3.1.** *La distancia para el modelo Kimura de dos parámetros es,*

$$d_{K_2}(S_1, S_2) = -\frac{1}{2}\ln\left[1 - 2\left(\hat{a} + \hat{b}\right)\right] - \frac{1}{4}\ln\left[1 - 4\hat{b}\right] \tag{3.4}$$

donde $\hat{a}$ y $\hat{b}$ son las estimaciones para las probabilidades de transición y transversión respectivamente.

Demostración. Se busca encontrar una expresión para $(\alpha + 2\beta)t$ la cual es la tasa total de sustituciones por sitio en un intervalo de tiempo t.

En el teorema (2.3) se demostró que las expresiones para los parámetros del modelo Kimura–2 son las siguientes,

(1) $\mathbf{a} = \mathbf{a}(t) = \frac{1}{4}\left(1 - 2\exp\{-2(\alpha + \beta)t\} + \exp\{-4\beta t\}\right)$

(2) $\mathbf{b} = \mathbf{b}(t) = \frac{1}{4}\left(1 - \exp\{-4\beta t\}\right)$

Primero despejamos la expresión βt de (2).

$$\mathbf{b}(t) = \frac{1}{4}\left(1 - \exp\{-4\beta t\}\right)$$

$$4\mathbf{b} = 1 - e^{-4\beta t}$$

$$e^{-4\beta t} = 1 - 4\mathbf{b}$$

$$-4\beta t = \ln(1 - 4\mathbf{b})$$

$$\beta t = -\frac{1}{4}\ln(1 - 4\mathbf{b})$$

Ahora se despeja $\alpha + \beta$ de la ecuación (1).

$$\mathbf{a} = \frac{1}{4}\left(1 - 2\exp\{-2(\alpha + \beta)t\} + \exp\{-4\beta t\}\right)$$

$$\mathbf{a} = \frac{1}{4}\left(1 - 2\exp\{-2(\alpha + \beta)t\} + 1 - 4\mathbf{b}\right)$$

$$\mathbf{a} = \frac{1}{2} - \frac{1}{2}e^{-2(\alpha+\beta)t} - \mathbf{b}$$

$$\frac{1}{2}e^{-2(\alpha+\beta)t} = \frac{1}{2} - (\mathbf{a}+\mathbf{b})$$
$$e^{-2(\alpha+\beta)t} = 1 - 2(\mathbf{a}+\mathbf{b})$$
$$-2(\alpha+\beta) = \ln\left[1-2(\mathbf{a}+\mathbf{b})\right]$$
$$\alpha t + \beta t = -\frac{1}{2}\ln\left[1-2(\mathbf{a}+\mathbf{b})\right]$$

Para obtener $(\alpha + 2\beta)t$ basta sumar $\beta t = -\frac{1}{4}\ln(1-4\mathbf{b})$ a la expresión anterior.

$$\alpha t + \beta t + \beta t = -\frac{1}{2}\ln\left[1-2(\mathbf{a}+\mathbf{b})\right] - \frac{1}{4}\ln(1-4\mathbf{b})$$
$$(\alpha + 2\beta)t = -\frac{1}{2}\ln\left[1-2(\mathbf{a}+\mathbf{b})\right] - \frac{1}{4}\ln(1-4\mathbf{b})$$

Sean $\hat{a}$ la estimación para la probabilidad $\mathbf{a}(t)$ y $\hat{b}$ la estimación para $\mathbf{b}(t)$. Entonces la distancia Kimura–2 es,

$$d_{K_2}(S_1,S_2) = -\frac{1}{2}\ln\left[1-2\left(\hat{a}+\hat{b}\right)\right] - \frac{1}{4}\ln\left[1-4\hat{b}\right]$$

□

Siguiendo la misma lógica, se puede deducir la fórmula de la distancia del modelo Kimura–3.

***Lema* 3.2.** *Sean $\hat{b}$, $\hat{c}$ y $\hat{d}$ las estimaciones para las probabilidades $b(t)$, $c(t)$ y $d(t)$ del modelo Kimura–3, respectivamente. Entonces, la distancia entre dos secuencias S_1 y S_2 cuyo proceso evolutivo puede ser modelado por un modelo Kimura de tres parámetros tiene la siguiente forma:*

$$d_{K_3}(S_1,S_2) = -\frac{1}{4}\left[\ln\left(1-2\hat{b}-2\hat{c}\right) + \ln\left(1-2\hat{b}-2\hat{d}\right) + \ln\left(1-2\hat{c}-2\hat{d}\right)\right] \tag{3.5}$$

Demostración. Se busca encontrar la expresión para la tasa total de mutación $(\beta+\gamma+\delta)t$. Para logar esto, se utilizará las fórmulas del teorema (2.4)

$$b = b(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} + e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$
$$c = c(t) = \frac{1}{4}\left(1 + e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} - e^{-2(\beta+\gamma)t}\right)$$
$$d = d(t) = \frac{1}{4}\left(1 - e^{-2(\beta+\delta)t} - e^{-2(\gamma+\delta)t} + e^{-2(\beta+\gamma)t}\right)$$

Sean $x = 2(\beta+\delta)t$, $y = 2(\gamma+\delta)t$ y $z = 2(\beta+\gamma)t$

$$b = 1/4(1-e^{-x}+e^{-y}) - e^{-z}) \iff 4b - 1 = -e^{-x} + e^{-y}) - e^{-z}$$
$$c = 1/4(1+e^{-x}-e^{-y}) - e^{-z}) \iff 4c - 1 = e^{-x} - e^{-y}) - e^{-z}$$

$$d = 1/4(1 - e^{-x} + e^{-y}) + e^{-z}) \iff 4d - 1 = -e^{-x} + e^{-y}) + e^{-z}$$

$$\begin{array}{ll} (1) & 4b - 1 = -e^{-x} + e^{-y}) - e^{-z} \\ (2) & 4c - 1 = e^{-x} - e^{-y}) - e^{-z} \\ (3) & 4d - 1 = -e^{-x} + e^{-y}) + e^{-z} \end{array}$$

Sumamos $(1) + (2)$, $(1) + (3)$ y $(2) + (3)$.

$$\begin{array}{ll} (1) + (2) & 4b + 4c - 2 = -2e^{-z} \\ (1) + (3) & 4b + 4d - 2 = -2e^{-x} \\ (2) + (3) & 4c + 4d - 2 = -2e^{-y} \end{array}$$

Tomando logaritmos a ambos lados de cada ecuación se tiene,

$$ln(-2e^{-z}) = ln(1/2) - z = ln(4b + 4c - 2)$$

$$ln(-2e^{-x}) = ln(1/2) - x = ln(4b + 4d - 2)$$

$$ln(-2e^{-y}) = ln(1/2) - y = ln(4c + 4d - 2)$$

Despejando x, y, z, obtenemos:

$$z = ln(1/2) - ln(4b + 4c - 2)$$

$$x = ln(1/2) - ln(4b + 4d - 2)$$

$$z = ln(1/2) - ln(4c + 4d - 2)$$

Reemplazando las expresiones para x, y, z, tenemos:

$$\begin{array}{ll} (1') & (2\beta + \delta)t = ln(1/2) + ln(2) - ln(1 - 2b - 2d) \\ (2') & (2\gamma + \delta)t = ln(1/2) + ln(2) - ln(1 - 2c - 2d) \\ (3') & (2\beta + \gamma)t = ln(1/2) + ln(2) - ln(1 - 2b - 2c) \end{array}$$

Finalmente, sumando las expresiones $(1')$, $(2')$ y $(3')$ se obtiene la expresión para la distancia:

$$(4\beta + 4\gamma + 4\delta)t = -\left[ln(1 - 2b - 2d) + ln(1 - 2c - 2d) + ln(1 - 2b - 2c)\right]$$

$$(\beta + \gamma + \delta)t = -\frac{1}{4}\left[ln(1 - 2b - 2d) + ln(1 - 2c - 2d) + ln(1 - 2b - 2c)\right]$$

$$d_{K_3}(S_1, S_2) = -\frac{1}{4}\left[ln(1 - 2\hat{b} - 2\hat{d}) + ln(1 - 2\hat{c} - 2\hat{d}) + ln(1 - 2\hat{b} - 2\hat{c})\right]$$

□

3.4 Distancias Log-Det

Una condición importante para poder utilizar las distancias definidas en la sección anterior, es que el proceso evolutivo de las secuencias comparadas pueda ser modelado por el respectivo modelo del cual se deriva la distancia.

Sin embargo, existen casos en donde esta condición no necesariamente se cumple y se debe recurrir al Modelo General de Markov. Por esta razón es necesario definir medidas de distancia para el MGM.

Estas medidas de distancia deberán satisfacer las siguientes condiciones para ser consideradas válidas.

Sean S_0 y S_1 dos secuencias de ADN, la distancia $d(S_0, S_1)$ debe satisfacer las siguientes propiedades:

1. $d(S_0, S_1) \geq 0$
2. $d(S_0, S_1) = 0 \Longleftrightarrow S_0 = S_1$
3. Si S_2 es la secuencia de ADN de un nodo que se encuentra en el camino entre los nodos que tienen como secuencias de ADN a S_0 y S_1, entonces

$$d(S_0, S_1) = d(S_0, S_2) + d(S_2, S_1)$$

Las propiedades anteriores son muy parecidas a las propiedades de una métrica. La propiedad (4) es llamada *aditividad* y simplemente indica que las distancias individuales en un linaje deben sumar el mismo resultado que la distancia total. [3]

A continuación se define la distancia *Log-Det* o *paralinear*, la cual permite calcular la distancia entre dos secuencias (infinitas) producidas de acuerdo a un MGM.

***Definición* 3.2.** *Sea $\hat{F}$ la matriz de frecuencias de dimensión 4×4, obtenida al comparar los sitios entre un par de secuencias S_0 y S_1. La entrada (i, j) de $\hat{F}$ es la proporción de sitios con base i en S_0 que cambiaron a j en S_1.*

Sean $f_0 = (f_0^A, f_0^G, f_0^C, f_0^T)$ y $f_1 = (f_1^A, f_1^G, f_1^C, f_1^T)$ los respectivos vectores de frecuencias de la secuencias S_0 y S_1 los cuales pueden obtenerse marginalizando $\hat{F}$ por filas y columnas. Se define

$$g_0 = \prod_{i=A}^{T} f_0^i \quad y \quad g_1 = \prod_{i=A}^{T} f_1^i.$$

La distancia paralinear *o* Log-Det *se define como sigue:*

$$d(S_0, S_1) = -\frac{1}{4}\left[\ln\left(det\left(\hat{F}\right)\right) - \frac{1}{2}\ln\left(g_0 g_1\right)\right] \tag{3.6}$$

La distancia paralinear o Log–Det es muy útil en el estudio del ADN, [1] indican las siguientes ventajas:

1. Al estar basada en el modelo más general de sustitución de nucleótidos, es decir el MGM, la distancia paralinear puede usarse incluso si la matriz de tasas varía a lo largo y entre linajes.

2. Bajo la suposición de la misma tasa de sustitución entre sitios, la distancia Log–Det es muy útil para reconstrucción filogenética cuando las frecuencias de los nucleótidos son no estacionarias.
3. Los largos de ramas pueden ser estimados en términos de la distancia paralinear.
4. La distancia Log–Det es útil para probar la hipótesis del *reloj molecular* bajo frecuencias no estacionarias.

Cabe indicar que el uso de la distancia Log-Det o paralinear está justificado cuando las secuencias analizadas tienen diferencias significativas en la composición de las bases, o cuando existe razón para dudar que el proceso de sustitución sea el mismo en todas las ramas del árbol. [3]

La distancia paralinear o Log–Det está basada en el modelo más general de sustitución de nucleótidos, sin embargo, siempre se recomienda emplear el modelo más restrictivo y su respectiva distancia, con el objetivo de evitar una posible sobre–parametrización de los datos.

3.5 Ejemplos

Para ilustrar el uso de las distancias deducidas en este capítulo, consideremos el siguiente ejercicio.

Se simuló 400 bases de un par de secuencias ancestral y descendiente de acuerdo a un modelo JC69. La comparación de los sitios alineados dio como resultado la matriz de frecuencias de la tabla (3.1), donde las filas representan a la secuencia descendiente S_1 y las columnas a la secuencia ancestral S_0

$S_1 \vert S_0$	A	G	C	T
A	90	3	3	2
G	3	79	8	2
C	3	4	96	5
T	2	1	3	94

TABLA 3.1: Tabla de frecuencias

a) Distancia Jukes–Cantor.

$$d_{JC} = -\frac{3}{4}\ln\left(1 - \frac{4}{3}\hat{a}\right)$$

El valor de $\hat{a}$ es simplemente la distancia de Hamming entre S_0 y S_1. Este valor se obtiene sumando los elementos fuera de la diagonal de la tabla (3.1).

$$\hat{a} = \frac{1}{400}\sum \text{ elementos fuera de la diagonal}$$

$$\hat{a} = \frac{3+3+2+3+8+2+3+4+5+2+1+3}{400}$$

$$\hat{a} = \frac{39}{400} = 0.0975$$

Por lo tanto, la distancia de Jukes-Cantor es:

$$d_{JC} = -\frac{3}{4}\ln\left(1 - \frac{4}{3} \times \frac{39}{400}\right)$$

$$d_{JC} = 0.1044$$

b) Distancia Kimura–2

$$d_{K_2} = -\frac{1}{2}\ln\left[1 - 2\left(\hat{a} + \hat{b}\right)\right] - \frac{1}{4}\ln\left[1 - 4\hat{b}\right]$$

En este caso se tiene que calcular $\hat{a}$ que es la proporción de cambios de tipo transición y $\hat{b}$ que son los cambios tipo transversión.

$$\hat{a} = \frac{\text{cambios tipo transición}}{400}$$

$$\hat{a} = \frac{3+3+5+3}{400} = \frac{7}{200} = 0.035$$

$$\hat{b} = \frac{\text{cambios tipo transversión}}{400}$$

$$\hat{b} = \frac{3+2+8+2+2+4+5+1}{400} = \frac{27}{400} = 0.0675$$

Entonces la distancia Kimura–2 es:

$$d_{K_2} = -\frac{1}{2}\ln\left[1 - 2\left(0.035 + 0.0675\right)\right] - \frac{1}{4}\ln\left[1 - 4 \times 0.0675\right]$$

$$d_{K_2} = 0.1933$$

c) Distancia Kimura–3

$$d_{K_3}(S_1, S_2) = -\frac{1}{4}\left[\ln\left(1 - 2\hat{b} - 2\hat{c}\right) + \ln\left(1 - 2\hat{b} - 2\hat{d}\right) + \ln\left(1 - 2\hat{c} - 2\hat{d}\right)\right]$$

$$\hat{b} = \frac{\sum \text{transiciones}}{400}$$

$$\hat{b} = \frac{3+3+5+3}{400} = \frac{14}{400} = 0.035$$

Ahora, el valor de $\hat{c}$ se calcula sumando las frecuencias de transversiones tipo AC, GT, CA y TG.

$$\hat{c} = \frac{3+2+2+1}{400} = \frac{8}{400} = 0.02$$

Finalmente, el valor de $\hat{d}$ se calcula sumando las frecuencias de transversiones tipo AT, GC, CG y TA.

$$\hat{c} = \frac{2+8+4+5}{400} = \frac{19}{400} = 0.0475$$

Luego, la distancia Kimura–3 es:

$$d_{K_3}(S_1, S_2) = -\frac{1}{4}\left[\ln\left(1 - 2\times\frac{14}{400} - 2\times\frac{8}{400}\right) + \right.$$

$$\left.\ln\left(1 - 2\times\frac{14}{400} - 2\times\frac{19}{400}\right) + \ln\left(1 - 2\times\frac{8}{400} - 2\times\frac{19}{400}\right)\right]$$

$$d_{K_3}(S_1, S_2) = 0.1104$$

4 Reconstrucción de árboles filogenéticos por máxima verosimilitud

Uno de los métodos más populares para hacer reconstrucción de árboles filogenéticos es la inferencia por Máxima Verosimilitud.

En el presente capítulo, se explicará de manera general cómo funciona este método.

Primero revisaremos algo de la teoría de Estimadores de Máxima Verosimilitud [*MLE*], luego se explicará cómo esta teoría puede ser usada para inferir un árbol de máxima verosimilitud, definiendo una función de máxima verosimilitud para un conjunto de datos representado por un alineamiento de secuencias de ADN.

Posteriormente, se presentará el Algoritmo *Pruning* de Felsenstein. Este es un algoritmo que permite calcular, de manera eficiente, el valor de la verosimilitud de un alineamiento dados una topología con largos de ramas y un modelo evolutivo.

Para finalizar, explicaremos los pasos que sigue el método de Máxima Verosimilitud para encontrar el árbol con largos de ramas y el conjunto de parámetros para el modelo evolutivo que maximizan la función de verosimilitud de los datos.

4.1 La función de verosimilitud

Antes de pasar a la aplicación del método de Máxima Verosimilitud para la reconstrucción de árboles filogenéticos, revisemos algunas definiciones y propiedades de los estimadores de máxima verosimilitud o MLE por sus siglas en inglés *Maximum Likelihood Estimation*.

Supongamos que deseamos estimar un parámetro desconocido $\theta \in \Theta$ de un conjunto de datos D.

El conjunto de datos D está compuesto por n observaciones, $x_1, x_2, \ldots, x_n$ independientes e idénticamente distribuidas (i.i.d) con función de distribución desconocida pero con función de densidad o frecuencias $f_\theta(\cdot)$

En palabras simples, la verosimilitud del parámetro θ, denotada $\mathcal{L}(\theta)$ puede verse como una "densidad conjunta" de los datos en función del parámetro θ [23] . Esto es,

$$\mathcal{L}(\theta) : \Theta \longrightarrow [0, \infty)$$

Además, gracias a la condición de independencia, se tiene la siguiente definición.

***Definición* 4.1.** *Se define la función de verosimilitud del parámetro θ dado el conjunto de datos D, compuesto por n observaciones i.i.d., mediante la expresión:*

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(x_i|\theta) \tag{4.1}$$

donde $f(\cdot|\theta)$ es la función de densidad o frecuencias de los datos.

Es muy usual expresar la verosimilitud en términos del logaritmo, por eso se define

$$\ell(\theta) = \log\left(\mathcal{L}(\theta)\right) \tag{4.2}$$

como el *logaritmo de la verosimilitud*. Luego, la expresión (4.1) puede escribirse como:

$$\ell(\theta) = \log\left(\mathcal{L}(\theta)\right) = \sum_{i=1}^{n} f(x_i|\theta). \tag{4.3}$$

Por otro lado, al momento de hacer inferencia por máxima verosimilitud, el objetivo es encontrar el valor de θ que maximiza la función de verosimilitud.

***Definición* 4.2.** *El* estimador de máxima de verosimilitud*, denotado $\hat{\theta}_{MLE}$, es el valor que maximiza la función de verosimilitud. Esto es,*

$$\hat{\theta}_{MLE} = arg\ máx_{\theta\in\Theta}\ \ell(\theta) \tag{4.4}$$

4.1.1 Ejemplo

Supongamos que $x_1, x_2, \ldots, x_n$ son variables aleatorias i.i.d. con distribución de Poisson de parámetro $\lambda > 0$.

La función de verosimilitud de λ dadas $x_1, x_2, \ldots, x_n$ es,

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \tag{4.5}$$

Al tomar el logaritmo a ambos lados de la expresión (4.5) se obtiene la expresión para el logaritmo de la verosimilitud:

$$\ell(\theta) = \log(\mathcal{L}(\lambda)) = \log\left[\prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right]$$

$$\ell(\theta) = -n\lambda + ln(\lambda)\sum_{i=1}^{n} x_i - \sum_{i=0}^{n} ln(x_i!) \tag{4.6}$$

Para encontrar el estimador de máxima verosimilitud, derivamos la expresión (4.6) respecto a λ, igualamos a 0 y despejamos λ.

$$\frac{d}{d\lambda}\ell(\lambda) = -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i$$

$$-n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i = 0$$

$$\frac{1}{\lambda}\sum_{i=1}^{n} x_i = n$$

Suponiendo que $\sum_{i=1}^{n} x_i > 0$, se tiene

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$$

Por último, verificamos si λ es un máximo o un mínimo derivando por segunda vez $\ell(\lambda)$.

$$\frac{d^2}{d\lambda^2}\ell(\lambda) = -\frac{1}{\lambda^2}\sum_{i=1}^{n} x_i \ < 0$$

Como la expresión anterior es siempre negativa pues, $\lambda > 0$ y $\sum_{i=1}^{n} x_i > 0$, el estimador de máxima verosimilitud es

$$\hat{\lambda}_{MLE} = \bar{x}$$

4.2 Propiedades asintóticas de los MLE

Sean $X_1, X_2, \ldots, X_n$, n variables aleatorias independientes e idénticamente distribuidas con función de verosimilitud $\mathcal{L}(\theta)$, donde $\theta \in \mathbb{R}$. Los estimadores de máxima verosimilitud poseen propiedades útiles en la práctica si la función $\ell(\theta)$ satisface las siguientes condiciones, las cuales usualmente se denominan *condiciones de regularidad*:

P1) El espacio de los parámetros Θ es un subconjunto de $\mathbb{R}$.

P2) El conjunto $A = \{x : \mathcal{L}(x;\theta) > 0\}$ no depende de θ.

P3) La función $\ell(\theta)$ es tres veces continuamente diferenciable respecto a θ, para todo $x \in A$.

P4) $\mathbb{E}_\theta[\ell'(\theta)]$=0 para todo θ y $\text{Var}_\theta[\ell'(\theta)] = I(\theta)$, donde $0 < I(\theta) < \infty$ para todo θ. $I(\theta)$ es llamada *información esperada de Fisher*.

P5) $\mathbb{E}_\theta[\ell''(\theta)] = J(\theta)$, donde $0 < J(\theta) < \infty$ para todo θ. $J(\theta)$ se conoce como *información observada de Fisher*.

P6) La función $\ell'''(\theta)$ es acotada en una vecindad de θ.

[10, pág. 256]

El símbolo $\mathbb{E}_\theta$ representa la esperanza de una variable aleatoria y

$$\mathbb{E}_\theta[\ell'(\theta)] = \int_A \ell'(x;\theta)\mathcal{L}(x;\theta)dx$$

Sea $\hat{\theta}_{MLE}$ el estimador de máxima verosimilitud de la función $\mathcal{L}(\theta)$.bajo las condiciones (P1) a (P6), $\hat{\theta}_{MLE}$ posee las siguientes propiedades, las cuales se enuncian sin demostración:

1. Es **consistente**, es decir, $\hat{\theta}_{MLE} \xrightarrow{p} \theta_*$ donde θ_* es el verdadero valor del parámetro θ y $\xrightarrow{p}$ significa convergencia en probabilidad.

2. Es **equivariante**, esto es, si $\hat{\theta}_{MLE}$ es el EMV de θ, entonces $g(\hat{\theta}_{MLE})$ es el EMV de $g(\theta)$.

3. Es **asintóticamente normal**, es decir,

$$\sqrt{n}\frac{(\hat{\theta}_{MLE}) - \theta_*)}{\hat{se}} \rightsquigarrow \mathcal{N}(0,1)$$

   donde $\hat{se}$ es la desviación estándar de $\hat{\theta}_{MLE}$ y $\rightsquigarrow$ significa convergencia en distribución.

4. Es **asintóticamente optimal** o **eficiente**. En palabras simples, esto significa que de entre todos los estimadores que se "comportan bien", $\hat{\theta}_{MLE}$ tiene la menor varianza.[23]

4.3 Aplicación a la reconstrucción de arboles filogenéticos

En el caso de la reconstrucción de árboles filogenéticos, se quiere determinar una topología, un conjunto de largos de ramas y parámetros del modelo evolutivo que maximizan la probabilidad de observar las secuencias de ADN de los taxones.

Sea D el conjunto de secuencias de ADN que han sido obtenidas a partir de los taxones. A este D también se le denomina *alineamiento*.

La función de verosimilitud es entonces, la probabilidad condicional de los datos D, dado un modelo evolutivo θ y un árbol τ con largos de ramas $\{t_e : e \in \mathbb{E}(\tau)\}$.

El modelo evolutivo está compuesto por un conjunto de parámetros: tasas de transición/transversión, frecuencias estacionarias de las bases, etc. [19]

La función de verosimilitud tiene la siguiente forma:

$$\mathcal{L}(\tau,\theta) = \mathbb{P}(D|\tau,\theta) = \mathbb{P}(\text{alineamiento}|\text{árbol, modelo evolutivo}) \tag{4.7}$$

Para poder calcular $\mathcal{L}(\tau,\theta)$ se debe hacer las siguientes suposiciones.

ML1) La *evolución en diferentes sitios* del árbol dado es *independiente*.

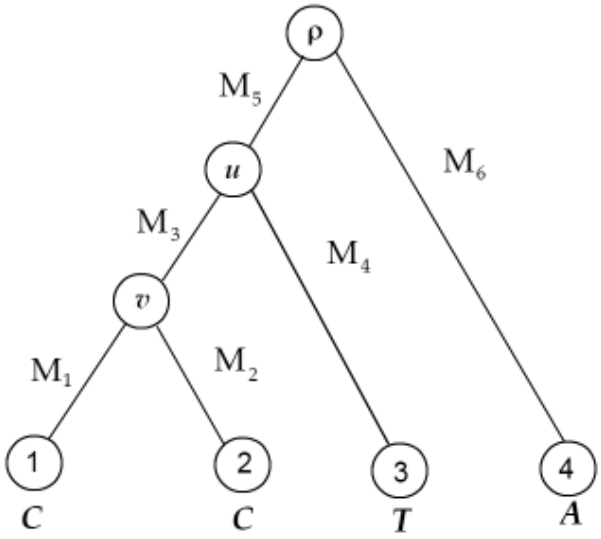


FIGURA 4.1: Árbol de ejemplo

ML2) La evolución en diferentes linajes es independiente. [5]

Un linaje en un árbol filogenético puede verse simplemente como un camino en dicho árbol.

Teniendo las dos suposiciones anteriores, la función de verosimilitud $\mathcal{L}(\tau, \theta)$ puede expresarse como el producto de las probabilidades de observar los distintos patrones de nucleótidos que aparecen en cada columna de la matriz de datos D dado un árbol con largos de ramas y un modelo evolutivo:

$$\mathcal{L}(\tau, \theta) = \mathbb{P}(D|\tau, \theta) = \prod_{j=1}^{n} \mathbb{P}(D^{(j)}|\tau, \theta) \tag{4.8}$$

donde n es el tamaño de las secuencias en D y $D^{(j)}$ representa su j–ésima columna.[1]

Tomando el logaritmo en ambos lados de la ecuación (4.8), se obtiene una suma de probabilidades.

$$\ell(\tau, \theta) = \sum_{j=1}^{n} \mathbb{P}(D^{(j)}|\tau, \theta) \tag{4.9}$$

De ahora en adelante, a menos que se enuncie lo contrario, se trabajará con $\ell(\tau, \theta)$ y cuando se mencione a la función de verosimilitud se debe entender que se trata de la expresión en la ecuación (4.9).

El objetivo ahora es calcular $\mathbb{P}(D^{(j)}|\tau, \theta)$. Una forma de lograr esto es considerando todos los patrones de bases en cada uno de los nodos internos. Por ejemplo, supongamos que queremos calcular la probabilidad de observar el patrón de las hojas en el árbol de la figura (4.1) y para simplificar supongamos que conocemos de antemano las matrices de Markov que modelan la evolución en cada una de sus ramas.

[1]Recordar que las secuencias de ADN de los taxones son las filas de la matriz D.

Para encontrar la probabilidad de obtener el patrón (C, C, T, A), tendíamos que sumar las probabilidades de todas las posibles combinaciones de nucleótidos desde la raíz hasta las hojas. Esto es,

$$\mathbb{P}(D^{(j)}|\tau, \theta) = \sum_{\rho} \sum_{u} \sum_{v} \mathbb{P}((C, C, T, A), \rho, u, v|\tau, \theta) \tag{4.10}$$

Haciendo uso de la suposición de independencia entre linaje, la probabilidad en (4.10) puede descomponerse en el producto de probabilidades condicionales, de la siguiente manera:

$$\mathbb{P}((C, C, T, A), \rho, u, v|\tau, \theta) = \mathbb{P}(\rho) \cdot \mathbb{P}(u|\rho, M_5) \cdot \mathbb{P}(A|\rho, M_6) \cdot$$

$$\mathbb{P}(v|u, M_3) \cdot \mathbb{P}(T|u, M_4) \cdot \mathbb{P}(C|v, M_1) \cdot \mathbb{P}(C|v, M_2) \tag{4.11}$$

Como se puede observar, la complejidad de la fórmula es mayor si el número de nodos internos aumenta, pero lo que hace a este método poco eficiente, es que la cantidad de sumandos en la expresión (4.10) crece exponencialmente con el número de nodos internos. En efecto, en un árbol binario perfecto con h nodos hoja, la cantidad de nodos internos (que no son hojas) es $h - 1$ y como se tiene cuatro posibles nucleótidos para cada nodo interno, la cantidad de sumandos en la expresión (4.10) es 4^{h-1}.

4.4 Algoritmo Pruning de Felsenstein

Este es un algoritmo de programación dinámica, introducido por Joseph Felsenstein, que calcula de manera eficiente la probabilidad $\mathbb{P}(D^{(j)}|\tau, \theta)$.

Para ilustrar este algoritmo, consideremos nuevamente el árbol de la figura (4.1). Supongamos que sus matrices de Markov pertenecen a la familia de los modelos reversibles en el tiempo.

El algoritmo empieza analizando el árbol desde las hojas y va avanzando hacia la raíz. Básicamente calcula vectores que contienen las probabilidades condicionales de observar un nucleótido dada la información que se tenga en el nodo anterior. Notar que la topología (τ), modelo evolutivo (θ) y sitio del alineamiento (j) están fijos.

Para empezar a cada hoja h del árbol se le asignan vectores binarios c_h tales que la componente $c_h(i)$, con $i \in \{A, G, C, T\}$ es:

$$c_h(i) = \begin{cases} 1 & \text{si la base en } \ h \ \text{ es igual a } \ i \\ 0 & \text{caso contrario} \end{cases} \tag{4.12}$$

Por ejemplo, para el árbol de la figura (4.1), los respectivos vectores binarios de las hojas 1, 2, 3 y 4 son los siguientes:

$$c_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad c_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad c_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad c_4 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Notar que se está utilizando el orden (A, G, C, T) para los nucleótidos.

El siguiente paso es calcular la probabilidad condicional de observar A, C, G o T en el nodo v dado que en sus descendientes se observó el patrón (C, C). Llamemos a este vector c_v y a sus componentes $c_v(i)$, donde $i \in \{A, G, C, T\}$. La primera componente, $c_v(A)$ se calcula de la siguiente manera,

$$c_v(A) = [M_1(A,A) \cdot 0 + M_1(A,G) \cdot 0 + M_1(A,C) \cdot 1 + M_1(A,T) \cdot 0] \cdot$$

$$[M_2(A,A) \cdot 0 + M_2(A,G) \cdot 0 + M_2(A,C) \cdot 1 + M_2(A,T) \cdot 0].$$

La siguiente componente $c_v(G)$ se calcula con,

$$c_v(G) = [M_1(G,A) \cdot 0 + M_1(G,G) \cdot 0 + M_1(G,C) \cdot 1 + M_1(G,T) \cdot 0] \cdot$$

$$[M_2(G,A) \cdot 0 + M_2(G,G) \cdot 0 + M_2(G,C) \cdot 1 + M_2(G,T) \cdot 0].$$

De manera similar se calculan las probabilidades $c_v(C)$ y $c_v(T)$.

Sin embargo, es más fácil expresar estas operaciones utilizando notación matricial. Como indican Allman y Rhodes [3], las fórmulas anteriores pueden calcularse de manera más directa definiendo los vectores:

$$w_1^v = M_1 c_1 \quad \text{y} \quad w_2^v = M_2 c_2.$$

Luego, el vector c_v se calcula multiplicando los vectores elemento por elemento, es decir:

$$c_v(i) = w_1^v(i) \cdot w_2^v(i), \quad \text{para } \ i \in \{A, G, C, T\}.$$

Haciendo un razonamiento similar, se puede calcular el vector de probabilidades del nodo interno u, en este caso:

$$w_1^u = M_3 c_v \quad \text{y} \quad w_2^u = M_4 c_3.$$

Luego, el vector de probabilidades de la raíz c_ρ se calcula con,

$$w_1^\rho = M_5 c_u \quad \text{y} \quad w_2^\rho = M_6 c_4.$$

Finalmente, para obtener la probabilidad $\mathbb{P}(D^{(j)}|\tau, \theta) = \mathbb{P}((C, C, T, A)|\tau, \theta)$, se utiliza la distribución estacionaria π_ρ del modelo y se tiene:

$$\mathbb{P}(D^{(j)}|\tau, \theta) = \pi_\rho^T \cdot c_\rho \tag{4.13}$$

$$\mathbb{P}(D^{(j)}|\tau,\theta) = \pi_\rho(A)\cdot c_\rho(A) + \pi_\rho(G)\cdot c_\rho(G) + \pi_\rho(C)\cdot c_\rho(C) + \pi_\rho(T)\cdot c_\rho(T)$$

Notar que esta expresión es una media ponderada de las cuatro bases.

Repitiendo este mismo procedimiento para todos los sitios en el alineamiento D, se puede calcular el valor de la verosimilitud del árbol usando la ecuación (4.9).

***Observación* 4.1.** *Este algoritmo funciona gracias a la suposición de independencia del proceso evolutivo entre las ramas del árbol, es decir, entre linajes. Esto permite hacer muchas de las multiplicaciones, además se puede ver que el vector c_ρ está bien calculado. Consideremos la i–ésima componente del vector c_ρ, primero notar que:*

$$w_1^\rho(i) = \sum_{j=1}^{4} M_5(i,j)c_4(j) = \sum_{j=1}^{4} \mathbb{P}(u=j|\rho=i)\mathbb{P}(S_1=C, S_2=C, S_3=T|u=j)$$

$$\mathbb{P}(S_1=C, S_2=C, S_3=T|\rho=i)$$

donde S_1, S_2, S_3 y S_4 son las variables aleatorias que representan un determinado sitio en las respectivas secuencias de ADN de los taxones.

Por otro lado, $w_2^\rho(i) = \mathbb{P}(S_4=A|\rho=i)$, luego

$$c_\rho(i) = \mathbb{P}(S_1=C, S_2=C, S_3=T|\rho=i)\cdot\mathbb{P}(S_4=A|\rho=i)$$

$$c_\rho(i) = \mathbb{P}(S_1=C, S_2=C, S_3=T, S_4=A|\rho=i) \tag{4.14}$$

Se puede ver que el esfuerzo para el cálculo de la verosimilitud de un árbol se reduce si los patrones de $D^{(j)}$ se repiten, pues es necesario calcular una sola vez la probabilidad $\mathbb{P}D^{(j)}|\tau,\theta)$.

En general, para un árbol con h taxones los cuales tienen secuencias[2] tamaño N, el algoritmo Pruning ejecuta $N(h-1)b^2$ operaciones, donde b es el número de bases posibles [5].

Si las secuencias son de ADN, $b=4$ y si son proteínas, $b=20$

4.5 Método de reconstrucción por Máxima Verosimilitud

Como se indicó al inicio de la sección (4.3), el objetivo de la Reconstrucción por Máxima Verosimilitud es encontrar la topología con largos de ramas y el conjunto de parámetros θ para el modelo evolutivo que maximizan la función de verosimilitud.

Dados un conjunto de taxones y un alineamiento D el cual puede contener sus secuencias de ADN, Allman y Rhodes [3] resumen el método de reconstrucción por Máxima Verosimilitud de la siguiente manera:

1. Contar el número de patrones $D(j)$ que se repiten en el alineamiento.

[2]Pueden ser secuencias de ADN o de proteínas.

2. Considerar todos los posibles árboles τ que puedan relacionar al grupo de taxones.

3. Para cada árbol τ construir la función de verosimilitud y asignar posibles valores para los parámetros, estos son: frecuencias estacionarias de las bases π_ρ, largos de ramas $\{t_e : e \in \mathbb{E}(\tau)\}$ y matrices de Markov para cada rama.

   Usualmente se asume que la evolución en todas la ramas está dominada por el mismo modelo evolutivo, el cual está representado por su respectiva matriz generadora infinitesimal Q. De esta forma, como se explica en el capítulo (2), las matrices de Markov de cada rama pueden calcularse por la expresión

   $$M_e = e^{Qt_e} \quad \text{con} \quad e \in \mathbb{E}(\tau)$$

   Entonces solo se necesita asignar valores a las tasas de la matriz Q.

4. Para cada función de verosimilitud construida en el paso (3) (del árbol τ), calcular los valores numéricos de los parámetros π_ρ, $\{t_e : e \in \mathbb{E}(\tau)\}$ y Q que la maximizan.

5. Por último, escoger la topología y conjunto de parámetros numéricos que produce la mayor verosimilitud de todas.

Observación **4.2.** El paso (2) del método de reconstrucción por Máxima Verosimilitud es el que más esfuerzo computacional requiere, porque el espacio de árboles crece de manera super–exponencial con el número de hojas o taxones de los árboles. En efecto, si un árbol tiene $N \geq 3$ hojas, la cantidad posible de árboles bifurcados y enraizados está dada por la siguiente expresión,

$$\frac{(2N-3)!}{2^{N-2}(N-2)!}$$

Por ejemplo, para un árbol con $N = 5$ taxones existen 105 posibles árboles bifurcados y enraizados. Pero el aumentar la cantidad de taxones a $N = 10$, hace que el número de posibles árboles supere los 34 millones.

Los parámetros numéricos para el modelo evolutivo y los largos de ramas se estiman utilizando métodos numéricos, pero es la exploración del espacio de árboles lo que representa un verdadero problema.

5 Herramientas computacionales

En el presente capítulo se explicarán los pasos que se tomaron para desarrollar las herramientas computaciones para este trabajo de titulación. Estas son:

1. Un programa que simula secuencias de ADN a partir de un árbol con largos de ramas y un modelo evolutivo.

2. Una aplicación web con la que se puede graficar, modificar y descargar las imágenes de las filogenias.

Ambas herramientas computacionales fueron creadas utilizando las facilidades del lenguaje de programación `R`.

5.1 El formato Newick para árboles filogenéticos

Antes de presentar los detalles de las herramientas computacionales, empecemos revisando una forma intuitiva y simple de representar árboles filogenéticos, en comparación a la forma tradicional como un par ordenado compuesto de un conjunto de nodos y otro de aristas, (V, E). Esta forma de representar árboles filogenéticos o filogenias es llamada Formato Newick Estándar.

Aunque no existen publicaciones formales sobre esta representación, es ampliamente utilizada por biólogos y demás profesionales que estudian la evolución.

Para representar un árbol en formato Newick, se agrupa entre paréntesis las etiquetas de los taxones para especificar el patrón de agrupamiento de los nodos del árbol. De esta forma se puede escribir todo el árbol el una sola línea.

Para ilustrar la representación en formato Newick, consideremos el siguiente ejemplo de un árbol escrito en este formato.

$$(A:3,(B:2,C:1)E:3,D:5)F; \tag{5.1}$$

La representación gráfica del árbol en (5.1) se encuentra en la figura (5.1) y como se puede observar, las hojas del árbol están etiquetadas por medio de las letras: A, B, C y D. En el formato Newick, las etiquetas de las hojas pueden tener una o más letras, números y el guión bajo (_), pero no pueden tener espacios en blanco.

Los largos de las ramas se escriben al lado derecho del nodo al que son incidentes, escribiendo primero dos puntos (:) y luego el número que representa el largo de esa rama.

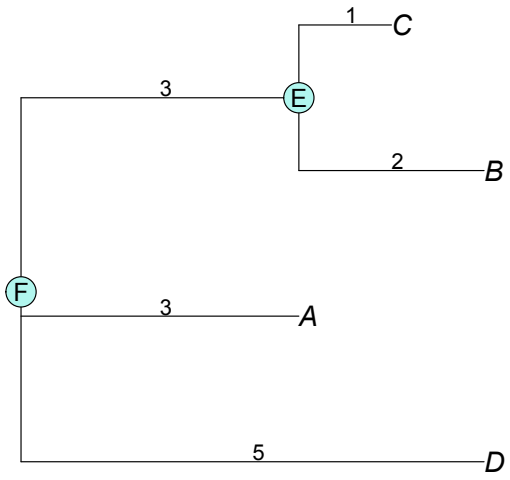


FIGURA 5.1: Representación gráfica del árbol en (5.1)

Un árbol filogenético tiene básicamente dos tipos de nodos: las hojas y los nodos internos. A los nodos hojas se les asigna el nombre de los taxones a los que representan. Por otro lado, los nodos internos pueden, o no, estar etiquetados. En el árbol de la expresión (5.1), los nodos internos están etiquetados por E y F. Sabemos que existe un nodo interno luego de cada paréntesis de cierre pues, el formato Newick consiste en agrupar primero lo nodos que tienen el ancestro más cercano en común.

Por último, todo árbol en este formato termina con un punto y coma (;).

***Observación* 5.1.** Para terminar esta se sección se presenta un par de observaciones.

i) La representación Newick no es única. En efecto, el orden de los descendientes de un nodo afecta la representación gráfica de las filogenias. Sin embargo, esto no es biológicamente importante pues las filogenias son las mismas; por esta razón, para un biólogo las siguientes representaciones:

$$(A,(B,C),D); \quad (A,(C,B),D);$$
$$(D,(C,B),A); \quad (D,A,(B,C)); \quad ((C,B),A,D);$$

corresponden al mismo árbol $(A,(B,C),D)$;

ii) Existen otros formatos para representar árboles como el formato NEXUS estándar para árboles. Sin embargo, este formato está basado en la representación Newick.

También existe el formato `PhyloXML`, el cual usa una representación XML de los árboles mediante clados anidados, es decir, usa etiquetas de tipo

<CLADE> ... </CLADE>

en lugar de paréntesis.

[25]

5.2 Programa para simular secuencias de ADN

Como se indicó al inicio del Capítulo (1), con el programa de simulación de secuencias, se busca ejecutar la tarea "inversa" a la de reconstrucción de árboles filogenéticos.

Recordemos que la Reconstrucción Filogenética consiste en construir una topología τ junto con un conjunto de largos de ramas $\{t_e : e \in \mathbf{E}(\tau)\}$ y parámetros para el modelo evolutivo M a partir de una matriz D que contiene N secuencias de ADN, cada una de longitud s, de tal forma que τ, $\{t_e : e \in \mathbf{E}(\tau)\}$ y M expliquen las relaciones evolutivas entre las secuencias de ADN de D.

Por otro lado, lo que se pretende hacer con el programa de simulación de secuencias de ADN es lo contrario, esto es: *a partir de un árbol filogenético τ con largos de ramas $\{t_e : e \in \mathbf{E}(\tau)\}$ y un modelo evolutivo M, se quiere obtener una matriz D de secuencias de ADN de longitud n.*

Las dimensiones de la matriz D son $(h \times n)$, donde h es el número de hojas del árbol τ.

Antes de describir la idea detrás del programa, recordemos que un modelo evolutivo M está representado por su generador infinitesimal Q y su vector de frecuencias estacionarias π. Como se indica en la sección (2.1) del Capítulo (2), para un árbol con una sola rama con S_0 y S_1 como las secuencias del ancestro y del descendiente respectivamente, la probabilidad condicional de observar cada uno de los nucleótidos en el nodo descendiente (dada la información del ancestro), se calcula mediante la siguiente ecuación matricial:

$$\mathbf{p}_{t_e} = (p_A, p_G, p_C, p_T | \pi, t_e) = \pi e^{Qt_e} \tag{5.2}$$

donde t_e es el largo de la rama que conecta S_0 con S_1 y π es el vector de frecuencias estacionarias de S_0 y Q es el generador infinitesimal del modelo que describe la evolución en la rama que une al ancestro con su descendiente.

Gracias a la propiedad Markoviana, se puede extender la idea anterior a un árbol τ con más de una rama, aplicando sucesivamente la ecuación (5.2) a las ramas de τ hasta llegar a sus hojas.

Entonces ahora ya podemos iniciar la descripción del programa de generación de secuencias y para ilustrar la idea detrás del mismo, consideremos el siguiente ejemplo.

Supongamos que se quiere simular un nucleótido para cada una de las hojas del árbol de la figura (5.2). Además la evolución de todas sus ramas

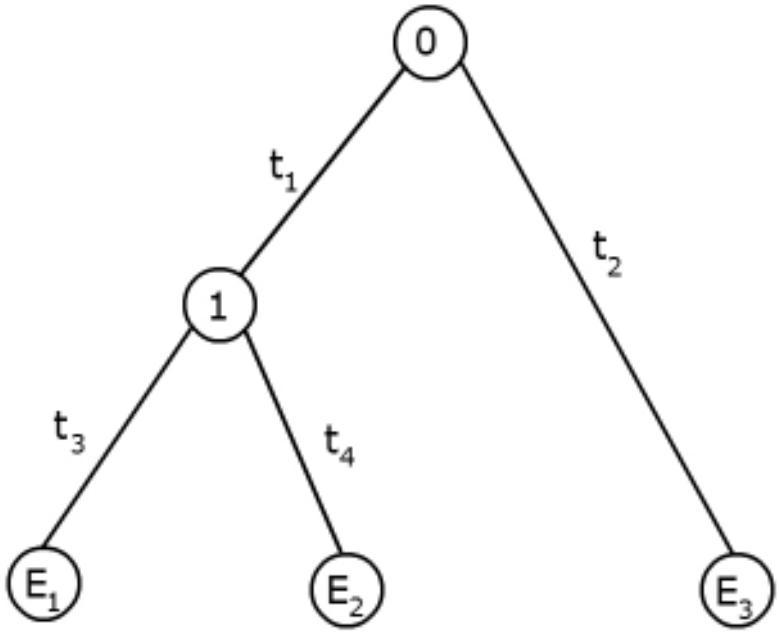


FIGURA 5.2: Filogenia de ejemplo

está explicada por medio de un modelo Kimura de dos parámetros con tasas α y β ($\alpha < \beta$).

Como se explica en la sección (2.4.2) del Capítulo (2), el modelo Kimura–2 tiene el siguiente generador infinitesimal:

$$Q = \begin{pmatrix} -(\alpha+2\beta) & \alpha & \beta & \beta \\ \alpha & -(\alpha+2\beta) & \beta & \beta \\ \beta & \alpha & -(\alpha+2\beta) & \beta \\ \beta & \beta & \alpha & -(\alpha+2\beta) \end{pmatrix} \tag{5.3}$$

y vector de frecuencias estacionarias $\pi = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$.

Empezamos simulando un nucleótido para la raíz 0. Supongamos que tenemos una "bolsa" llena de nucleótidos: A, G, C y T. El tener un vector $\pi = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$ para la raíz, significa que todos los nucleótidos son equiprobables. Esto significa que en nuestra bolsa tenemos la misma proporción de A's, G's, C's y T's.

Entonces sacamos aleatoriamente un nucleótido y supongamos que resulta ser una G (Guanina).

Habiendo ya simulado un nucleótido para la raíz, el siguiente paso es escoger uno de sus descendientes. En el árbol de la figura (5.2) tenemos dos: el nodo 1 y la hoja E_3. Escojamos el nodo 1.

Ahora necesitamos simular un nucleótido para el nodo 1, pero antes debemos calcular su vector de probabilidades condicionales utilizando la ecuación (5.1). Como ya sabemos que en la raíz existió una G, el vector que debemos utilizar en (5.1) es

$$p^0_{bin} = (0, 1, 0, 0).$$

Notar que se está utilizando el orden: A,G,C,T para los nucleótidos.

Entonces el vector de probabilidades condicionales para el nodo 1 se calcula con,

$$\mathbf{p}_1 = (p_A, p_G, p_C, p_T | p^0_{bin}, t_1) = p^0_{bin} e^{Qt_1}$$

$$\mathbf{p}_1 = (a(t_1),\ 1 - a(t_1) - 2b(t_1),\ b(t_1),\ b(t_1)) \tag{5.4}$$

donde $a(t_1)$ y $b(t_1)$ son las componentes de la matriz de Markov para el modelo Kimura–2 como se explica en la sección 2.4.2).

Notar que en la expresión (5.4), la probabilidad de obtener una G es mayor. Podemos imaginar esto como si la proporción de G's en nuestra bolsa de nucleótidos hubiera aumentado en comparación a los demás. En otras palabras, las proporciones de nucleótidos en nuestra bolsa varían conforme avanzamos por las ramas del árbol.

Teniendo ya el vector $\mathbf{p}_1$, podemos sacar aleatoriamente un nucleótido para el nodo 1. Supongamos que en este caso sacamos una A (Adenina).

Los descendientes del nodo 1 son dos hojas: E_1 y E_2. Para simular sus respectivos nucleótidos hacemos un razonamiento similar al anterior.

Escogemos uno de los descendientes del nodo 1, por ejemplo E_1. El largo de la rama que los conecta es t_3 y sabemos que existió una A en el nodo 1, por lo tanto, el vector probabilístico para E_1 se calcula con:

$$\mathbf{p}_{E_1} = (p_A, p_G, p_C, p_T | p^1_{bin}, t_3) = p^1_{bin} e^{Qt_3}$$

donde $p^1_{bin} = (1, 0, 0, 0)$, entonces

$$\mathbf{p}_{E_1} = (1 - a(t_3) - 2b(t_3), a(t_3),\ b(t_3),\ b(t_3)) \tag{5.5}$$

En este caso, la probabilidad de sacar una A de la bolsa de nucleótidos es mayor.

Entonces, sacamos aleatoriamente un nucleótido y supongamos que resulta ser una G. Como estamos en una hoja, necesitamos guardar este último nucleótido simulado en la matriz D.

A continuación, repetimos el procedimiento que hemos estado ejecutando para E_2 que es el otro descendiente del nodo 1. En este caso,

$$\mathbf{p}_{E_2} = (p_A, p_G, p_C, p_T | p^1_{bin}, t_4) = p^1_{bin} e^{Qt_4}$$

$$\mathbf{p}_{E_2} = (1 - a(t_4) - 2b(t_4), a(t_4),\ b(t_4),\ b(t_4)) \tag{5.6}$$

Sacamos aleatoriamente un nucleótido y supongamos que es una A, a la cual también guardamos en la matriz D porque estamos en un nodo hoja.

Hemos terminado con todos los descendientes del nodo 1, entonces volvemos al nodo 0 y revisamos si tiene descendientes sin explorar. En efecto, falta simular un nucleótido para la hoja E_3.

Siguiendo el mismo razonamiento, calculamos el vector de probabilidades condicionales para E_3 con:

$$\mathbf{p}_{E_3} = (p_A, p_G, p_C, p_T | p^0_{bin}, t_2) = p^0_{bin} e^{Qt_2}$$

$$\mathbf{p}_{E_3} = (a(t_2),\ 1 - a(t_2) - 2b(t_2),\ b(t_2),\ b(t_2)) \tag{5.7}$$

Teniendo $\mathbf{p}_{E_3}$ sacamos aleatoriamente un nucleótido y supongamos que es una T (Timina). Como E_3 es una hoja, guardamos el nucleótido T en la matriz D.

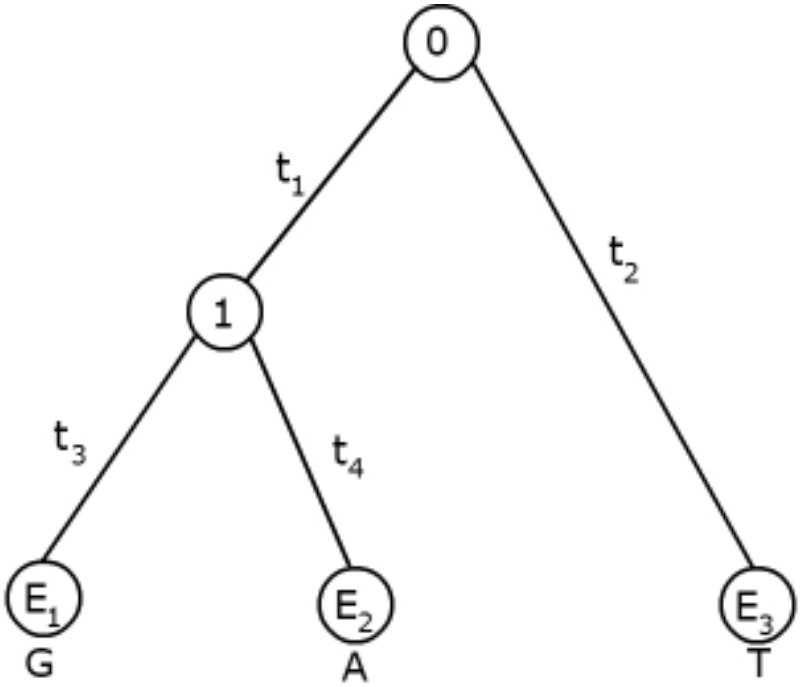


FIGURA 5.3: Nucleótidos simulados

Hemos terminado porque ya no tenemos nodos sin explorar en el árbol de la figura (5.2). En la figura (5.3) se pueden observar los nucleótidos simulados para cada hoja del árbol que usamos como ejemplo.

Por otro lado, la matriz D tiene tres filas y una columna, cuyas entradas corresponden a los nucleótidos simulados anteriormente. Podemos repetir el proceso descrito arriba n veces para simular más columnas para la matriz D, entonces esta tendría la siguiente forma:

	1	2	...	n
E_1	G	T	...	G
E_2	A	T	...	A
E_3	T	C	...	C

5.3 Implementación en R

En la sección anterior se explicó la idea detrás del programa de simulación de secuencias de ADN, el cual se implementó en lenguaje `R`, utilizando funciones de las librerías `ape` y `expm`.

Se puede revisar el código de este programa en el Anexo A.

La librería `ape`, desarrollada por Paradis, Claude y Strimmer [16], contiene varias funciones que permiten manipular filogenias y secuencias de ADN. Una de estas es la función `read.tree()`, la cual toma como argumento un árbol en formato Newick y lo transforma en un objeto de tipo `phylo`. Los objetos de tipo `phylo` son básicamente listas que contienen los siguientes elementos:

- `edge`: Es una matriz de pares ordenados que representan cada una de las ramas del árbol; los nodos internos y hojas están simbolizados por números de la siguiente manera: en un árbol de N hojas, estas se enumeran de 1 a N . Los nodos internos de este árbol están enumerados a partir de $N + 1$.

  En cada fila de la matriz `edge`, los ancestros están en la primera posición del par ordenado.

- `edge.length`: Es un vector que contiene los largos de las ramas en `edge`.

- `tip.label`: Es un vector de tipo `character` que contiene los nombres de las hojas del árbol. El orden de estos nombres corresponde al número con el que se encuentran enumeradas las hojas en la matriz `edge`.

- `Nnode`: Es el número de nodos internos.

- `node.label` Es un vector de tipo `character` para guardar los nombres de los nodos internos. (opcional)

- `root.edge`: Es un valor de tipo numérico que indica el largo de la rama en la raíz si existe. (opcional)

De los elementos anteriores, los que nos interesan son: `tip.label`, `edge.length`, `Nnode` y por supuesto la matriz `edge`.

La segunda librería necesaria para implementar este programa es `expm`, desarrollada por Goulet et al. [6], la cual contiene funciones que permiten efectuar operaciones como el exponencial o el logaritmo de una matriz.

Específicamente, se utilizó la función `expm()` de esta librería para calcular los exponenciales de la ecuación (5.1). Esta función toma como argumento una matriz cuadrada `x` y calcula su exponencial mediante uno de los siguientes métodos: "Higham08.b", Higham08", "AlMohy-Hi09","Ward77", "PadeRBS", "Pade", "Taylor", PadeO", "TaylorO", "R_Eigen", "R_Pade", "R_Ward77" o "hybrid_Eigen_Ward".

El método por defecto es "Higham08.b" y es la opción que se utilizó para implementar el programa de simulación de secuencias.

Teniendo las herramientas en mano, podemos pasar a la implementación del programa de simulación. Como se detalla en el Anexo A, este se compone de tres funciones:

1. Función `sim_nuc(p_0)`.-

   Esta función recibe un vector probabilístico `p_0` y devuelve un vector binario que representa uno de los cuatro posibles nucleótidos.

   La función `sim_nuc()` puede interpretarse como la "bolsa de nucleótidos" de la cual se sacan aleatoriamente los nucleótidos, como se explicó en la sección anterior. Esta funciona de la siguiente manera.

   Primero se obtiene un número aleatorio `aleat`, con distribución uniforme entre $[0, 1]$.

   Luego,

   - Si $0 \leq$ `aleat` $<$ `p_0[1]`, retornar $p_{bin} = (1, 0, 0, 0)$.
   - Si `p_0[1]` $\leq$ `aleat` $< \sum_{i=1}^{2}$ `p_0[i]`, retornar $p_{bin} = (0, 1, 0, 0)$.
   - Si $\sum_{i=1}^{2}$ `p_0[i]` $\leq$ `aleat` $< \sum_{i=1}^{3}$ `p_0[i]`, retornar $p_{bin} = (0, 0, 1, 0)$.
   - Si $\sum_{i=1}^{3}$ `p_0[i]` $\leq$ `aleat` ≤ 1, retornar $p_{bin} = (0, 0, 0, 1)$.

   `p_0[i]` es la i–ésima posición del vector `p_0`, en pocas palabras se devuelve el vector binario de acuerdo a los acumulados del vector probabilístico p_0.

2. Función `codificar(p_bin)`.-

   Esta función simplemente transforma un vector binario `p_bin` en una letra, de la siguiente manera:

   - Si `p_bin` $= (1, 0, 0, 0)$, retornar "A".
   - Si `p_bin` $= (0, 1, 0, 0)$, retornar "G".
   - Si `p_bin` $= (0, 0, 1, 0)$, retornar "C".
   - Si `p_bin` $= (0, 0, 0, 1)$, retornar "G".

3. Función `sim_evolucion(tr,Q,pi,n_sitios)`.-

   Por último, la función `sim_evolucion()` simula el número de columnas especificado en la variable `n_sitios` para la matriz D con el procedimiento descrito en la sección anterior (5.2) para un árbol `tr`, en el cual la evolución de sus ramas es acorde a un modelo evolutivo con generador infinitesimal `Q` y vector de frecuencias estacionarias `pi`.

   La función empieza determinando el número de hojas del árbol (`n_taxa`), el número de ramas del árbol (`n_ramas`) y el número de nodos internos (`n_int`). Luego de definen dos matrices:

   - `nuc_taxa`: de dimensiones `N` $\times$ `n_sitios`, donde `N` es el número de hojas del árbol `tr`. En esta matriz se guardarán los nucleótidos simulados para las hojas.
   - `nodos_int`: es una matriz cuyas dimensiones son `n_int` $\times$ 4, donde `n_int` es el número de nodos internos del árbol `tr`. En esta matriz se almacenarán los vectores binarios simulados para los nodos internos del árbol.

Luego se tienen dos lazos `for` anidados, el lazo exterior avanza en `j`, de 1 a `n_sitios`. Antes de pasar al lazo interno, se simula un nucleótido para la raíz utilizando la función `sim_nuc()` y el vector `pi`. Este vector, al que se ha denotado `p_0`, es guardado en la primera fila de la matriz `nodos_int`.

En seguida, se tiene el lazo `for` interno, el cual avanza en i de 1 a `n_ramas`. Notar que el número de ramas de `tr` es la longitud del vector `edge.lenght` y este vector necesariamente debe tener los largos de todas las ramas del árbol `tr`.

El lazo `for` interno recorre la matriz de ramas `edge` del árbol `tr`. Empezamos ubicando los nodos ancestro y descendiente, a los cuales se ha denotado `n_ini` y `n_fin` respectivamente.

El siguiente paso es calcular vector de probabilidades condicionales para el descendiente con la ecuación (5.1). A este vector se lo ha denotado `p_1`; enseguida se simula un nucleótido para el descendiente utilizando la función `sim_nuc()` y el vector `p_1`. Finalmente, determinamos si el descendiente es un nodo interno o una hoja. Si es nodo interno, simplemente guardamos el vector binario obtenido en el paso anterior en la matriz `nodos_int` y avanzamos a la siguiente rama. Caso contrario, transformamos el vector binario (obtenido en el paso anterior) al nucleótido que corresponda con la función `codificar()` y guardamos esa letra en la matriz `nuc_taxa`.

Al terminar, el programa retorna la matriz `nuc_taxa`.

¿Qué hacer luego de ejecutar `sim_evolucion()`?

La matriz que se obtiene al aplicar la función `sim_evolucion()` puede usarse para escribir un archivo en el formato que el usuario desee. Por ejemplo, puede usarse la función `write.nexus.data()` de la librería `ape` para escribir las secuencias de ADN en formato `nexus`.

En este caso se ha utilizado la función `write.phyDat` de la librería `phangorn` para escribir las secuencias simuladas en archivos de formato `phylip`, los cuales van a usarse para hacer pruebas estadísticas y reconstrucciones con el programa `IQTREE`, como se detallará en el siguiente capítulo.

5.4 Aplicación web para graficar árboles filogenéticos

Para finalizar este capítulo, se presenta la segunda herramienta computacional desarrollada como un aporte extra de este trabajo de titulación.

Entre las funciones que se encuentran implementadas en la librería `ape`, está `plot()`. Esta función permite representar gráficamente árboles en formato Newick o NEXUS.

Sin embargo, es necesario conocer al menos los conceptos básicos de programación en `R` para poder utilizar las herramientas desarrolladas por Paradis, Claude y Strimmer [16]. De ahí es donde surge la idea de crear una aplicación web, la cual permita hacer uso de las herramientas de la librería `ape` de una forma más interactiva y amigable con el usuario. Entonces nació *Yura*, una aplicación web creada utilizando las facilidades de la librería `Shiny`, para dibujar, manipular y descargar las imágenes de las filogenias en diferentes formatos.

5.4.1 Sobre la librería Shiny y el desarrollo de Yura

La librería `Shiny`, desarrollada por Chang et al. [4], es un paquete que facilita la creación de páginas web interactivas desde `R`.

Las aplicaciones Shiny se componen de una función para definir la interfaz de usuario y una función de servidor, en inglés se conoce como un *UI/Server pair*.

Como su nombre lo indica, en la parte de la interfaz de usuario se implementan todas las herramientas que le ayudarán al usuario a interactuar con la aplicación web. Por ejemplo, en esta parte se escribe el título de la página, los paneles de navegación, los menús de navegación o los objetos de entrada y salida.

En la parte del `server`, se encuentran las librerías y las funciones de R que van a ejecutar las instrucciones del usuario.

Interfaz de usuario de Yura

FIGURA 5.4: Pestañas de navegación

La aplicación Yura tiene tres pestañas de navegación: "Ayuda/Subir archivo", "Graficar, Modificar árbol" y "Detalles del árbol".

La pestaña de "Ayuda/Subir archivo" se encuentra la imagen (5.4). En esta pestaña se encuentra la información sobre la página y la opción de subir un archivo que contenga un árbol en formato Newick o NEXUS.

FIGURA 5.5: Pestañas para graficar y modificar el árbol

La segunda pestaña consta de un menú de navegación con dos opciones: "Gráfico del árbol" y "Modificar". En la figura (5.5) se puede apreciar el gráfico de un árbol como se presenta en la opción "Gráfico del árbol" de la segunda pestaña. Como se puede observar en esta imagen, en la parte lateral izquierda se encuentra un panel de navegación en el que se encuentran todas las posibles opciones para modificar el gráfico del árbol, por ejemplo: el formato para descargar la imagen, el tamaño de la letra para las etiquetas de las hojas, el tipo y la dirección del gráfico de árbol, etcétera.

Por otro lado, en la opción "Modificar", en esta pestaña se puede eliminar hojas del árbol para modificar la filogenia.

Por último, en la pestaña de "Detalles del árbol", se puede ver información acerca del árbol. En esta pestaña se puede ver la matriz de ramas del árbol, el número de nodos internos, los nombres de las hojas o taxones, los largos de las ramas y las etiquetas de los nodos internos.

Función de servidor de Yura

Todo lo descrito en la sección anterior está definido en la parte de la interfaz de usuario de la aplicación.

En el `server` de la aplicación Yura se llama a las librerías necesarias para ejecutar las funciones que dibujan y modifican el árbol con las opciones que el usuario elige.

El código en R de la aplicación Yura se encuentra en el Anexo B.

La versión más reciente de la aplicación web Yura, se encuentra disponible en el siguiente enlace:

https://phylotreeplotapp.shinyapps.io/yura_app/

6 Resultados computacionales

En el presente capítulo se describen los resultados de experimentos computacionales, realizados con el objetivo de medir el desempeño de algoritmos de reconstrucción de filogenias basados en el paradigma de la Máxima Verosimilitud y empleando tres modelos evolutivos distintos.

Para realizar los experimentos, se simularon un total de 45000 matrices de secuencias de ADN empleando la función `sim_evolucion()`, descrita en la sección (5.3), de la siguiente forma:

- Se consideraron cinco árboles diferentes de 5, 12, 20, 30 y 50 taxones.
- Los modelos evolutivos utilizados fueron Kimura–2 (K2P), Jukes–Cantor (JC) y el modelo F81, el cual se explicará en la siguiente sección.
- Se establecieron longitudes de 500, 1000 y 2000 sitios para las secuencias de ADN.
- Se simularon 1000 matrices para cada una de las combinaciones árbol + modelo + número de sitios.

Las instancias para el modelo F81 fueron simuladas utilizando el vector de frecuencias estacionarias

$$(\pi_A, \pi_G, \pi_C, \pi_T) = (0.3241, 0.1055, 0.304, 0.2663) \tag{6.1}$$

y escalando la matriz generadora Q por el factor

$$\beta = \frac{1}{1 - (\pi_A^2 + \pi_G^2 + \pi_C^2 + \pi_T^2)}$$

para que los largos de ramas representen el número esperado de cambios por sitio en las secuencias de ADN.

La representación de la ecuación (2.15) es la que se utilizó para hacer los análisis de las instancias con modelos K2P utilizando un $\kappa = 5$ para las instancias de 5, 20, 30 y 50 taxones y un $\kappa = 4.572$ para la instancia de 12 taxones.

Tanto los valores del vector π para el modelo F81 como los valores de κ para el modelo K2P, fueron obtenidos al realizar la reconstrucción de secuencias de ADN reales de un grupo de primates. De este análisis se obtuvo las frecuencias en (6.1) y el valor de $\kappa = 4.572$.

Luego de tener el grupo de 1000 matrices para cada una de las combinaciones árbol + modelo + número de sitios, utilizando el programa `IQ-TREE v. 1.6.2` se corrieron análisis de selección de modelos y reconstrucciones filogenéticas para cada una de las 45000 matrices.

Notar que la cantidad de taxones representa las filas de las matrices y la longitud de las secuencias de ADN es el número de columnas de las matrices.

A cada una de las combinaciones *árbol + modelo + número de sitios* se le denominará *instancia*. En total tenemos 45 instancias y para cada instancia tenemos 1000 simulaciones.

6.1 Criterios de selección de modelos y distancias entre árboles

6.1.1 Criterios de selección de modelos

Un primer e importante paso en los análisis de reconstrucción filogenética es la selección del modelo evolutivo (o modelos en el caso de tener particiones en los datos) que se emplearan para hacer la reconstrucción.

El programa `IQ-TREE` tiene implementado un algoritmo de selección de modelos, que utiliza los criterios de información de Akaike, Akaike corregido y Bayesiano para seleccionar el mejor modelo que ajuste a un determinado conjunto de datos.

Este algoritmo de selección modelo fue aplicado a cada una de las instancias simuladas para determinar qué modelo era escogido por lo diferentes criterios.

El problema de selección de modelos puede plantearse de la siguiente forma: se tiene un conjunto de datos D y R posibles modelos con diferente número de parámetros para ajustar estos datos. Denominaremos a estos modelos $M_1, M_2, \dots M_R$ y denotaremos $p_1, p_2, \dots p_R$, la cantidad de parámetros que tiene cada modelo.

Cada modelo M_i, con $i = 1, \dots R$, ajusta en mayor o menor medida a los datos D. A esta medida de ajuste la denotaremos ℓ_i con $i = 1, 2, \dots, R$.

Entonces queremos encontrar el modelo que ajuste mejor a los datos y contenga la menor cantidad posible de parámetros. Una forma de resolver este problema es utilizando el criterio de Información de Akaike.

Su definición general es la siguiente.

***Definición* 6.1.** *Dados un conjunto de datos D y un conjunto de posibles modelos $M_1, M_2, \dots M_R$ cada uno con $p_1, p_2, \dots p_R$ parámetros respectivamente, se define el criterio de información de Akaike para el modelo M_i por,*

$$AIC_i = -2\ell_i + 2p_i \tag{6.2}$$

El modelo seleccionado será el que genere el menor valor del criterio, en comparación a los demás modelos.

En el caso de árboles filogenéticos, además del número de parámetros del modelo evolutivo (tasa instantáneas de cambio y frecuencias estacionarias), se debe añadir la cantidad total de ramas del árbol para obtener el total de parámetros y la medida de ajuste ℓ_i la brinda el valor de la función de verosimilitud para el alineamiento D dados un árbol filogenético, un conjunto de largos de ramas y los parámetros para el modelo evolutivo.

Cuando la cantidad de parámetros p_i es relativamente grande en comparación al tamaño de la muestra n, se aconseja sumar un término de corrección al AIC, el dual está dado por $p_i(p_i+1)/(n-p_i-1)$.

***Definición* 6.2.** *El criterio de información de Akaike corregido (AICc), para un modelo con p_i parámetros y un tamaño de muestra n, se define por*

$$AICc_i = AIC_i + \frac{2p_i(p_i+1)}{n-p_i-1}$$

$$AICc_i = -2\ell_i + 2p_i + \frac{2p_i(p_i+1)}{n-p_i-1} \tag{6.3}$$

Por otro lado, el criterio de información Bayesiano (BIC), también conocido como criterio de información de Schwarz, está basado en un enfoque bayesiano de comparación de modelos. El BIC, básicamente compara las probabilidades posteriores de los datos dado un modelo y escoge el que genere la mayor probabilidad posterior para los datos.

El Teorema de Bayes indica que la probabilidad posterior del modelo M_i es:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{\sum_{j=1}^{R} P(D|M_j)P(M_j)} \tag{6.4}$$

donde $P(M_i)$ para $i = 1, 2, \ldots, R$ es conocida como la probabilidad a priori del modelo M_i.

Como indican Konishi y Kitagawa [11], si suponemos que todas las $P(M_i)$ son iguales, es decir, que los modelos son equiprobables, se puede definir al BIC de la siguiente forma.

***Definición* 6.3.** *Cuando el tamaño de muestra n es grande en comparación al número de parámetros p_i del modelo M_i, el BIC puede ser aproximado por la siguiente expresión:*

$$BIC_i \approx -2\ell_i + p_i \log(n) \tag{6.5}$$

donde $\ell_i = \log(P(D|M_i))$, es la medida en la que el modelo M_i ajusta los datos D.

Observación 6.1.

- En reconstrucción filogenética, el tamaño de la muestra n corresponde a la longitud de las secuencias de ADN en el alineamiento, en otras palabras, es la cantidad de columnas de la matriz de datos.

- Los criterios anteriormente descritos buscan medir, tanto la falta de ajuste, como la complejidad del modelo para los datos.

  En el caso del AIC, esta "complejidad" del modelo se mide con el número de parámetros p_i y en el caso del BIC, esta se mide con la expresión $p_i \log(n)$.

  La falta de ajuste es medida por medio del logaritmo de la verosimilitud y a esta se le suma una "penalidad" por la complejidad del modelo.

6.1.2 Distancias entre árboles filogenéticos

Para poder medir las diferencias entre los árboles reconstruidos del árbol original con el que se hicieron las respectivas simulaciones, se hizo uso de dos distancias entre árboles: la *distancia de Robinson y Foulds* y la distancia *Branch Score* de Kuhner y Felsenstein.

La distancia de Robinson y Foulds está definida para árboles filogenéticos sin largos de ramas, es decir, solo para topologías. Dadas dos topologías τ_1 y τ_2, esta distancia básicamente cuenta la cantidad de ramas en las que se diferencian.

Tanto la distancia RF como la distancia BS (Branch Score) se basan en la propiedad de los árboles en la que cada una de sus ramas induce una partición en el conjunto de las hojas. Sea S el conjunto de nodos hoja de un árbol τ, si eliminamos una rama $e \in \mathbb{E}(\tau)$ del árbol τ, se genera una partición del conjunto S en dos conjuntos $\{R_1, R_2\}$.

Denotamos $\Sigma(\tau)$ a la colección de todas las particiones inducidas por el conjunto de ramas $\mathbb{E}(\tau)$, entonces la distancia RF puede definirse de la siguiente forma.

***Definición* 6.4** (Distancia RF)**.** *Sean τ_1 y τ_2 dos árboles filogenéticos sin largos de ramas con conjuntos de particiones inducidas por las ramas $\Sigma(\tau_1)$ y $\Sigma(\tau_2)$, respectivamente. Se define la distancia RF entre τ_1 y τ_2 por,*

$$d_{RF}(\tau_1, \tau_2) = |\Sigma(\tau_1) \triangle \Sigma(\tau_2)| \tag{6.6}$$

donde $A \triangle B$ representa la diferencia simétrica entre los conjuntos A y B, y el símbolo $|A|$ representa la cardinalidad del conjunto A.

Por tanto, la distancia RF se define como la cardinalidad del conjunto $\Sigma(\tau_1) \triangle \Sigma(\tau_2)$.

Por otro lado, la distancia Branch Score toma en cuenta el largo de las ramas y fue creada por Kuhner y Felsenstein [12], estos la definen como la suma de los cuadrados de las diferencias entre cada uno de los largos de ramas entre un arbol τ y otro τ', donde las ramas que aparecen en un árbol pero no en otro, fueron marcadas como si fueran comparadas con una rama de largo cero.

Para definir formalmente la distancia Branch Score, consideremos primero la siguiente definición.

***Definición* 6.5.** *Dado un árbol métrico τ con largos de ramas $\{t_e : e \in \mathbb{E}(\tau)\}$ y con h hojas, se define el vector $B_\tau := (b_1, b_2, \ldots, b_N)$, donde $N = 2^{h-1} - 1$ es el número de todas las posibles particiones del conjunto de hojas. Suponemos que estas particiones han sido ordenadas según algún criterio fijo.*

Para la partición i–ésima, el valor de b_i está dado por:

$$b_i = \begin{cases} t_e, & \text{si } \tau \text{ tiene una arista } e \text{ asociada a la partición } i \\ 0, & \text{caso contrario} \end{cases}$$

Luego, la distancia Branch Score puede definirse de la siguiente manera.

***Definición* 6.6** (Distancia BS o Distancia KF)**.** *Para dos árboles métricos τ y τ' que tienen la misma cantidad de hojas y cuyos vectores inducidos por las particiones de las ramas son $B_\tau = (b_1, b_2, \ldots, b_N)$ y $B_{\tau'} = (b'_1, b'_2, \ldots, b'_N)$ se define la distancia Branch Score o distancia de Kuhner y Felsenstein, como la distancia euclidiana entre esos dos vectores. Es decir:*

$$d_{BS}(\tau, \tau') = ||B_\tau - B_{\tau'}||_{L^2} = \left(\sum_{i=1}^{N} (b_i - b'_i)^2 \right)^{1/2} \tag{6.7}$$

Observación 6.2.

1. Notar que la distancia RF es la igual a la distancia KF pero para árboles con largos de ramas iguales a 1.

2. Si τ y τ' son dos árboles filogenéticos con topologías idénticas y los mismos largos en cada rama, entonces $B_\tau = B_{\tau'}$ y por tanto la distancia $d_{BS}(\tau, \tau') = 0$.

3. Notar, sin embargo, que la distancia d_{BS} puede ser cero para árboles filogenéticos con topologías totalmente distintas pero con largos de ramas iguales a cero. En efecto, si τ y τ' son dos árboles filogenéticos con topologías distintas pero con largos de ramas iguales a cero, entonces $B_\tau = B_{\tau'} = \mathbf{0}$, donde $\mathbf{0}$ es un vector compuesto por ceros.

4. Teniendo en cuenta la observación del punto (3), el tener valores relativamente pequeños de d_{BS} no permite concluir inmediatamente que dos árboles filogenéticos son muy parecidos, porque puede ser que sus largos de ramas sean muy cercanas a cero pero sus topologías sean distintas.

   Por esta razón, en los análisis de la sección (6.3), se compararán las dos distancias d_{BS} y d_{RF} para no caer en el sesgo de la distancia Branch Score.

6.2 Sobre el programa `IQ-TREE v. 1.6.2`

Para hacer los análisis a las 45000 matrices simuladas se utilizó el programa `IQ-TREE v. 1.6.2`. Este tiene implementada entre muchas otras cosas, el algoritmo de reconstrucción de árboles filogenéticos por Máxima Verosimilitud desarrollado por Nguyen et al. [15] y el algoritmo de selección de modelos desarrollado por Kalyaanamoorthy et al. [8].

Los análisis que se hicieron con este programa fueron los siguientes.

1. Primero se corrió el algoritmo de selección de modelos con el comando `-m TESTONLY` para obtener el mejor modelo para los datos simulados según los criterios AIC, AICc y BIC.

2. Luego se corrieron análisis de reconstrucción por Máxima Verosimilitud con los comandos:

- `-m JC+FQ` para las instancias simuladas con el modelo de Jukes–Cantor.
- `-m K2P+FQ` para las instancias simuladas con el modelo de Kimura–2.
- `-m F81+FO` para las instancias simuladas con el modelo de F81.

6.3 Resultados computacionales

Con el objetivo de facilitar la lectura de esta sección, se ha organizado el contenido de la siguiente manera: primero se presentará los gráficos de los árboles utilizados para hacer las simulaciones, luego se mostrará los gráficos de las distribuciones las estimaciones de los parámetros para los diferentes modelos (κ para K2P y $\pi_A, \pi_G, \pi_C, \pi_T$ para F81), seguidos de las distancias entre árboles y por último se presentará las tablas que contienen la información obtenida con el algoritmo de selección de modelos.

Todos los árboles en formato Newick se encuentran en el Anexo C.

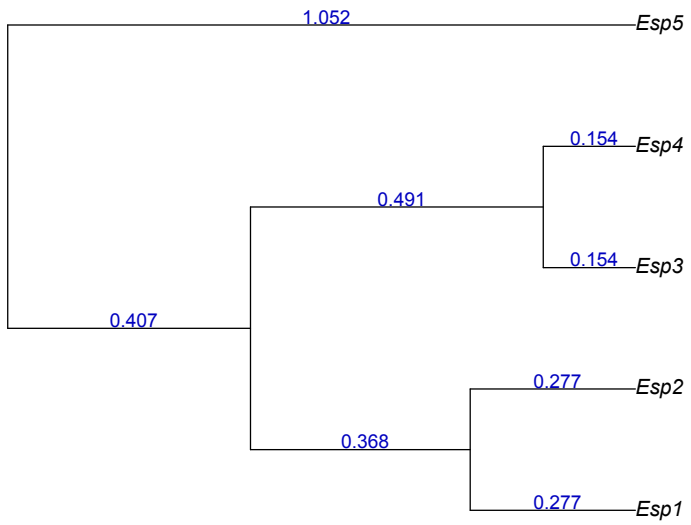


FIGURA 6.1: Árbol de 5 taxones

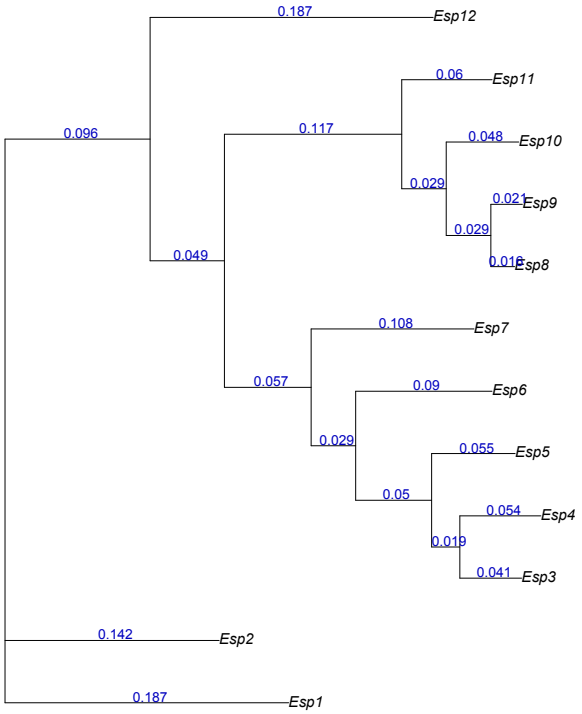


FIGURA 6.2: Árbol de 12 taxones

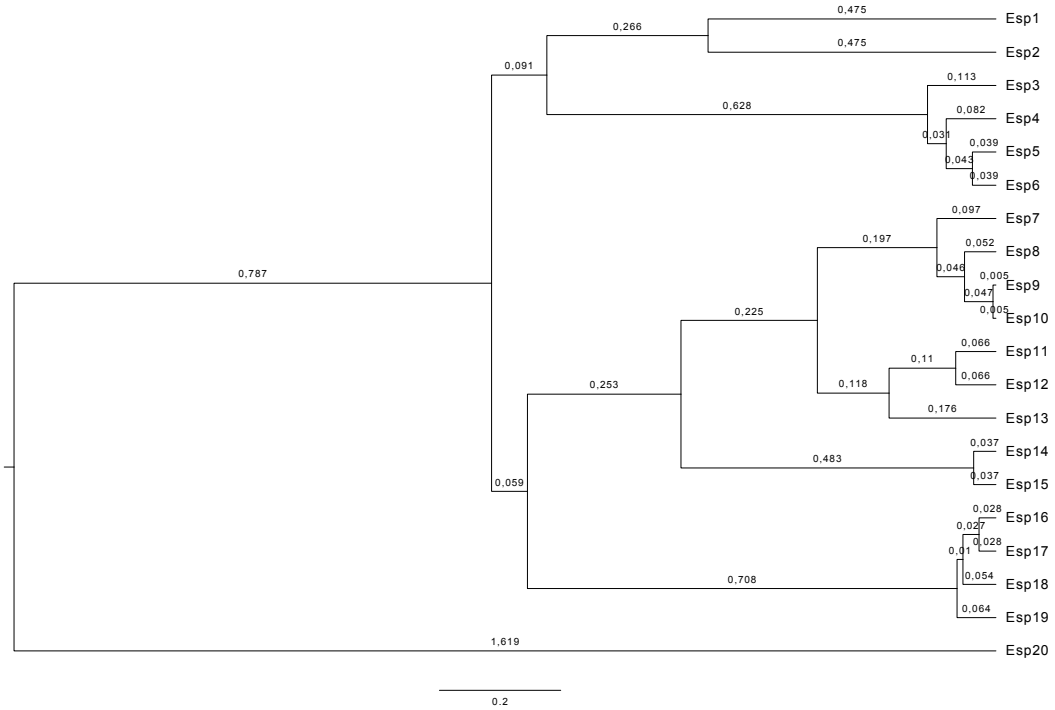


FIGURA 6.3: Árbol de 20 taxones

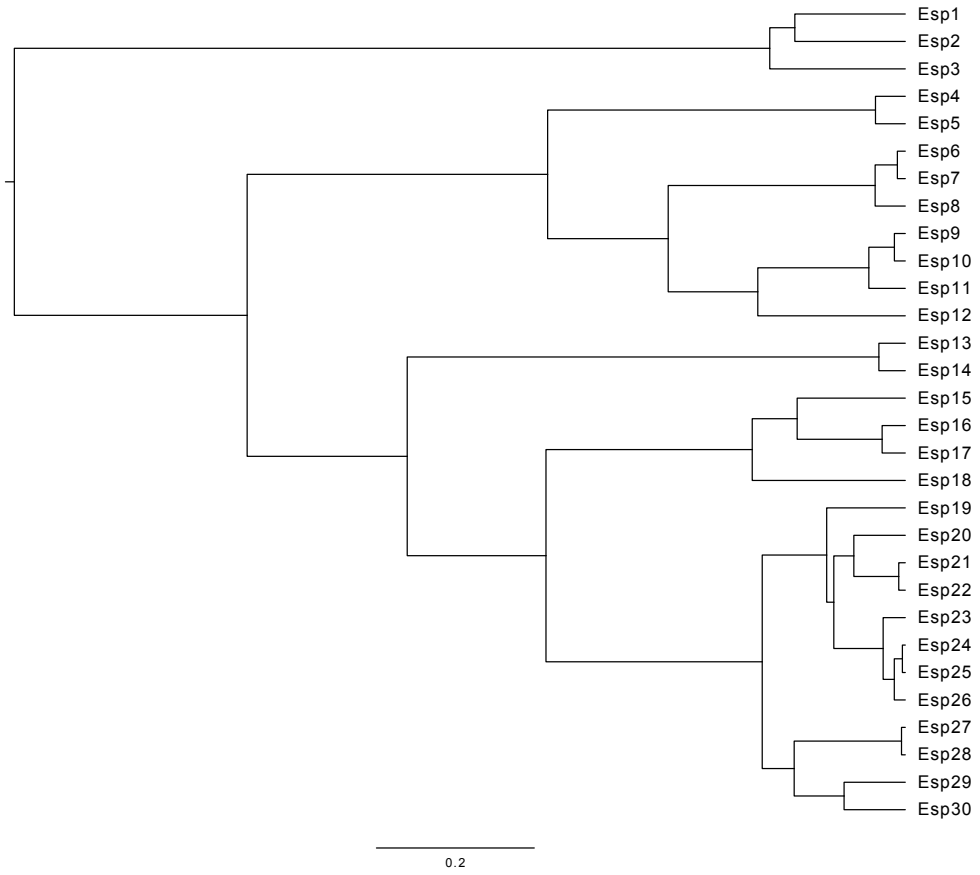


FIGURA 6.4: Árbol de 30 taxones

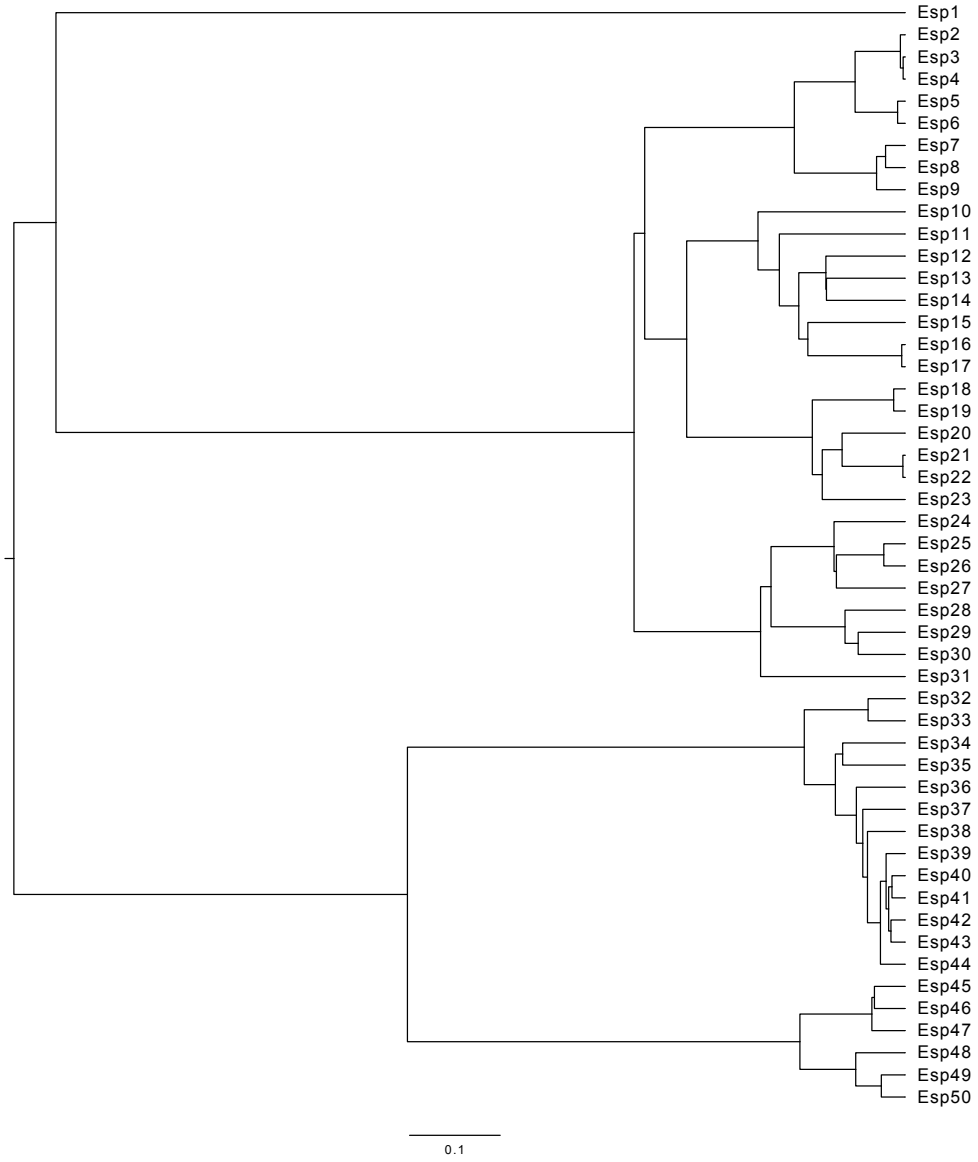


FIGURA 6.5: Árbol de 50 taxones

6.3.1 Resultados de las reconstrucciones: estimaciones de kappa

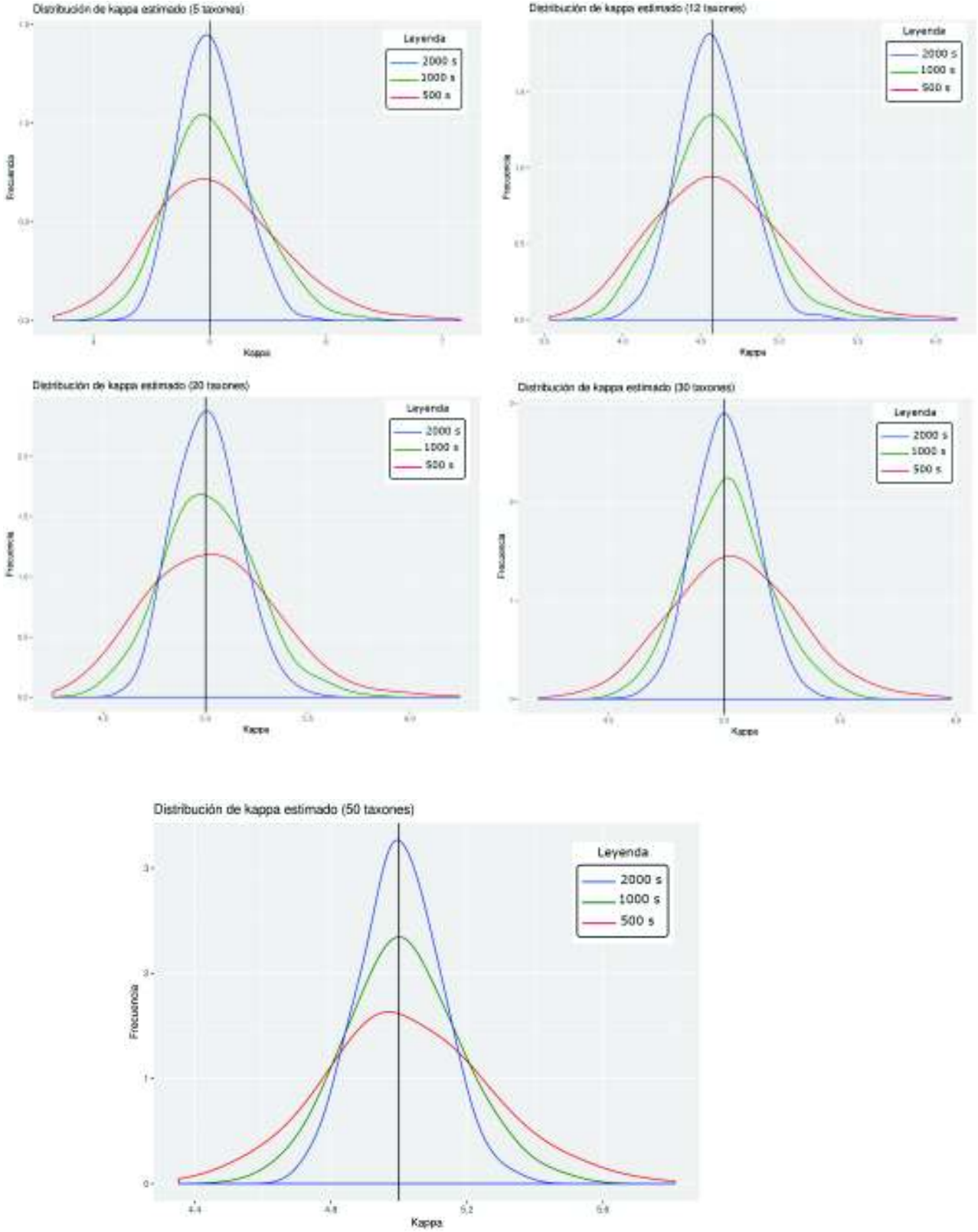


FIGURA 6.6: Estimaciones de kappa para 5, 12, 20, 30 y 50 taxones

En la figura (6.6) se puede apreciar los gráficos de las distribuciones de las estimaciones de máxima verosimilitud del parámetro κ para las instancias que fueron simuladas con el modelo K2P para 5, 12, 20, 30 y 50 taxones.

Los resultados muestran el comportamiento esperado de los estimadores de máxima verosimilitud pues, las curvas de las distribuciones se van afinando conforme aumenta la longitud de las secuencias (tamaño de la muestra).

La línea en negro es el verdadero valor de κ, es decir, es el valor con el que se simularon los datos. Este valor fue 4.572 para las instancias de 12 taxones y 5 para las demás.

Se observa que la distribución de las estimaciones de kappa para secuencias de 500 sitios es plana, pero la varianza va disminuyendo para 1000 sitios; para 2000 sitios la varianza disminuye y valores se van agrupando alrededor del verdadero valor del parámetro como se esperaba.

6.3.2 Estimaciones de los parámetros del modelo F81

En la figura (6.7) se encuentran los gráficos de las distribuciones de las estimaciones de la frecuencia π_A del modelo F81. Nuevamente se han agrupado los resultados para 5, 12, 20, 30 y 50 taxones, y la línea vertical en negro es $\pi_A = 0.3241$, que es el valor con el que se hizo las simulaciones.

Los gráficos muestran cierto sesgo, sobre todo para las estimaciones de las instancias de 500 y 1000 sitios.

Las estimaciones de π_G se encuentran en el gráfico (6.8). El valor que se utilizó para las simulaciones fue $\pi_G = 0.1055$.

Como se puede observar, el comportamiento de estos estimadores es el esperado, pues la varianza disminuye y las estimaciones se van agrupando alrededor del verdadero valor del parámetro conforme se aumenta el tamaño de las secuencias.

Las estimaciones para el parámetro π_C se encuentran en la figura (6.9). El valor que se utilizó para las simulaciones fue $\pi_C = 0.304$ y es por donde pasan las líneas de color negro.

Al parecer, las estimaciones de π_C tienen cierto sesgo, sobre todo para las instancias simuladas con 500 y 1000 sitios.

Finalmente, las estimaciones para π_T de las instancias simuladas con el modelo F81 se encuentran en la figura (6.10). El verdadero valor del parámetro es $\pi_T = 0.2663$.

En general, el comportamiento de estas estimaciones es el esperado, porque su distribución es plana para secuencias de 500 sitios, pero se va afinando conforme se aumenta la longitud de las secuencias.

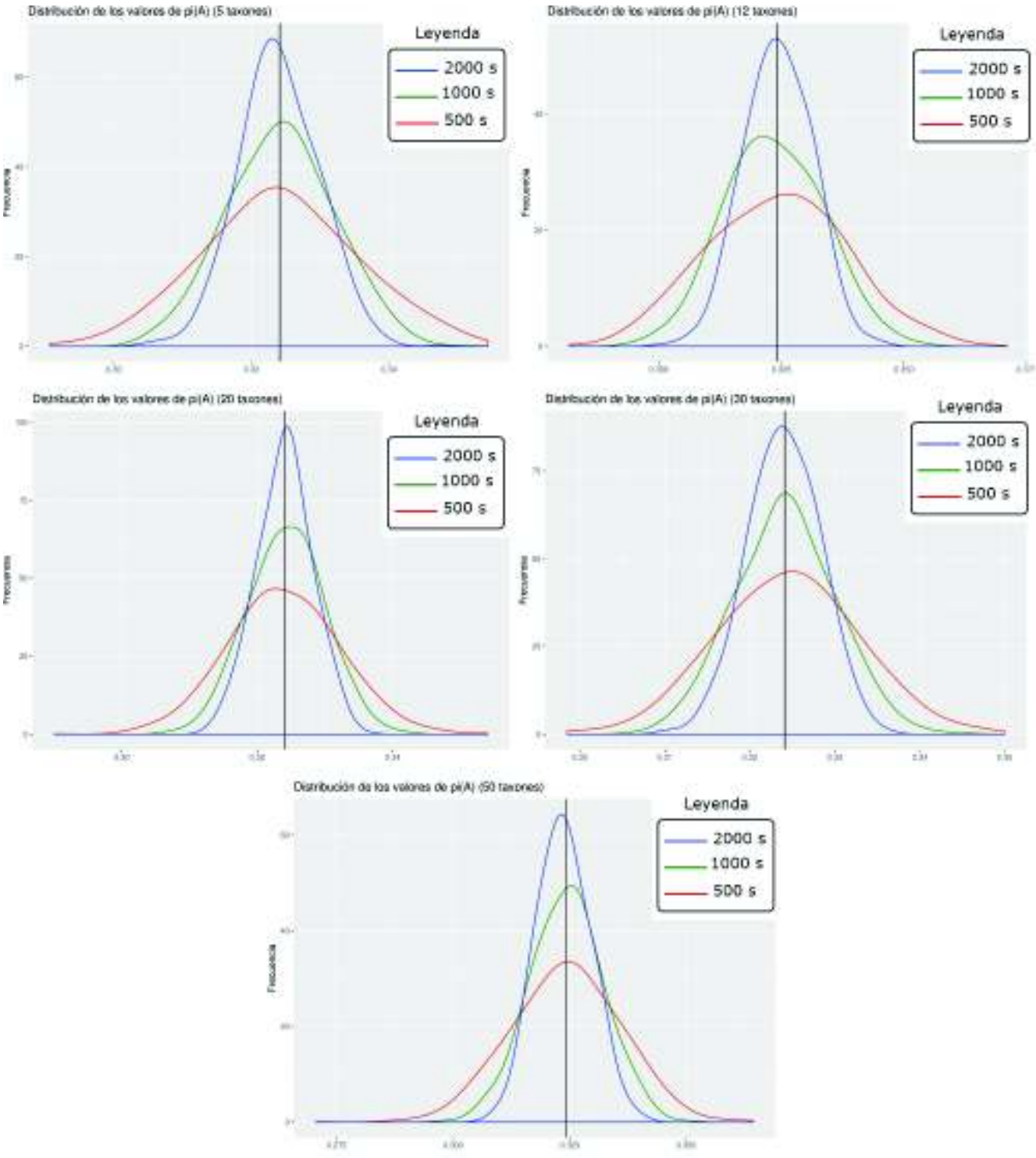


FIGURA 6.7: Estimaciones de π_A para 5, 12, 20, 30 y 50 taxones

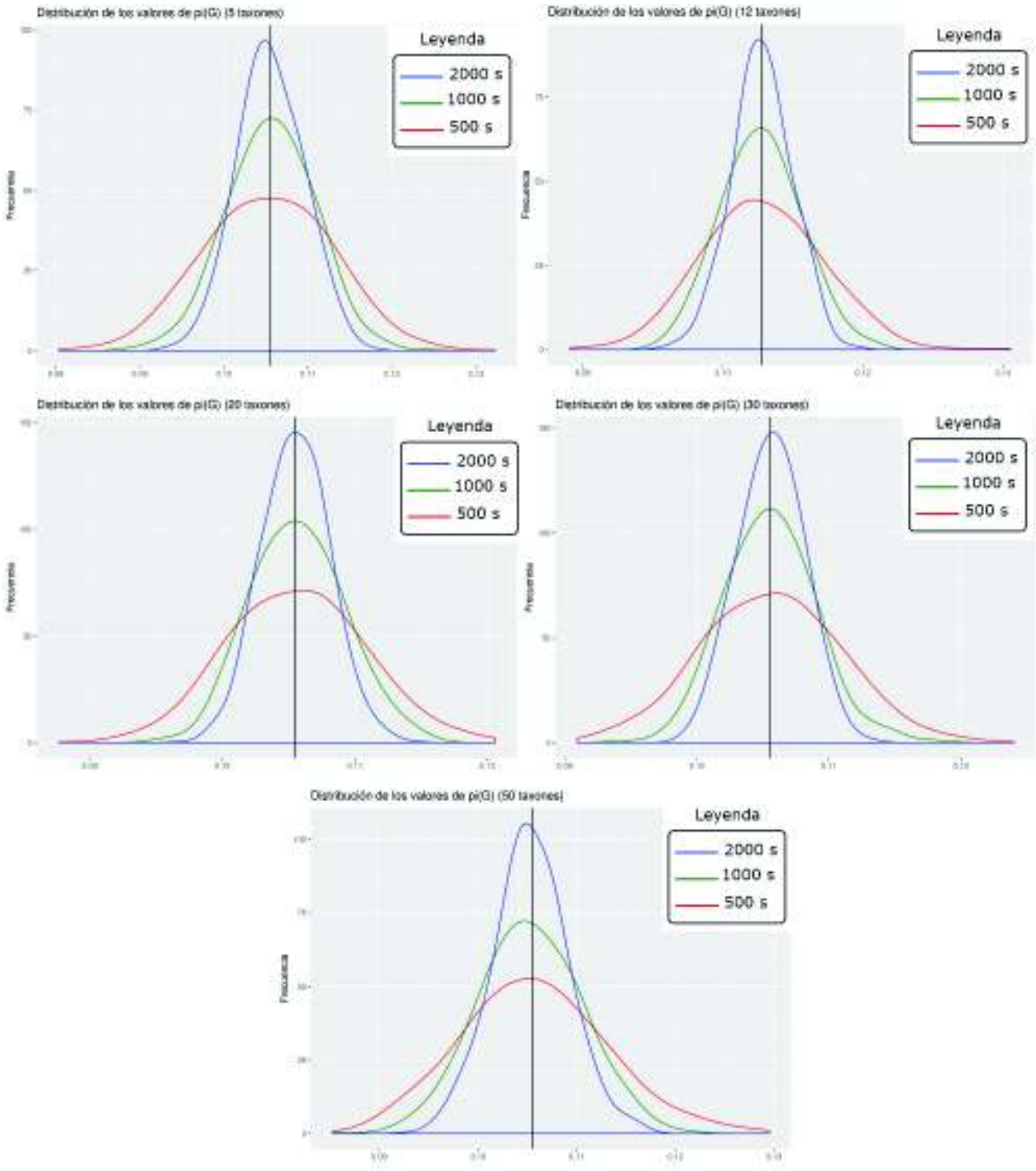


FIGURA 6.8: Estimaciones de π_G para 5, 12, 20, 30 y 50 taxones

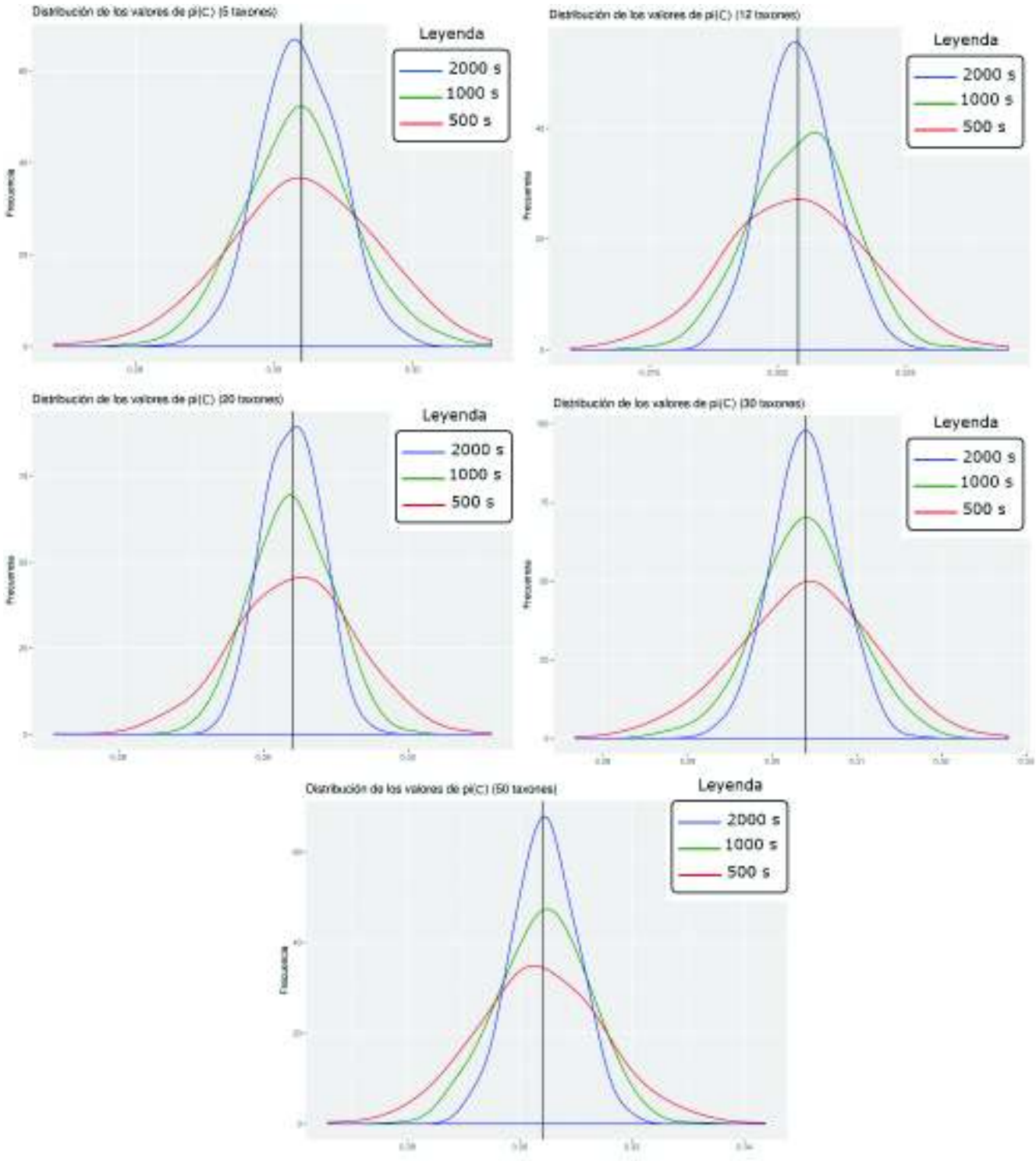


FIGURA 6.9: Estimaciones de π_C para 5, 12, 20, 30 y 50 taxones

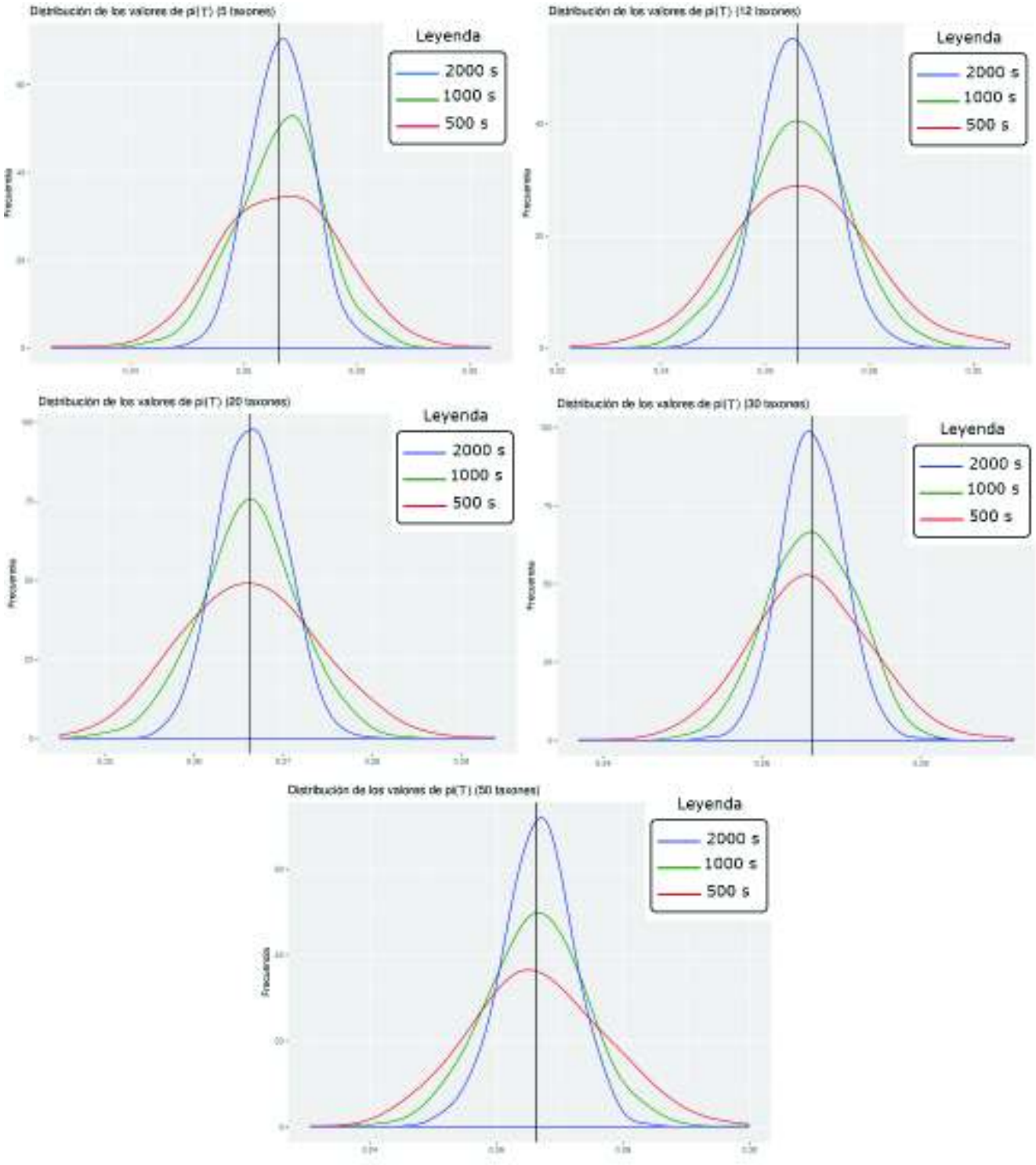


FIGURA 6.10: Estimaciones de π_T para 5, 12, 20, 30 y 50 taxones

6.3.3 Distancias entre árboles

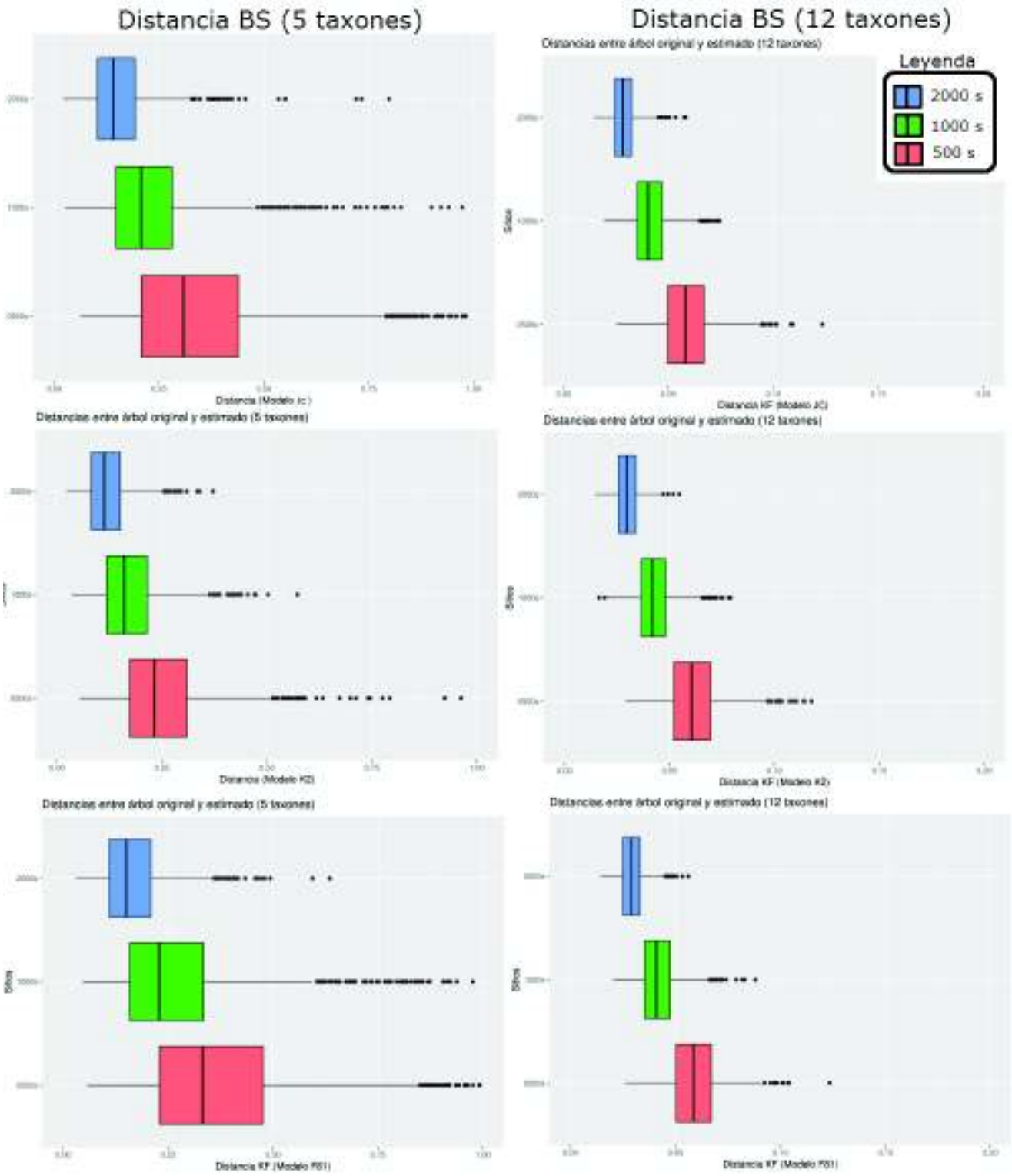


FIGURA 6.11: Branch Score árboles de 5 y 12 taxones

En la figura (6.11) se encuentran los diagramas de caja para los valores de las distancia *Branch Score* entre el árbol original y el estimado de 5 y 12 taxones.

Los valores de la distancia RF para estas instancias no pasaban de 2, es decir, no existían más de dos ramas diferentes en los árboles estimados en comparación con el árbol original. Estos valores y sus frecuencias se muestran en las tablas (6.1) y (6.2).

Modelo JC					
Dist. RF 500 sitios	Frec.	Dist. RF 1000 sitios	Frec.	Dist. RF 2000 sitios	Frec.
0	854	0	966	0	997
2	146	2	34	2	3
Modelo K2P					
0	996	0	1000	0	1000
2	4				
Modelo F81					
0	814	0	930	0	998
2	186	2	70	2	2

TABLA 6.1: Distancia RF para árboles de 5 taxones

Modelo JC					
Dist. RF 500 sitios	Frec.	Dist. RF 1000 sitios	Frec.	Dist. RF 2000 sitios	Frec.
0	994	0	999	0	1000
2	6	2	1		
Modelo K2P					
0	981	0	1000	0	1000
2	19				
Modelo F81					
0	985	0	1000	0	1000
2	15				

TABLA 6.2: Distancia RF para árboles de 12 taxones

Los resultados de la figura (6.11) junto con las tablas (6.1) y (6.2) nos permiten concluir que la topología estimada, esto es la forma de los árboles estimados, en general es la misma que la del árbol original y que solo existen variaciones en las estimaciones de los largos de ramas. Además, los valores de las distancias Branch Score y RF se reducen conforme aumenta la longitud de las secuencias.

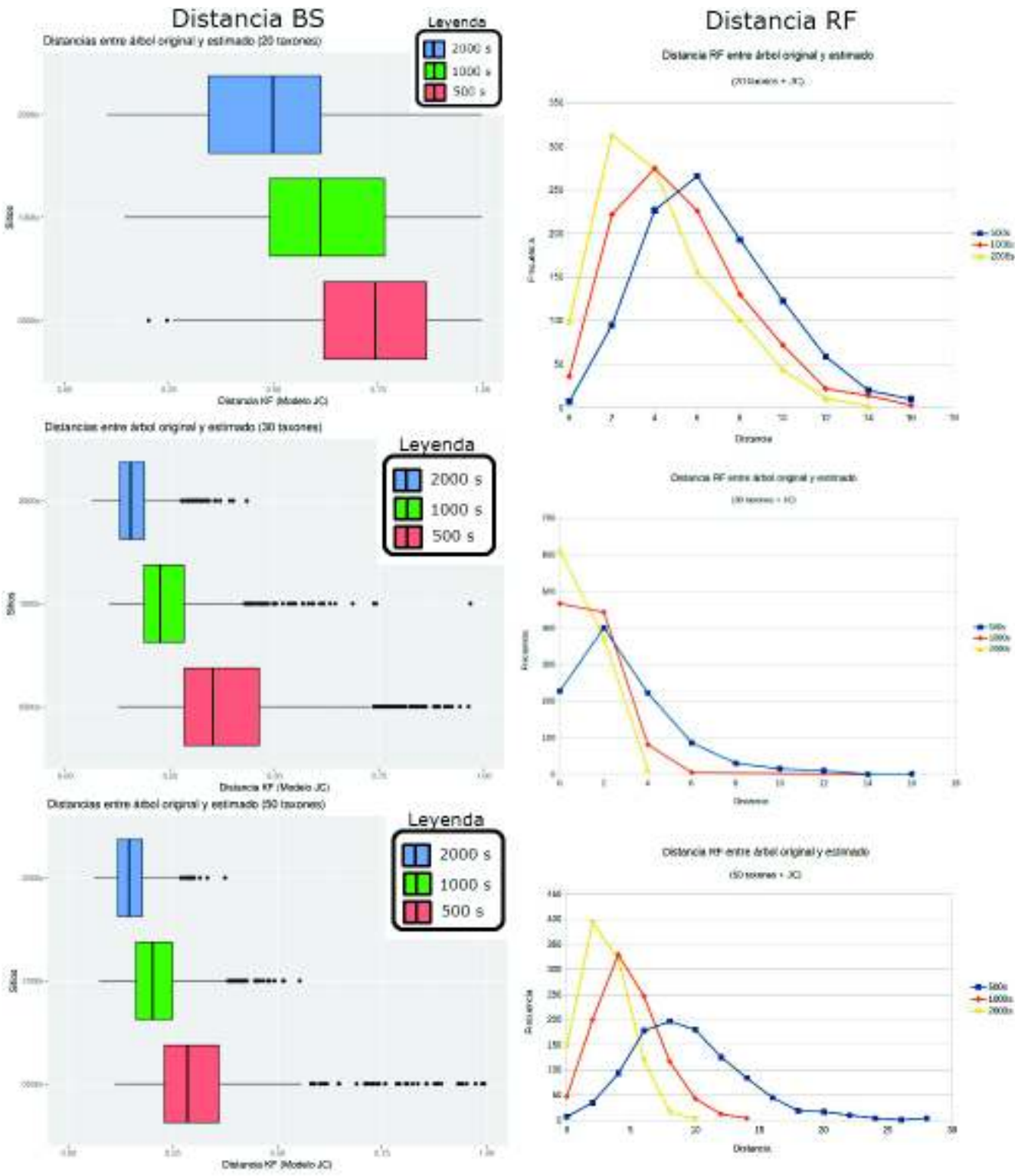


FIGURA 6.12: Instancias con modelo JC de 20, 30 y 50 taxones

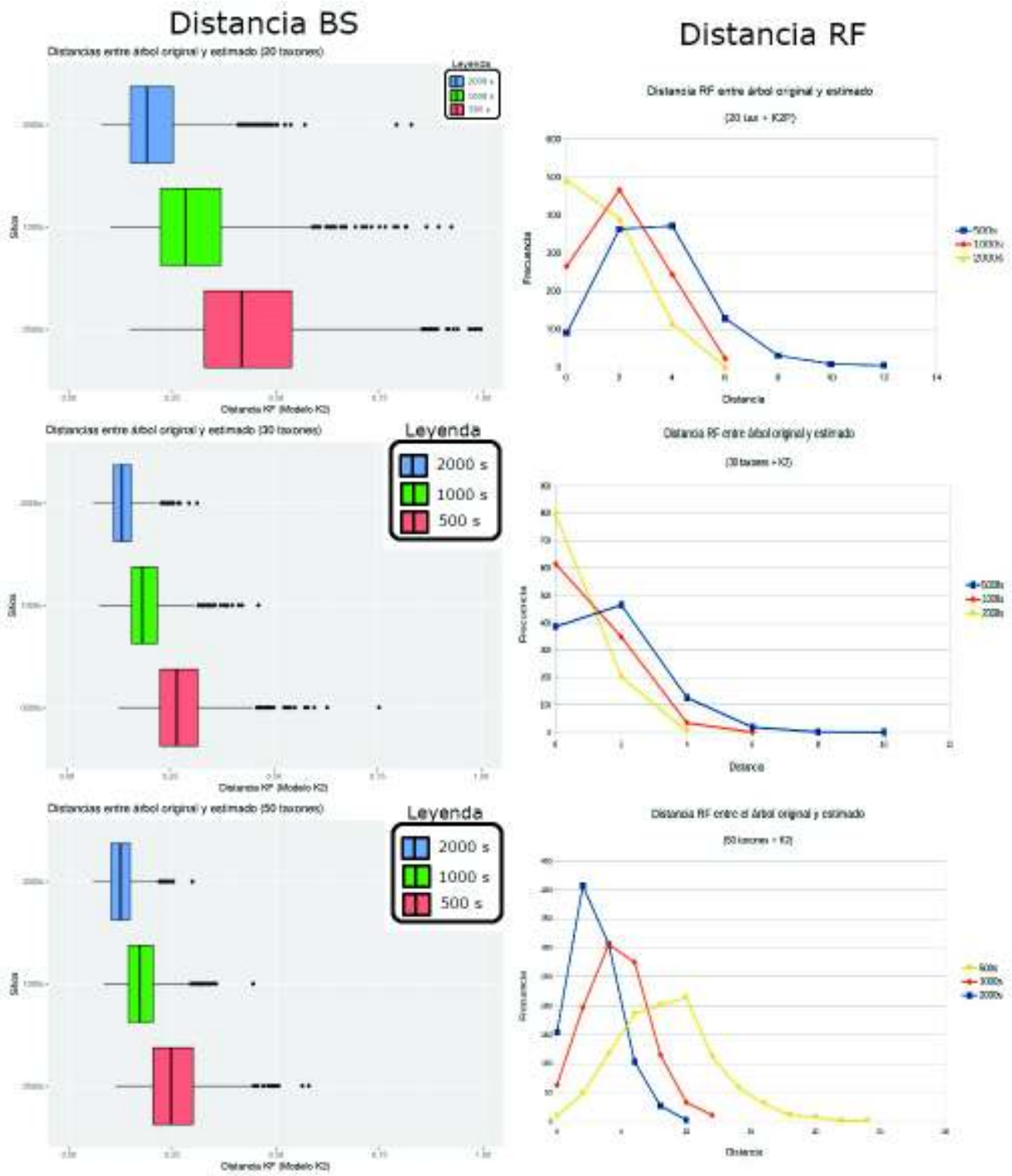


FIGURA 6.13: Instancias con modelo K2P de 20, 30 y 50 taxones

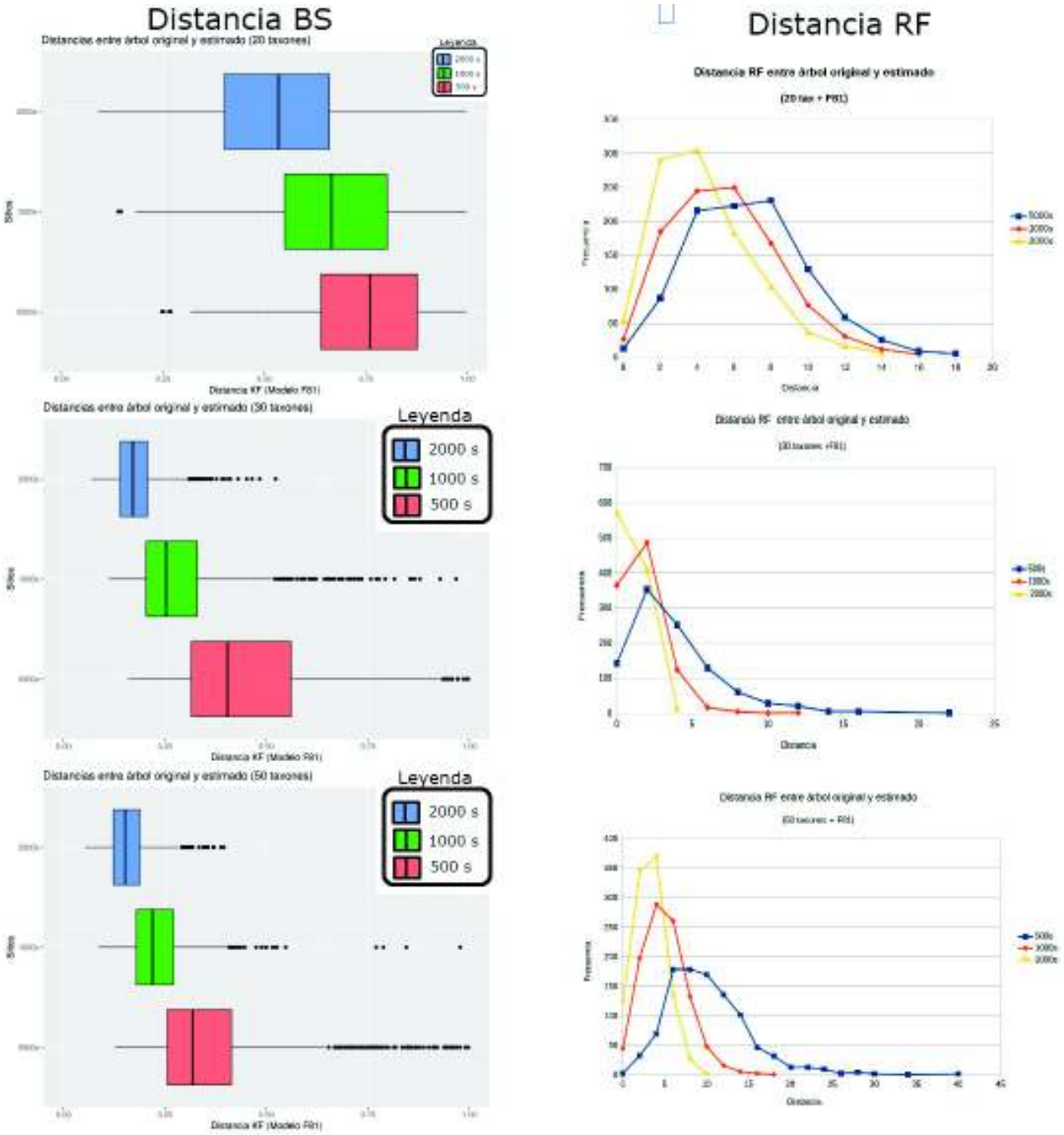


FIGURA 6.14: Instancias con modelo F81 de 20, 30 y 50 taxones

En las figuras (6.12), (6.13) y (6.14) se encuentran los resultados de las distancias Branch Score y RF para las instancias de 20, 30 y 50 taxones simuladas según los modelos JC, K2P y F81, respectivamente. Estas distancias se calcularon entre el árbol original y los reconstruidos por Máxima Verosimilitud.

Al aumentar el número de taxones, aumenta la cantidad de ramas del árbol y por tanto la cantidad de parámetros a estimar es mayor, por eso las estimaciones de la topología para las instancias de 20, 30 y 50 taxones ya no son tan exactas comparadas con las estimaciones para 5 y 12 taxones. Sin embargo, ambas distancias se hacen más pequeñas conforme se aumenta el número de columnas en las matrices de secuencias.

6.3.4 Selección de Modelos

En las tablas (6.3), (6.4), (6.5), (6.6) y (6.7) se muestran los resultados de los análisis de selección de modelos para los árboles de 5, 12, 20, 30 y 50 taxones respectivamente. El objetivo de este análisis era comprobar el comportamiento del algoritmo de selección de modelos de `IQ-TREE` y puede considerarse como un aporte extra de este trabajo de titulación.

La información está ordenada de la siguiente manera: en la primera columna se indica el criterio de selección BIC, AIC y AICc. En la segunda columna se detalla la información para las instancias con 500 sitios, en la tercera está la información para las instancias de 1000 sitios y en la cuarta, la información para las de 2000 sitios. Para cada instancia se muestra dos columnas, la primera es el modelo seleccionado y la segunda muestra la frecuencia con la que fue seleccionado por el respectivo criterio.

Cabe indicar que solo se muestra la información de los modelos más frecuentemente seleccionados y los demás se agrupan en la categoría "OTROS. La razón es que se seleccionaba una amplia cantidad de modelos, sobre todo por parte de los criterios de Akaike.

Los resultados de las tablas (6.3), (6.4), (6.5), (6.6) y (6.7) muestran que el comportamiento del BIC es el mejor en comparación a los criterios AIC y AICc, pues escoge al verdadero modelo para los datos con mayor frecuencia (en más del 90 por ciento de las veces) y la precisión con la que lo selecciona es mayor conforme aumenta el tamaño de las secuencias.

Por otro lado, los criterios AIC y AICc escogen una mayor cantidad de modelos diferentes para los datos simulados, pero el verdadero modelo es seleccionado en al menos el 50 por ciento de las veces.

Criterio	I: 5tax + JC +500s		I: 5tax + JC +1000s		I: 5tax + JC +2000s	
	Modelo	**Frec.**	**Modelo**	**Frec.**	**Modelo**	**Frec.**
BIC	JC	982	JC	988	JC	990
	K2P	11	K2P	11	K2P	10
	K3P	3	K3P	1		
	OTROS	4				
AIC	JC	589	HKY	7	JC	596
	K2P	90	K3P	55	K2P	90
	K3P	73	F81	58	K3P	64
	F81	47	K2P	103	F81	49
	HKY	5	JC	598	HKY	12
	Otros	196	Otros	179	Otros	189
AICc	JC	619	JC	606	JC	600
	K2P	85	K2P	102	K2P	90
	K3P	73	F81	58	K3P	64
	F81	42	K3P	52	F81	48
	HKY	5	HKY	7	HKY	11
	Otros	176	Otros	175	Otros	187
Criterio	**I: 5tax + K2P +500s**		**I: 5tax + K2P +1000s**		**I: 5tax + K2P +2000s**	
BIC	K2P	975	K2P	981	K2P	984
	K3P	16	K3P	12	K3P	10
	Tne	8	HKY	1	Tne	6
	HKY	1	OTROS	6		
AIC	K2P	557	K2P	561	K2P	541
	K3P	122	K3P	108	K3P	107
	HKY	50	HKY	47	HKY	44
	OTROS	271	OTROS	284	OTROS	308
AICc	K2P	579	K2P	574	K2P	550
	K3P	120	K3P	109	K3P	104
	HKY	48	HKY	43	HKY	43
	OTROS	253	OTROS	274	OTROS	303
Criterio	**I: 5tax + F81 +500s**		**I: 5tax + F81 +1000s**		**I: 5tax + F81 +2000s**	
BIC	F81	971	F81	991	F81	990
	HKY	17	K3P	4	HKY	8
	OTROS	12	HKY	3	TPM	2
			OTROS	2		
AIC	F81	587	F81	620	F81	614
	HKY	67	HKY	76	HKY	80
	K3P	52	K3P	47	K3P	59
	GTR	6	GTR	6	GTR	5
	OTROS	288	OTROS	251	OTROS	242
AICc	F81	616	F81	628	F81	616
	HKY	65	HKY	77	HKY	81
	K3P	49	K3P	46	K3P	59
	GTR	2	GTR	6	GTR	5
	OTROS	268	OTROS	243	OTROS	239

TABLA 6.3: Selección de modelo para árbol de 5 taxones

Criterio	I: 12tax + JC +500s		I: 12tax + JC +1000s		I: 12tax + JC +2000s	
	Modelo	**Frec.**	**Modelo**	**Frec.**	**Modelo**	**Frec.**
BIC	JC	992	JC	993	JC	991
	K2P	5	K2P	5	K2P	9
	K3P	1	OTROS	2		
	OTROS	2				
AIC	JC	591	JC	562	JC	629
	K2P	92	K2P	87	K2P	80
	F81	76	F81	77	K3P	56
	K3P	63	K3P	61	F81	51
	OTROS	178	OTROS	213	OTROS	184
AICc	JC	651	JC	600	JC	644
	K2P	85	K2P	87	K2P	82
	F81	66	F81	65	K3P	53
	K3P	55	K3P	62	F81	46
	OTROS	143	OTROS	186	OTROS	175
Criterio	**I: 12tax + K2P +500s**		**I: 12tax + K2P +1000s**		**I: 12tax + K2P +2000s**	
BIC	K2P	972	K2P	984	K2P	992
	K3P	14	K3P	8	TNe	6
	OTROS	14	OTROS	8	K3P	2
AIC	K2P	544	K2P	560	K2P	578
	K3P	107	K3P	111	K3P	90
	HKY	51	HKY	53	HKY	58
	OTROS	298	OTROS	276	OTROS	274
AICc	K2P	616	K2P	583	K2P	589
	K3P	96	K3P	109	K3P	88
	HKY	48	HKY	49	HKY	53
	OTROS	240	OTROS	259	OTROS	270
Criterio	**I: 12tax + F81 +500s**		**I: 12tax + F81 +1000s**		**I: 12tax + F81 +2000s**	
BIC	F81	967	F81	978	F81	989
	HKY	17	HKY	16	HKY	6
	K3P	2	K3P	2	K3P	2
	OTROS	14	OTROS	4	OTROS	3
AIC	F81	555	F81	552	F81	569
	HKY	79	HKY	77	HKY	74
	K3P	50	K3P	55	K3P	53
	OTROS	316	OTROS	316	OTROS	304
AICc	F81	613	F81	579	F81	583
	HKY	74	HKY	81	HKY	74
	K3P	44	K3P	50	K3P	53
	OTROS	269	OTROS	290	OTROS	290

TABLA 6.4: Selección de modelo para árbol de 12 taxones

Criterio	I: 20tax + JC +500s		I: 20tax + JC +1000s		I: 20tax + JC +2000s	
	Modelo	**Frec.**	**Modelo**	**Frec.**	**Modelo**	**Frec.**
BIC	JC	979	JC	987	JC	986
	K2P	16	K2P	12	K2P	11
	K3P	2	Tne	1	K3P	1
	Tne	3	Tne	2		
AIC	JC	586	JC	616	JC	542
	K2P	94	K2P	111	K2P	124
	K3P	71	K3P	69	K3P	77
	F81	34	F81	32	F81	30
	OTROS	215	OTROS	172	OTROS	227
AICc	JC	683	JC	667	JC	567
	K2P	88	K2P	106	K2P	127
	K3P	59	K3P	57	K3P	69
	F81	31	F81	24	F81	30
	OTROS	139	OTROS	146	OTROS	207
Criterio	**I: 20tax + K2P +500s**		**I: 20tax + K2P +1000s**		**I: 20tax + K2P +2000s**	
BIC	K2P	977	K2P	984	K2P	989
	K3P	11	K3P	9	K3P	2
	OTROS	12	OTROS	7	OTROS	9
AIC	K2P	535	K2P	563	K2P	565
	K3P	116	K3P	114	K3P	119
	HKY	60	HKY	46	HKY	43
	OTROS	289	OTROS	277	OTROS	273
AICc	K2P	650	K2P	615	K2P	585
	K3P	99	K3P	102	K3P	116
	HKY	40	HKY	37	HKY	43
	OTROS	211	OTROS	246	OTROS	256
Criterio	**I: 20tax + F81 +500s**		**I: 20tax + F81 +1000s**		**I: 20tax + F81 +2000s**	
BIC	F81	976	F81	990	F81	992
	HKY	13	HKY	6	HKY	6
	K3P	2	OTROS	4	OTROS	2
	OTROS	9				
AIC	F81	582	F81	571	F81	566
	HKY	68	HKY	77	HKY	85
	K3P	56	K3P	46	K3P	44
	OTROS	294	OTROS	306	OTROS	305
AICc	F81	672	F81	610	F81	585
	HKY	70	HKY	77	HKY	86
	K3P	40	K3P	43	K3P	41
	OTROS	218	OTROS	270	OTROS	288

TABLA 6.5: Selección de modelo para árbol de 20 taxones

Criterio	I: 30tax + JC +500s		I: 30tax + JC +1000s		I: 30tax + JC +2000s	
	Modelo	**Frec.**	**Modelo**	**Frec.**	**Modelo**	**Frec.**
BIC	JC	986	JC	989	JC	991
	K2P	13	K2P	11	K2P	9
	F81	1				
AIC	JC	615	JC	570	JC	564
	K2P	96	K2P	125	K2P	123
	K3P	67	K3P	59	K3P	64
	F81	38	F81	37	F81	35
	OTROS	184	OTROS	209	OTROS	214
AICc	JC	742	JC	654	JC	597
	K2P	91	K2P	125	K2P	123
	K3P	45	K3P	40	K3P	62
	F81	27	F81	30	F81	33
	OTROS	95	OTROS	151	OTROS	185
Criterio	**I: 30tax + K2P +500s**		**I: 30tax + K2P +1000s**		**I: 30tax + K2P +2000s**	
BIC	K2P	961	K2P	978	K2P	986
	K3P	10	K3P	6	K3P	8
	F81	5	OTROS	16	OTROS	6
	OTROS	24				
AIC	K2P	570	K2P	558	K2P	582
	K3P	110	K3P	104	K3P	95
	HKY	48	HKY	46	HKY	57
	OTROS	272	OTROS	292	OTROS	266
AICc	K2P	738	K2P	627	K2P	619
	K3P	78	K3P	97	K3P	85
	HKY	31	HKY	36	HKY	50
	OTROS	153	OTROS	240	OTROS	246
Criterio	**I: 30tax + F81 +500s**		**I: 30tax + F81 +1000s**		**I: 30tax + F81 +2000s**	
BIC	F81	962	F81	973	F81	992
	HKY	18	HKY	17	HKY	5
	K3P	2	K3P	2	OTROS	3
	OTROS	18	OTROS	8		
AIC	F81	529	F81	509	F81	534
	HKY	79	HKY	85	HKY	62
	K3P	57	K3P	50	K3P	45
	OTROS	335	OTROS	356	OTROS	359
AICc	F81	672	F81	591	F81	574
	HKY	74	HKY	91	HKY	66
	K3P	34	K3P	46	K3P	41
	OTROS	220	OTROS	272	OTROS	319

TABLA 6.6: Selección de modelo para árbol de 30 taxones

Criterio	I: 50tax + JC +500s		I: 50tax + JC +1000s		I: 50tax + JC +2000s	
	Modelo	**Frec.**	**Modelo**	**Frec.**	**Modelo**	**Frec.**
BIC	JC	977	JC	993	JC	993
	K2P	22	K2P	7	K2P	7
	Tne	1				
AIC	JC	619	JC	673	JC	618
	K2P	101	K2P	87	K2P	90
	K3P	70	K3P	67	K3P	44
	F81	28	F81	33	F81	80
	OTROS	182	OTROS	140	OTROS	168
AICc	JC	834	JC	784	JC	681
	K2P	79	K2P	77	K2P	92
	K3P	29	K3P	47	K3P	33
	F81	4	F81	23	F81	60
	OTROS	54	OTROS	69	OTROS	134
Criterio	**I: 50tax + K2P +500s**		**I: 50tax + K2P +1000s**		**I: 50tax + K2P +2000s**	
BIC	K2P	975	K2P	979	K2P	986
	K3P	12	K3P	12	K3P	7
	OTROS	13	OTROS	9	OTROS	7
AIC	K2P	590	K2P	561	K2P	563
	K3P	116	K3P	108	K3P	122
	HKY	13	HKY	17	HKY	47
	OTROS	281	OTROS	314	OTROS	268
AICc	K2P	808	K2P	682	K2P	606
	K3P	69	K3P	93	K3P	113
	HKY	8	HKY	11	HKY	44
	OTROS	115	OTROS	214	OTROS	237
Criterio	**I: 50tax + F81 +500s**		**I: 50tax + F81 +1000s**		**I: 50tax + F81 +2000s**	
BIC	F81	965	F81	980	F81	993
	HKY	23	HKY	11	HKY	7
	K3P	3	OTROS	9		
	OTROS	9				
AIC	F81	519	F81	532	F81	660
	HKY	59	HKY	52	HKY	88
	K3P	50	K3P	50	K3P	51
	GTR	9	GTR	12	GTR	5
	OTROS	363	OTROS	354	OTRO	196
AICc	F81	757	F81	662	F81	701
	HKY	60	HKY	51	HKY	83
	K3P	20	K3P	25	K3P	44
	OTROS	163	OTROS	262	OTROS	172

TABLA 6.7: Selección de modelo para árbol de 50 taxones

7 Conclusiones y trabajo futuro

En este trabajo se ha implementado un programa que simula secuencias de ADN utilizando las facilidades del lenguaje de programación R. Como un aporte extra, se ha desarrollado una aplicación web que permite graficar y modificar árboles filogenéticos, también implementada utilizando librerías de R.

Para poder desarrollar el programa que simula secuencias de ADN se ha estudiado el fundamento matemático que modela la evolución del ADN como una cadena de Markov homogénea en tiempo continuo.

Específicamente, se supone que la evolución de una secuencia de ADN ancestral en una secuencia descendiente, se puede modelar por medio de una cadena de Markov homogénea en tiempo continuo, la cual tiene asociada una matriz generadora Q y un vector de distribución estacionaria π. Esta combinación (Q, π) es lo que los biólogos suelen llamar *modelo evolutivo*.

Cada sitio en la secuencia de ADN del descendiente es visto como una variable aleatoria y por simplicidad, se supone que estos sitios son independientes e idénticamente distribuidos.

Para modelar la evolución en árboles que tienen más de un par *ancestro–descendiente* conectados por ramas, se extiende la idea anterior para cada una de estas ramas. Pero, por simplicidad, se supone que todas las ramas de un árbol filogenético tienen asociada la misma matriz generadora Q y se supone que el proceso comienza con la distribución estacionaria π. En otras palabras, se supone que la evolución en las ramas del árbol ha ocurrido de acuerdo al modelo evolutivo cuya matriz generadora es Q y que tiene distribución estacionaria π.

Esta modelización se explica formalmente en el capítulo 2.

Imponiendo ciertas restricciones sobre la matriz Q y el vector π se obtienen modelos con propiedades matemáticas útiles, los cuales han sido nombrados en honor a las personas que los han desarrollado y publicado. Por eso tenemos los modelos de Jukes–Cantor, Kimura–2, Kimura–3, F81, etcétera.

A partir de estos modelos evolutivos, se puede definir distancias entre secuencias de ADN, tal como se indica en el capítulo 3. Estas distancias son utilizadas en métodos de reconstrucción filogenética por medio de Matrices de Distancias, por ejemplo Neighbor-joining, UPGMA o el método Fitch-Margoliash.

Sin embargo, el método que se trata en este trabajo es el de reconstrucción por Máxima Verosimilitud. Todos los métodos de reconstrucción filogenética reciben como dato una matriz compuesta por secuencias de ADN de longitud l correspondientes a N taxones. Estos taxones pueden ser, por ejemplo, especies de animales y la matriz de datos también suele ser llamada *alineamiento*.

Como se explica en la sección (4.3), bajo suposiciones de independencia entre sitios y linajes, se puede definir una función de verosimilitud para cada columna de la matriz de secuencias de ADN dados un modelo evolutivo y un árbol filogenético con largos de ramas.

Para calcular el valor de la función de verosimilitud de un alineamiento, se utilizó el Algoritmo Pruning de Felsenstein. Tal como se explica en la sección (4.4), el Algoritmo Pruning permite calcular de manera eficiente la verosimilitud de una alineamiento utilizando programación dinámica.

En la sección (4.5) se explica el Algoritmo de Reconstrucción Filogenética por Máxima Verosimilitud. Igual que con todos los métodos de reconstrucción, la tarea computacionalmente más costosa es la exploración del espacio de árboles pues, para un árbol con $N \geq 3$ taxones (hojas), la cantidad de árboles bifurcados y enraizados que existen es $(2N-3)!/(2^{N-2}(N-2)!)$.

Entonces, el verdadero problema de la reconstrucción filogenética es el crecimiento super–exponencial del espacio de árboles. Notar que para 5 taxones existen 105 posibles árboles enraizados, pero simplemente aumentar la cantidad de taxones a 12, hace que el número de posibles árboles crezca a más de 1.37×10^{10}. Esto es más preocupante aún porque en la práctica se busca hacer reconstrucciones de más de 50 taxones.

Dejando de lado el problema de la reconstrucción filogenética, el objetivo de este trabajo era crear un programa que, en cierta forma, hiciera el proceso inverso al de la reconstrucción. Lo que se buscaba era crear un programa, el cual a partir de un árbol filogenético fijo, cuya evolución en sus ramas ocurre de acuerdo a un modelo evolutivo con generado Q y vector de distribución estacionaria π, permita simular secuencias de ADN de largo l para los taxones.

Como se explica en el capítulo 5, esto se logró implementando la función `sim_evolución()` en `R`, utilizando las librerías `ape` y `expm`. Esta función simula la evolución de un sitio, o de una columna de la matriz, empezando desde la raíz hasta llegar a las hojas.

La función `sim_evolución()` básicamente simula la cantidad de columnas que el usuario desee para la matriz o alineamiento.

En el capítulo 5 también se explica el desarrollo de la aplicación web que permite graficar y modificar árboles filogenéticos en formato Newick y NEXUS. Esta aplicación fue creada utilizando funciones de las librerías `Shiny` y `ape`.

Luego de haber implementado la función `sim_evolución()`, se generaron un total de 45000 matrices con esta función, utilizando árboles de distintos tamaños y diferentes modelos evolutivos, tal como se explica en el capítulo 6, donde también se muestran los resultados de los análisis realizados a estos datos simulados.

Sobre estas 45000 matrices simuladas se corrieron reconstrucciones de árboles filogenéticos y el algoritmo de selección de modelos del programa `IQ-TREE`. Los resultados de estos análisis se encuentran resumidos mediante gráficos y tablas en la sección (6.3). En esta sección se puede ver que el comportamiento general de los estimadores de máxima verosimilitud de

los parámetros numéricos es el esperado, es decir, la precisión de las estimaciones aumenta conforme aumenta la longitud de las secuencias, que en nuestro caso representa el tamaño de la muestra.

También se puede ver cómo la complejidad del problema de la reconstrucción crece conforme aumenta la cantidad de taxones para los árboles. Esto se puede comprobar con los gráficos de las distancias entre árboles, en las imágenes (6.11), (6.12), (6.13) y (6.14).

La distancia RF mide la diferencia en la topología de los árboles estimados en comparación con el árbol original con el que se simularon cada uno de los datos. Los resultados muestran que las diferencias entre las topologías estimadas con la original son cada vez mayores conforme el tamaño de los árboles aumenta. Así pues, para árboles relativamente pequeños de 5 y 12 taxones, el valor de la distancia RF no supera el valor de 2. Pero para árboles de 50 taxones, esta distancia alcanza el valor de 40. Esto significa que en las instancias de 50 taxones los árboles estimados se diferencian del árbol original hasta en 40 ramas.

Los resultados de las distancia Branch Score muestran que su valor se reduce conforme se aumenta el tamaño de las secuencias. Como sabemos, el Branch Score puede llevar a hacer conclusiones erróneas sobre la topología de los árboles, si no se analiza en conjunto con la distancia RF.

Por ejemplo, si examináramos solamente la distancia Branch Score para 20, 30 y 50 taxones, podríamos concluir erróneamente que las estimaciones de la topología para 50 taxones son mejores que las de 20 y 30 juntas porque el valor de esta distancia es menor para los árboles de 50 taxones, en comparación a los de 20 y 30. Pero, en realidad, las diferencias en la topología para árboles de 50 taxones son las mayores, como lo indican los resultados de la distancia RF.

Los resultados del Capítulo 6 muestran que, en general, los estimadores de Máxima Verosimilitud para árboles filogenéticos se comportan bien conforme aumenta la longitud de las secuencias. Pero esta longitud también representa un problema en la práctica, pues los biólogos suelen trabajar con alineamientos de menos de 1000 sitios. En otras palabras, en la práctica se tendría una visión bastante pobre de lo que verdaderamente pasó durante la evolución de las secuencias de ADN.

En la sección (6.3.4) se analizó el comportamiento del algoritmo de selección de modelos del programa `IQ-TREE`. En general, los resultados muestran que el verdadero modelo para los datos es escogido en más del 50 por ciento de las veces.

De los tres criterios de selección, el BIC es el que escoge los modelos de una forma más precisa. Para todas las instancias, este criterio escoge el modelo con el que se han simulado los datos en más del 90 por ciento de las veces. Además la frecuencia con la que se escoge el verdadero modelo aumenta conforme aumenta el tamaño de las secuencias. Esto se puede ver en las tablas (6.3), (6.4), (6.5), (6.6) y (6.7).

Los análisis del Capítulo 6 corresponden a reconstrucciones de árboles filogenéticos por Máxima Verosimilitud, no obstante, existen muchas otras

formas de abordar el problema de la reconstrucción, por ejemplo: la Inferencia Bayesiana, los métodos que usan Matrices de Distancias o la Máxima Parsimonia. Cada uno de estos tiene sus fundamentos y características específicas.

La razón por la cual se decidió trabajar con Máxima Verosimilitud en este trabajo fue, en primer lugar, por las propiedades estadísticas de sus estimadores y en segundo lugar, por la relativa rapidez de la ejecución de los análisis con este método, en comparación a los de Inferencia Bayesiana. Para tener una idea, las reconstrucciones por Máxima Verosimilitud para todas las instancias de 5 taxones (9000 en total), se demora más o menos un par de horas utilizando un computador portátil. Por otro lado, el realizar estos mismos análisis utilizando Inferencia Bayesiana y la capacidad de cómputo del HPC–MODEMAT, puede tardar más de 10 horas en terminar y este tiempo de cómputo será mayor cuanto mayor sea la cantidad de taxones. Esto se debe a que los métodos de Inferencia Bayesiana utilizan Cadenas de Markov Monte Carlo para explorar el espacio de árboles, además el objetivo de este método es obtener una distribución de probabilidad para los árboles y los parámetros numéricos. El método de Máxima Verosimilitud, en cambio, busca obtener un único árbol y un único conjunto de parámetros numéricos que maximice la función de verosimilitud.

Trabajo futuro

Hay que recalcar que en este trabajo, se ha implementado un programa que simula secuencias de ADN para árboles cuyas ramas han evolucionado de acuerdo a un modelo GTR o una de sus variantes más restringidas. Pero no se han tratado otras variantes de modelos evolutivos que incorporan, por ejemplo, heterogeneidad con distribución Gamma entre sitios del alineamiento y el desarrollar programas que incorporen este tipo de heterogeneidad para simular secuencias queda como trabajo futuro.

Otro punto a mejorar son las medidas de distancia entre árboles métricos, pues en este trabajo se comprobó que el Branch Score desarrollado por Kuhner y Felsenstein no permite medir correctamente las diferencias entre árboles con largos de ramas muy cercanas a cero.

Otro punto en el cual se podría trabajar a futuro es definir mejor qué consideramos como tamaño de muestra para hacer reconstrucciones filogenéticas. Se sabe que la calidad de la información también depende del tipo de gen de donde se extraiga. Entonces podríamos considerar el tipo y la cantidad de genes de los cuales se obtuvo la información para definir el tamaño de la muestra. Pero esta cuestión en especial debe ser resuelta junto con biólogos y no solo por matemáticos.

Sin duda el problema más complejo de la reconstrucción filogenética es el desarrollar métodos que permitan explorar el espacio de árboles de una manera más eficiente. Sin embargo, para llegar al punto de la reconstrucción se deben hacer muchos otros procesos previos, entre ellos están: la extracción del ADN, su secuenciación y posteriormente su alineamiento. Estos son

temas en los que los matemáticos pueden trabajar, sobre todo en la parte de alineamiento de secuencias.

A Código en R del programa de simulación de secuencias de ADN

```
#########################################################
#           Simular secuencias de ADN                   #
#########################################################
library(ape)
library(expm)

sim_nuc <- function(p_0){
  aleat <- runif(1)
  a <- sum(p_0[1:2])
  b <- sum(p_0[1:3])

  if(0<=aleat & aleat<p_0[1])
    p <- c(1,0,0,0)
  else if(p_0[1]<=aleat& aleat<a)
    p <- c(0,1,0,0)
  else if(a<=aleat & aleat<b)
    p <- c(0,0,1,0)
  else if(b<=aleat & aleat<=1)
    p <- c(0,0,0,1)

  return(p)
}

codificar <- function(p_bin){
  if(p_bin[1]>=0.99)
    nuc <- "A"
  else if(p_bin[2]>=0.99)
    nuc <- "G"
  else if(p_bin[3]>=0.99)
    nuc <- "C"
  else if(p_bin[4]>=0.99)
    nuc <- "T"
  return(nuc)
}


sim_evolucion <- function(tr, Q, pi, n_sitios){
  n_taxa <- length(tr$tip.label)
```

```
  n_ramas <- length(tr$edge.length)
  n_int <- tr$Nnode

  nuc_taxa <- matrix(nrow = n_taxa, ncol = n_sitios)#matriz para
  #guardar los nucleótidos de la taxa
  rownames(nuc_taxa) <- tr$tip.label

  nodos_int <- matrix(nrow =n_int ,ncol = 4)#matriz para guardar
  #vectores binarios de nodos internos


  for(j in 1:n_sitios){
    p_0 <- sim_nuc(pi)
    #print(p_0)
    nodos_int[1, ] <- p_0

    for(i in 1:n_ramas){
      n_ini <- tr$edge[i,1] - n_taxa
      n_fin <- tr$edge[i,2] - n_taxa

      M <- expm(Q*tr$edge.length[i])

      p_1 <- nodos_int[n_ini, ]%*%M
      p_bin <- sim_nuc(p_1)
      if(n_fin>0)
        nodos_int[n_fin, ] <- p_bin
      else{
        nuc <- codificar(p_bin)
        nuc_taxa[(n_fin+n_taxa), j] <- nuc
      }
      #cat("Matriz nodos internos")
      #print(nodos_int)
    }
  }
  rownames(nuc_taxa) <- tr$tip.label
  return(nuc_taxa)
}
#########################################################
#                    Ejemplo                            #
#########################################################

arbol <- "(((Esp1:0.01,Esp2:0.01):0.02,Esp3:0.03):0.01,
(Esp4:0.01,Esp5:0.01):0.03,Out:0.05);"
tr <- read.tree(text = arbol)
plot(tr, type="p", use.edge.length = T)
nodelabels(frame = "c")
edgelabels(text = tr$edge.length,frame = "none")
```

```
pi <- c(0.25,0.25,0.25,0.25)
kappa <- 5
beta <- 1/(2+kappa)
alfa <- kappa*beta
K80 <- matrix(data = c(-(alfa+2*beta),alfa,beta,beta,
                       alfa,-(alfa+2*beta), beta, beta,
                       beta, beta, -(alfa+2*beta), alfa,
                       beta, beta, alfa, -(alfa+2*beta)),
                       nrow = 4, ncol = 4)
ejemplo <- sim_evolucion(tr, K80, pi, 800)
```

B Código en R de la aplicación web para graficar filogenias

B.1 Interfaz de usuario

```
library(shiny)
shinyUI(
navbarPage(title = "Yura Shiny App",
tabPanel(title = "Ayuda/Subir archivo",
tags$img(style = "float: left;width:20%;",
  src = "http://www.epn.edu.ec/gui/header/logo.svg",
  alt = "EPN logo"),
  tags$img(style = "float: right;width:15%;",
  src = "https://www.r-project.org/Rlogo.png",
   alt = "R logo"),
p(style="font-family:verdana; color:grey; font-size:x-large;",
'Yura es una aplicación web que permite dibujar árboles
 filogenéticos de un archivo en formato nexus o newick.'),
p(style="font-family:verdana; color:grey; font-size:x-large;",
 'Ha sido creada utilizando las librerías Shiny y
 Ape del programa R.'),
p(style="font-family:verdana; color:grey; font-size:x-large;",
  'Preguntas, comentarios o sugerencias al email:'),
    a(style="font-size:large",
   "raquel.vargas@epn.edu.ec"),
fileInput(inputId = "arbol",
  label = "Subir archivo Newick/Nexus",
 accept = c(".nex", ".tree", ".nwk",
 ".nex.con.tre")),
tags$img(style = "display: block;margin: auto;width: 30%;",
 src = "http://www.modemat.epn.edu.ec/images/logo.svg",
   alt = "MODEMAT logo")
                      ),
navbarMenu(title= "Graficar, Modificar árbol",
 tabPanel(title = "Gráfico del árbol",
 sidebarLayout(
 sidebarPanel(
 p(style="color:#DF0174;",strong("Opciones de descarga")),

 radioButtons("formato", "Formato del gráfico",
          c("PDF", "JPEG"), inline = TRUE),
```

```
downloadButton("descargar", "Descargar gráfico"),
radioButtons(inputId = "dir", label = "Dirección del árbol",
 choices = c("Derecha","Izquierda","Arriba","Abajo"),
 inline = TRUE),
radioButtons(inputId = "type", label = "Tipo de árbol",
  choices = c("Filograma","Cladograma","Unrooted",
checkboxInput("largosramas", "Usar largos de las ramas",
value = TRUE),
sliderInput(inputId = "fsize",
label = "Tamaño de la letra",
value = 0.8, min = 0.1, max = 2.1),
radioButtons(inputId = 'tfont', label = "Tipo de letra",
'negrita cursiva'),inline = TRUE),
sliderInput(inputId = "edgew", label = "Grosor de las ramas",
textInput(inputId = "title", label = "Escribir título",

p(style="color:#DF0174;",strong("Etiquetas de nodos y ramas")),

checkboxInput("node.l",
"Etiquetas/Soportes de nodos internos"),
radioButtons("node.frame",
"Marco para etiqueta de nodos",
choices = c("rect", "circle", "none"), inline = TRUE),

sliderInput("adj.node.v", "Ajuste vertical de etiqueta", min = -2,
value = 0.5, step = 0.01),
checkboxInput("edge.l", "Etiquetas/Largos de ramas"),
radioButtons("edge.frame", "Marco para etiqueta de ramas",
"Ajuste vertical de etiqueta", min = -2, max = 2,
),
mainPanel(imageOutput("treeplot"))
    )
 ),
tabPanel(title = "Modificar",
#Texto de ayuda
p(style="font-family:Helvetica; color:blue; font-size:x-large;",
 "Selecione los taxones para eliminar del árbol.
 En la opción 'Gráfico del árbol'
 aparecerá la filogenia sin los taxones seleccionados."),

 tabPanel(title ="Eliminar taxones", uiOutput("tipLabels"))
    )
 ),

   #Ventana de detalles
   navbarMenu(title = "Detalles del árbol",
   tabPanel(title = "Matriz de ramas del árbol",
```

```
    h4("Matriz de ramas del árbol"),
    tableOutput("edges")
     ),
    tabPanel(title = "Número de nodos internos",
        h4("Número de nodos internos"),
    textOutput("Nnodes")
     ),
    tabPanel(title = "Nombres de las especies/taxa",
        h4("Nombres de las especies/taxa"),
    tableOutput("tip.label")),
    tabPanel(title = "Largos de las ramas",
        h4("Largos de las ramas"),
    tableOutput("edge.length")),
    tabPanel(title = "Etiquetas de los nodos internos",
         h4("Nombres/Etiquetas de los nodos internos"),
    tableOutput("node.label"))
             )

))
```

B.2 Server Function

```
library(shiny)
library(ape)
library(grDevices)

shinyServer(
function(input, output) {


    tree.file <- reactive({
      infile <- input$arbol
      if (is.null(infile)) {
        return(NULL)
      }
        if (grepl(".nex", infile$name)==FALSE) {
          read.tree(file = infile$datapath)
        }
      else{
        read.nexus(file = infile$datapath)
      }
      })

    datos <- reactive({
      if(class(tree.file())=="multiPhylo"){
        tree.file()$con_50_majrule
      }
```

```
  else
    tree.file()
})

#Para seleccionar las especies
get.tip.labels <- function(){
  if(is.null(datos())){return(NULL)}
  nombres <- datos()$tip.label
  return(nombres)
}
output$tipLabels <- renderUI({
  nombres.tax <- get.tip.labels()
  checkboxGroupInput("remove.tips", "Especies/Taxa",
  nombres.tax, inline = FALSE)
})

#Para poner o quitar las etiquetas de nodos o ramas
etiquetas <- function(){
  if(input$node.l==TRUE && input$edge.l==TRUE)
  {
    if(is.null(datos()$edge.length))
    {edgelabels( adj = c(input$adj.edge.h,input$adj.edge.v),
    frame = input$edge.frame)
    }
    else
    edgelabels(text = round(datos()$edge.length, digits = 3),
    adj = c(input$adj.edge.h,input$adj.edge.v),
    frame = input$edge.frame)

    if(is.null(datos()$node.label))
  {nodelabels(adj = c(input$adj.node.h,input$adj.node.v),
  frame = input$node.frame)}
    else {
      nodelabels(text = datos()$node.label,
      adj = c(input$adj.node.h,input$adj.node.v),
      frame = input$node.frame)}
  }

  else {
    if(input$node.l==TRUE)
      {
        if(is.null(datos()$node.label))
          nodelabels(adj = c(input$adj.node.h,
          input$adj.node.v),
          frame = input$node.frame)
        else
          nodelabels(text = datos()$node.label,
```

```
            adj = c(input$adj.node.h,input$adj.node.v),
            frame = input$node.frame)
        }
      if(input$edge.l==TRUE)
        if(is.null(datos()$edge.length))
        {edgelabels( adj = c(input$adj.edge.h,input$adj.edge.v),
        frame = input$edge.frame)
        }
      else
        edgelabels(text = round(datos()$edge.length, digits = 3),
        adj = c(input$adj.edge.h,input$adj.edge.v),
        frame = input$edge.frame)
      }
  }

  #Modificar árbol
  tree.mod <- reactive({
      drop.tip(datos(), input$remove.tips)
  })

#Gráfico del árbol
  output$treeplot <- renderImage({
    if(length(datos()$tip.label)<=50){w.px <- 500; h.px <- 980}
    if(length(datos()$tip.label)>50){w.px <- 980; h.px <- 1200}

    outfile <- tempfile(fileext='.png')
    png(outfile, width=w.px, height = h.px)
    plot(tree.mod(), main = input$title,
         cex = input$fsize, edge.width = input$edgew,
         font = switch(input$tfont, 'normal'=1, 'negrita'=2,
         'cursiva'=3, 'negrita cursiva'=4),
         direction = switch (input$dir, "Derecha"="r",
         "Izquierda"="l", "Arriba"="u", "Abajo"="d"),
         type = switch (input$type, "Filograma"="p",
         "Cladograma"="c","Unrooted"="u",
         "Radial"="r", "Radial 'fan'"="f"),
         use.edge.length = input$largosramas)
    etiquetas()
    dev.off()

    list(src = outfile,
         contentType = 'image/png',
         width = w.px,
         height = h.px,
         alt = "Error, no se puede reproducir la imagen")
  }, deleteFile = TRUE)
```

```
fname <- reactive({
  nombre <- paste('filogenia',sep = ".",
  switch (input$formato, PDF = "pdf", JPEG="jpeg"))
  return(nombre)
})

#Para descargar el gráfico
output$descargar <- downloadHandler(
  filename = fname,
  content =  function(archivo){
    owd <- setwd(tempdir())
    on.exit(setwd(owd))

    #Para las dimensiones
    w <- 10; h <- 20

    if(input$formato=='PDF')
    pdf(file = archivo, width = w, height = h)
    else
      jpeg(filename = archivo)

    plot(tree.mod(), main = input$title,
         cex = input$fsize, edge.width = input$edgew,
         font = switch(input$tfont, 'normal'=1, 'negrita'=2,
         'cursiva'=3, 'negrita cursiva'=4),
         direction = switch (input$dir, "Derecha"="r",
         Izquierda"="l", "Arriba"="u",
         "Abajo"="d"),
         type = switch (input$type, "Filograma"="p",
         "Cladograma"="c","Unrooted"="u",
         "Radial"="r", "Radial 'fan'"="f"),
         use.edge.length = input$largosramas)
    etiquetas()
    dev.off()
  }
)

#Detalles del árbol
output$edges <- renderTable({
  m.ramas <- data.frame(datos()$edge)
  colnames(m.ramas) <- c("Ancestro", "Descendiente")
  m.ramas
})
 output$Nnodes <- renderText(datos()$Nnode)

output$tip.label <- renderTable({
```

```
    data.frame(Taxa=datos()$tip.label)
  })

  output$edge.length <- renderTable(
    data.frame(LargoRamas=datos()$edge.length))

  output$node.label <- renderTable({
    data.frame(Etiquetas.Nodos=datos()$node.label)
  })
})
```

C Árboles utilizados para la simulación

C.1 Árbol de 5 taxones

```
(((Esp1:0.2767745421,Esp2:0.2767745421)
:0.3682187609,(Esp3:0.1543530229,
Esp4:0.1543530229):0.4906402801):0.4068158017,
Esp5:1.051809105);
```

C.2 Árbol de 12 taxones

```
(Esp1:0.1874619216,Esp2:0.1415238567,(((((Esp3:0.04083097,
Esp4:0.0535701986)74:0.0186475642,Esp5:0.0551866504)
100:0.0501977619,Esp6:0.0901109406)97:0.0294390552,
Esp7:0.1076220863)100:0.0573132911,
(((Esp8:0.0156938025,Esp9:0.0208285911)99:0.0294991163,
Esp10:0.0479320065)97:0.0293973899,Esp11:0.059718508)
100:0.1172002889)96:0.0491263137,Esp12:0.1870942635)
100:0.0958952321);
```

C.3 Árbol de 20 taxones

```
((((Esp1:0.4745814383,Esp2:0.4745814383):0.2657732357,
(Esp3:0.1127902263,(Esp4:0.08197239806,(Esp5:0.03877361002,
Esp6:0.03877361002):0.04319878804):0.0308178282):0.6275644478)
:0.09110654227,((((Esp7:0.09731731053,(Esp8:0.05177796147,
(Esp9:0.004858303452,Esp10:0.004858303452):0.04691965802):
0.04553934906):0.1970761599,((Esp11:0.06631926325,
Esp12:0.06631926325):0.1098622721,Esp13:0.1761815353)
:0.1182119351):0.2248615831,(Esp14:0.03671757555,
Esp15:0.03671757555):0.482537478):0.2530718227,
(((Esp16:0.02787680828,Esp17:0.02787680828):0.02653497435,
Esp18:0.05441178263):0.009713991678,Esp19:0.0641257743)
:0.7082011019):0.05913434007):0.78725396,Esp20:1.618715176);
```

C.4 Árbol de 30 taxones

```
(((Esp1:0.1399630802,Esp2:0.1399630802):0.03176639743,
Esp3:0.1717294776):0.9572007808,(((Esp4:0.03777564588,
Esp5:0.03777564588):0.4154038938,(((Esp6:0.01003981613,
Esp7:0.01003981613):0.02815875034,Esp8:0.03819856648)
:0.2622123016,(((Esp9:0.01381010978,Esp10:0.01381010978)
:0.03236759718,Esp11:0.04617770696):0.1407124613,Esp12
:0.1868901683):0.1135206998):0.1527686716):0.3807530829,
((Esp13:0.0334587832,Esp14:0.0334587832)
:0.5976542791,(((Esp15:0.1369992106,(Esp16:0.0293369574,
Esp17:0.0293369574):0.1076622532):0.05692743639,
Esp18:0.193926647):0.2613551342,((Esp19:0.09960164318,
((Esp20:0.06517793557,
(Esp21:0.008290120494,Esp22:0.008290120494):0.05688781508)
:0.02539282503,(Esp23:0.028046858,((Esp24:0.003667273303,
Esp25:0.003667273303):0.01025751023,Esp26:0.01392478353)
:0.01412207447):0.0625239026):0.009030882579):0.08166649654,
((Esp27:0.004787450124,Esp28:0.004787450124):0.1359987046,
(Esp29:0.07755279389,Esp30:0.07755279389):0.06323336082)
:0.04048198501):0.2740136414):0.1758312812):0.2028195603)
:0.2949976358);
```

C.5 Árbol de 50 taxones

```
((Esp1:0.9366434721,((((((Esp2:0.004882765841,
(Esp3:0.001836560233,Esp4:0.001836560233):0.003046205608)
:0.04997090215,(Esp5:0.007946961466,Esp6:0.007946961466)
:0.04690670652):0.06705953444,
((Esp7:0.0212383486,Esp8:0.0212383486):0.009918700402,
Esp9:0.03115704901):0.09075615342):0.1651178778,
((Esp10:0.162031337,(Esp11:0.1384426766,((Esp12:0.08709613182,
(Esp13:0.08645268998,Esp14:0.08645268998):0.0006434418484)
:0.02990923063,(Esp15:0.1069924107,(Esp16:0.003402187836,
Esp17:0.003402187836):0.1035902229):0.01001295172)
:0.02143731418):0.02358866034):0.07866578158,
((Esp18:0.01228839366,Esp19:0.01228839366)
:0.08977716314,((Esp20:0.06920151437,(Esp21:0.00213362092,
Esp22:0.00213362092):0.06706789345):0.02189209821,
Esp23:0.09109361258):0.01097194422):0.1386315618):0.04633396171)
:0.01173354969,(((Esp24:0.0781194656,((Esp25:0.02318403769,
Esp26:0.02318403769):0.05230580689,Esp27:0.07548984457)
:0.002629621032):0.0695182389,(Esp28:0.06597341607,
(Esp29:0.05144582059,Esp30:0.05144582059):0.01452759548)
:0.08166428843):0.01150461572,Esp31:0.1591423202):0.1396223097)
:0.6378788422):0.04646696642,(((Esp32:0.04054040381,
```

```
Esp33:0.04054040381):0.07039695599,((Esp34:0.06857357654,
Esp35:0.06857357654):0.008037447316,(Esp36:0.05345715835,
(Esp37:0.04637192798,(Esp38:0.0411931923,((Esp39:0.02065076431,
((Esp40:0.01416095365,Esp41:0.01416095365):0.003680943033,
(Esp42:0.0152971009,Esp43:0.0152971009):0.002544795783)
:0.002808867624):0.006243268167,Esp44:0.02689403247)
:0.01429915983):0.005178735681):0.007085230369):0.0231538655)
:0.03432633595):0.4378915969,(((Esp45:0.03352538199,
Esp46:0.03352538199):0.002825115122,Esp47:0.03635049711)
:0.07941479518,(Esp48:0.0541651468,(Esp49:0.02592506894,
Esp50:0.02592506894):0.02824007787):0.06160014548):0.4330636645)
:0.4342814818);
```

Referencias

[1] X. GU and W. Li. "Bias Corrected Paralinear and Log-Det Distances and Test of Molecular Clocks and Phylogenies Under Nonstationary Nucleotide Frequencies". In: *Human Genetics Center* (1996), pp. 1375–1383.

[2] Simon Y. W. Ho. "Molecular Clocks". In: *Encyclopedia of Scientific Dating Methods*. Ed. by W. Jack Rink and Jeroen Thompson. Dordrecht: Springer Netherlands, 2013, pp. 1–9. ISBN: 978-94-007-6326-5. DOI: 10.1007/978-94-007-6326-5_92-2. URL: https://doi.org/10.1007/978-94-007-6326-5_92-2.

[3] E. S. ALLMAN and J. RHODES. *Lectures Notes: The Mathematics of Phylogenetics*. University of Alaska Fairbanks, 2012.

[4] Winston CHANG et al. *shiny: Web Application Framework for R*. R package version 1.0.5. 2017. URL: https://CRAN.R-project.org/package=shiny.

[5] J. FELSENSTEIN. *Inferring Phylogenies*. Sunderland, USA.: Sinauer Associates Inc., 2004.

[6] Vincent GOULET et al. *expm: Matrix Exponential, Log, 'etc'*. R package version 0.999-2. 2017. URL: https://CRAN.R-project.org/package=expm.

[7] M. HASEGAWA, H. KISHINO, and T. YANO. "Dating of the human-ape splitting by a molecular clock of mitochondrial DNA". In: *Journal of Molecular Evolution* 22 (1985), pp. 160–174.

[8] S. KALYAANAMOORTHY et al. "ModelFinder: fast model selection for accurate phylogenetic estimates". In: *Nature Methods* 14 (May 2017), 587–589. DOI: 10.1038/nmeth.4285. URL: +http://dx.doi.org/10.1038/nmeth.4285.

[9] M. KIMURA. "A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences". In: *Journal of Molecular Evolution* (1980).

[10] K. KNIGHT. *Mathematical Statistics*. Chapman & Hall/CRC, 2000.

[11] S. KONISHI and G. KITAGAWA. *Information Criteria and Statistical Modeling*. New York, USA.: Springer-Verlag, 2008.

[12] M. KUHNER and J. FELSENSTEIN. "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates." In: *Mol Biol Evol* 3 (1994), pp. 459–468.

[13] G. F. LAWLER. *Introduction to Stochastic Processes*. Second Edition. Chapman & Hall/CRC, 2006.

[14] MCCLELLAN, *B. Using trees to make predictions about poorly-studied species: A new drug*. Recuperado de https://evolution.berkeley.edu/evolibrary/article/0_0_0/phylogenetics_12.

[15] L. T. NGUYEN et al. "IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies." In: *Mol. Biol. Evol.* 32 (2015), pp. 268–274. DOI: 10.1093/molbev/msu300. URL: +https://doi.org/10.1093/molbev/msu300.

[16] E. PARADIS, J. CLAUDE, and K. STRIMMER. "APE: analyses of phylogenetics and evolution in R language". In: *Bioinformatics* 20 (2004), pp. 289–290.

[17] S. M. ROSS. *Introduction to Probability Models*. Tenth Edition. Los Angeles, USA.: Academic Press, 2010.

[18] Mohammed A. S. SALMAN and V. C. BORKAR. "Exponential Matrix and Their Properties". In: *International Journal of Scientific and Innovative Mathematical Research (IJSIMR)* 4 (2016), pp. 53–62.

[19] H. A. SCHMIDT and A. VON HAESELER. "Phylogenetic inference using maximum likelihood methods". In: *The Phylogenetic Handbook*. Ed. by LEMEY P., Salemi M., and Vandamme A. M. Cambridge University Press, 2009, pp. 181–198.

[20] K. STRIMMER and A. VON HAESELER. "Genetic Distances and Nucleotide Sustitution Models". In: *The Phylogenetic Handbook*. Ed. by LEMEY P., Salemi M., and Vandamme A. M. Cambridge University Press, 2009, pp. 111–125.

[21] G. TAKAHARA. "Continuous-Time Markov chains". [archivo PDF] Recuperado de http://www.mast.queensu.ca/~stat455/lecturenotes/set5.pdf. 2017.

[22] E. WAHLÉN. "The matrix exponential". [archivo PDF] Recuperado de http://www.ctr.maths.lu.se/media/MATC12/2014ht2014/exp_6.pdf. 2014.

[23] L. WASSERMAN. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010.

[24] Z. YANG. "Molecular Clock". In: *Oxford Encyclopedia of Evolution*. Ed. by M. PAGEL. Oxford University Press, 2002, pp. 747–748. URL: http://abacus.gene.ucl.ac.uk/ziheng/pdf/2002YangOEEp747.pdf.

[25] *The Newick tree format*. Recuperado de http://evolution.genetics.washington.edu/phylip/newicktree.html. s.f.