Elementos matemáticos en la construcción de árboles filogenéticos

Ignacio Morales Rodríguez

Trabajo de Fin del Grado de Matemáticas

Universidad de Zaragoza

Prólogo

Desde sus orígenes, el ser humano se ha preguntado sobre su procedencia, pero es Charles Darwin en 1850 quien con su libro *El origen de las especies* plantea que toda especia viva se ha originado a partir de otra anterior y, dado que todos los seres vivos compartimos el mismo código genético, no es extraño intentar agrupar a las diferentes especies por su origen y desarrollo evolutivo, es decir, mediante su filogenia, término acuñado por el naturalista y filósofo alemán Ernst Haeckel (1834-1919) el cual fue un ferviente evolucionista que popularizó el trabajo de Charles Darwin creando nuevos términos como "phylum" y "ecología".

La filogenética[W] se ocupa de determinar la filogenia, y consiste en el estudio de las relaciones evolutivas entre los diferentes grupos de organismos utilizando matrices de información de ADN y de morfología. Con esta información se construyen los árboles filogenéticos, y aunque las teorías de reconstrucción filogenética son ideas tardías en biología, actualmente es un aspecto muy importante de la sistemática, el área de la biología encargada de clasificar a las especies a partir de su historia evolutiva, y forman parte de su núcleo central.

Como he mencionado, los árboles han sido utilizados en biología para representar las relaciones evolutivas entre especies y genes. Un "árbol con raíz" por definición desciende en dos direcciones desde un único nodo llamado raíz y se bifurca en otros dos nodos llamados comunmente hojas. Estas hojas están etiquetadas con el nombre de las especies y, aquellas hojas situadas en los extremos del árbol se denominan taxones. Eliminando la raíz del árbol y juntando sus dos ramas que descienden de esta raíz, da lugar a lo que llamamos árbol sin raíz (ver figura 1).

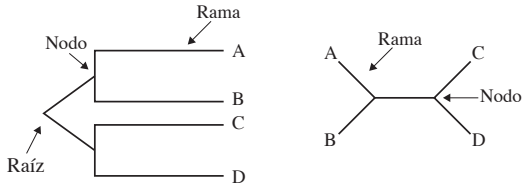

Figura 1: Árbol con raíz (izquierda) y correspondiente árbol sin raíz (derecha).

La longitud de cada rama es normalmente un número positivo que representa el grado de relación entre las especies situadas en sus extremos pero, si no tenemos en cuenta esta longitud y dado un número de taxones fijo asignado a las hojas, enseguida nos damos cuenta que podemos obtener múltiples diseños de árboles o mejor dicho, diferentes topologías del árbol, sin embargo, evolutivamente sólo puede haber una secuencia de eventos que haya conducido a la formación de estas especies y por ende, al árbol.

En la actualidad, la reconstrucción filogenética del árbol dada la cadena de ADN es el final de un proceso en el cual no sólo se ha elegido una familia de secuencias homólogas de especies con una relación filogenética suficientemente fuerte sino que, previamente también se ha realizado un proceso de "alineado" mediante algorítmica. Nosotros nos vamos a centrar, dada una colección de secuencias de ADN ya optimizadas, en inferir, mediante distintas técnicas, el mejor árbol filogenético y modelo evolutivo asociado a él.

Summary

We all know that trees have been used in biology to represent the evolutionary relationships among species and genes. As this phylogenetic representation has been highly developed in the last decades, nowadays it has a crucial importance in bioinformatics. This summary will aim at giving an overall view of the following chapters that deals with the mathematical elements in these phylogenetic trees.

We start defining a distance for our tree T so, a **metric tree** (T,w) is a rooted or unrooted tree T together with a function $w : E(T) \longrightarrow \mathbb{R}$ assigning non-negative numbers to edges. This leads to a way of measuring distances between vertices. For any $v_i, v_j \in V(T)$ we define the *tree-generated distance function* as:

$$d(v_i, v_j) = d_{ij} = \sum_{e \in \text{path}(v_i, v_j)} w(e).$$

The main objective is to ensure the uniqueness of the tree given a set of distances between species called *disimilarities*. Two kind of distances have an important role in that task: the *additive distance* and the *ultrametric distance*:

From one hand, we say that a distance $d(\cdot,\cdot)$ is **additive** if it verifies:

$$d_{ij} + d_{kl} \leq \text{máx}\left(d_{ik} + d_{jl}, d_{il} + d_{jk}\right) \text{ for all } i,j,k,l \in V,$$

on the other hand, a distance d is **ultrametric** if it verifies:

$$d_{ij} \leq \text{máx}\left(d_{ik}, d_{jk}\right) \textit{ for all } i,k,j \in V,$$

both under some assumptions lead to the unique tree. After the existence and uniqueness of the tree we must focus on the reconstruction. There are two important algorithms to reconstruct the phylogenetic tree given a set of taxa: the ***UPGMA*** that is a fast clustering algorithm that works properly under ultrametric distances and the ***Neighbour-Joining*** that is a more general algorithm and most commonly used in bioinformatics. Also, we introduce another method of tree reconstruction: the least squares algorithm (Fitch-Margoliash).

So far, those reconstruction methods do not take into account the mutation process of the DNA. Firstly we think of this mutation as an inusual evolutionary event. That is the idea behind the parsimony criterion that measures the level of compatibility of data. This overall parsimony score is calculated as the sum of the parsimony score of every site of the DNA sequence $\{\chi_1, ..., \chi_m\}$:

$$ps_\chi(T) = \min_{\widetilde{\chi} \in Ext_T(\chi)} c(\widetilde{\chi}, T),$$

$$ps_{\{\chi_1,...,\chi_m\}}(T) = \sum_{i=1}^{m} ps_{\chi_i}(T).$$

In practice, this parsimony score is calculated by the *Fitch-Hartigan algorithm* and it can be considered as a method for selecting a good starting tree topology. However parsimony has several problems such us the named *long branch attraction* that may lead to a wrong tree.

The parsimony method is a very simple approach because it does not pay attention to multiple substitutions that may take place in the mutation process between one nucleotide and another. In order to consider those changes and above all, to be biologically consistent, we must introduce evolutionary models through Markov chains. The first model used in phylogenetics was the Jukes-Cantor Model, with the following transition Markov matrix:

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 1-a & a/3 & a/3 & a/3 \\ a/3 & 1-a & a/3 & a/3 \\ a/3 & a/3 & 1-a & a/3 \\ a/3 & a/3 & a/3 & 1-a \end{pmatrix} \end{array}, \tag{1}$$

where each site of the matrix represents the probability of substitution between characters (nucleotides A,C,G,T) and a is the probability of given two sequences, of having different characters in a site (assuming its independence). When we have an estimator $\widehat{a}$ of a, we can introduce a model based tree distance:

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

But the Jukes-Cantor model assumes equal base frequencies and equal mutation rates. Because of that, more complex models have been developed in order to consider more realistic cases, such as different mutation rates (Felsenstein model) that are displayed on Chapter 2. What is more, this method presupposes not only the independence between sites of the sequence but also that all the nucleotides mutate at the same speed. That is not biologically correct so we must modify the model to include parameters such as r, indicating that with probability r a site is variable or λ_i, which are factors to speed up or slow down the substitution process.

Finally, As we have a wide range of models that can fit our data, a way to compare different evolutionary models is needed. Then, we introduce maximum likelihood, for every tree T with n taxa. The likelihood function is:

$$\ln L_T = \sum_{(i_1,\ldots,i_n)\in\{A,G,C,T\}^n} n(i_1,\ldots,i_n)\ln(p(i_1,\ldots,i_n)),$$

where

$$p(i_1,...,i_n) = \sum_{k=1}^{4}\sum_{j=1}^{4}\ldots\sum_{s=1}^{4}\pi_k P_1(k,i_1)P_2(k,j)P_3(j,i_2)\ldots P_r(s,i_n).$$

In closing, we apply in chapter 3 all the ideas to two different set of data ussing R software. The first consists of 14 DNA sequences about mammals species, we want to reconstruct the tree that fits best to them and also find through statistical tools (maximum likelihood, bootstrapping, ...) the best evolutionary model that fits our data. The other data set consists of 13 birds geospiza genus but the main difference beeween this set and the other is that in this one we have phenotypic traits instead of DNA sequences. These two examples are intended to illustrate the process carried out to obtain and select a phylogenetic tree. Furthermore, they depict how mathematical models and tools evolve in order to suit the needs of the phylogenetics.

Índice general

Capítulo 1

Árboles filogenéticos a partir de distancias

En este primer capítulo introducimos las definiciones asociadas a árboles filogenéticos y el número posible de éstos. Consideramos la noción de distancia entre nodos y los algoritmos elementales de reconstrucción de un árbol dada la matriz de distancias entre las hojas.

1.1. Definición de árbol y propiedades.

Definición 1.1.1. *Un **árbol** $T = (V,E)$ es un grafo conexo y sin ciclos donde donde $V = V(T)$ es un conjunto de vértices o nodos, y $E = E(T)$ es un conjunto de aristas. Cada eje $e \in E$ es un conjunto de dos elementos $e = \{v_1, v_2\}$ de vértices $v_1, v_2 \in V$.*

Definición 1.1.2. *Un **árbol binario** es un árbol en el que cada vértice interior tiene grado tres.*

Es biológicamente concebible trabajar con árboles no binarios, los nodos de los árboles no binarios pueden ser modelizados por una sucesión de bifurcaciones con periodos pequeños de tiempo. Por ello trabajaremos únicamente con árboles binarios.

Cabe también señalar que la raíz de un árbol no la consideramos como vértice interior. Además, los árboles utilizados en filogenética tienen una característica fundamental: mientras las hojas representan taxones conocidos y usados para inferir el árbol, los nodos internos representan taxones de los cuales no tenemos información directa. Incluso si tenemos información de especies más antiguas, no podemos asumir que éstas sean ancestros directos de las existentes, es más probable que sean vástagos de linajes que conducen hacia las especies actuales. Por ello etiquetamos los taxones de la siquiente manera:

Definición 1.1.3. *Sea X un conjunto finito de taxones. Entonces un **árbol filogenético** (phylogenetic X-tree) es un árbol T con una correspondencia biyectiva $\phi : X \longrightarrow L$, donde $L \subset V$ denota el conjunto de hojas del árbol.*

1.1.1. Combinatoria básica en árboles.

Teorema 1.1.4. *Un árbol sin raíz con $n \geq 2$ hojas tiene $2n-2$ vértices y $2n-3$ aristas .*

Demostración. Basta prodecer por inducción, para $n = 2$ no cabe duda que se cumple, luego, suponiendo que es cierto para $n-1$, suponer que T tiene $n \geq 3$ hojas. Si v_1 es una de las hojas de T, entonces estará en un único eje $\{v_1, v_2\}$ donde v_2 al ser un nodo interno estará situado en los ejes $\{v_2, v_3\}$ y $\{v_2, v_4\}$. Eliminando estas tres aristas y los dos nodos v_1 y v_2 de T, e introduciendo una nueva arista $\{v_3, v_4\}$ tenemos un árbol binario con $n-1$ hojas. Aplicamos la hipótesis de inducción, tendremos: $(2(n-1)-2)+2 = 2n-2$ vértices y $(2(n-1)-3)+2 = 2n-3$ aristas. □

Teorema 1.1.5. *Sea X un conjunto finito de $n \geq 3$ taxones, entonces:*

- *El número de posibles árboles sin raíz distintos es:*

$$b(n) = (2n-5)!! = \frac{(2n-5)!}{2^{n-3}(n-3)!} \tag{1.1}$$

- *El número de posibles árboles con raíz distintos es:*

$$b(n+1) = (2n-3)!! = \frac{(2n-3)!}{2^{n-2}(n-2)!} \tag{1.2}$$

Demostración. Para ver esta igualdad basta proceder por inducción. Si $n = 3$ hay sólo hay un posible árbol filogenético sin raíz: todas las hojas unidas mediante un vértice interior de grado tres. Suponiendo cierto para $n-1$ procedemos con el mismo razonamiento que antes, eliminando v_1 y ajustando los ejes apropiadamente tenemos el árbol T'. Para cada v_1 fijo, la aplicación $T \longrightarrow (T', \{v_3, v_4\})$ es una biyección de árboles de n hojas a pares de árboles de $n-1$ hojas y ejes. En dicho par, pensamos en el eje $\{v_3, v_4\}$ como el eje en el cual injertar un nuevo eje con v_1 para recuperar T. De manera que contando estos pares tenemos que:

$$b(n) = b(n-1) * (2(n-1)-3) = b(n-1) * (2n-5)$$

Y por la hipótesis de inducción se tiene (1.1). Ahora una vez obtenido este resultado, obtener (1.2) es inmediato ya que éste puede ser pensado como un árbol sin raíz de n hojas en el cual introducimos la raíz en cualquiera de sus posibles $2n-3$ aristas:

$$(2n-3)b(n) = (2n-3)(2n-5)!! = (2n-3)!!$$

□

1.1.2. Distancia.

Definición 1.1.6. *Sea V un conjunto de elementos, diremos que la función $d : V \times V \longrightarrow \mathbb{R}$ es una función **distancia** si verifica las siguientes condiciones:*

(i) $d(u,v) \geq 0$ *para todo u,v* $\in V$, $u \neq v$,

(ii) $d(u,u) = 0$ *para todo* $u \in V$

(iii) $d(u,v) = d(v,u)$ *para todo u,v* $\in V$

(iv) Desigualdad triangular: $d(u,v) \leq d(u,w) + d(w,v)$ *para todo u,v,w* $\in V$.

Definición 1.1.7. *Un **árbol métrico** (T,w) es un árbol (con o sin raíz) con una función $w : E(T) \longrightarrow \mathbb{R}$ asginando números no negativos a las aristas. Llamamos $w(e)$ a la longitud de la arista e.*

La definición anterior nos conduce a una forma de medir distancias entre vértices. Para cualesquiera $v_i, v_j \in V(T)$ definimos:

$$d^{\mathcal{T}}(v_i, v_j) = \sum_{e \in \text{path}(v_i, v_j)} w(e), \tag{1.3}$$

donde path(v_i,v_j) es el único camino desde v_i hasta v_j.

Dado un árbol métrico (T,w), como $d^{\mathscr{T}}$ verifica (1.1.6) tenemos que es una función distancia, denominada *distancia generada por el árbol*. Notar que estamos interesados en el caso en que X es un conjunto finito X=$\{x_1,...,x_n\}$ de nodos etiquetados con el nombre de la especie correspondientes a los hojas. Además $d^{\mathscr{T}}$ tiene que ser "biológicamente relevante" es decir, que proporcione información sobre el grado de separación de éstas, por ejemplo si x_a y x_b se han separado más de su antecesor que x_c y x_d entonces tendríamos que $d^{\mathscr{T}}(x_a,x_b) > d^{\mathscr{T}}(x_c,x_d)$.

Para facilitar la notación, llamamos $d^{\mathscr{T}}$ simplemente d y pensamos en ésta como una matriz simétrica donde sus elementos son de la forma $d(v_i,v_j) = d_{ij}$ para todo v_i, $v_j \in V$. En la práctica, se nos da una matriz de distancias (δ_{ij}) calculada mediante algún procedimiento (en el que no entramos ahora, pero se verá en el capítulo 3 en el caso de secuencias de ADN) y nuestro objetivo será intentar conseguir un árbol tal que la distancia que genere d, esté lo más próxima posible a la distancia δ original. Formalizando esta idea, podemos definir la distancia inicial δ entre taxones como:

Definición 1.1.8. *Una **disimilaridad** para un conjunto X de taxones es una función $\delta : X \times X \longrightarrow \mathbb{R}$ tal que $\delta(x,x) = 0$, $\delta(x,y) = \delta(y,x)$ y $\delta(x,y) \geq 0$ para todo $x,y \in X$.*

Problema principal: Dada una disimilaridad δ, esperamos encontrar un árbol métrico (T,w) con función distancia d tal que la disimilaridad δ sea la restricción de d a $X \times X$.

Sin embargo, hasta ahora no hay garantía de que δ sea dicha restricción. Presentamos una serie de teoremas y propiedades que aseguran no sólo la unicidad de árbol, sino también la coincidencia de ambas funciones.

Definición 1.1.9. *Una distancia $d(\cdot,\cdot)$ en un conjunto de vértices se dice que es **aditiva** si satisface:*

$$d_{ij} + d_{kl} \leq \max\left(d_{ik} + d_{jl}, d_{il} + d_{jk}\right) \text{ para todo } i,j,k,l \in V \tag{1.4}$$

Proposición 1.1.10. *(Condición de los cuatro puntos)*
Sea d una función distancia en V y $n \geq 4$. Entonces d es aditiva si y sólo si para cualesquiera $x,y,u,v \in V$ se verifica:

$$d_{xy} + d_{uv} \leq d_{xu} + d_{yv} = d_{xv} + d_{yu} \tag{1.5}$$

Veamos que para $n = 3$ hojas existe dicho árbol con métrica aditiva (el caso $n = 2$, es obvio):
Sea X=$\{x_1, x_2, x_3\}$, buscamos x,y,z positivos tal que:

$$\begin{aligned} x + y &= \delta_{12}, \\ x + z &= \delta_{13}, \\ y + z &= \delta_{23} \end{aligned} \tag{1.6}$$

La solución al anterior sitema de ecuaciones es:

$$\begin{aligned} x &= \frac{1}{2}(\delta_{12} + \delta_{13} - \delta_{23}), \\ y &= \frac{1}{2}(\delta_{12} + \delta_{23} - \delta_{13}), \\ z &= \frac{1}{2}(\delta_{13} + \delta_{23} - \delta_{12}) \end{aligned} \tag{1.7}$$

Observar que si δ no verifica la desigualdad triangular, x,y,z podrían tener valor nulo o negativo. Estas ramas de longitud cero en Biología se interpretan como ramas de longitud muy pequeña en comparación con las demás. Sin embargo para $n \geq 4$ no toda función distancia es aditiva, como se puede ver en el ejemplo 5.6 de [I04].

Vamos a introducir un tipo especial de distancia más restrictiva que esta última, la *distancia ultramétrica*:

Definición 1.1.11. *Una distancia d en un conjunto V de vértices diremos que es **ultramétrica** si satisface:*

$$d_{ij} \leq \max\left(d_{ik}, d_{jk}\right) \text{ para todo } i,k,j \in V \tag{1.8}$$

Proposición 1.1.12. *(Condición de los tres puntos)*
Sea d una función distancia en V y $n \geq 3$. Entonces d es ultramétrica si y sólo si para cualesquiera $x,y,z \in V$ se verifica:

$$d_{xy} \leq d_{xz} = d_{yz} \tag{1.9}$$

La demostración de la siguiente proposición se puede encontrar en [EG01] y asegura, bajo esta nueva distancia, la unicidad del árbol:

Teorema 1.1.13. *Sea d una distancia ultramétrica entre especies, entonces existe un único árbol con raíz agrupándolas.*

La condición de los cuatro puntos es necesaria y suficiente para la aditividad. Como la distancia ultramétrica la cumple, nos confirma que es una distancia aditiva. Esta condición de los cuatro puntos va a garantizar, bajo ciertas condiciones la unicidad a la hora de reconstruir el árbol para una distancia no ultramétrica. Para ilustrar esta idea, planteamos una serie de proposiciones y un teorema cuyas demostraciones las podemos encontrar detalladamente en [AR12] y que hacen uso de las siguientes definiciones:

Definición 1.1.14. *Una **división** de X es cualquier partición de X en dos subconjuntos no vacíos. Escribimos $X_0|X_1$ para denotar la división entre los dos subconjuntos X_0, X_1.*

Sea T es un árbol filogenético, $e \in E(T)$, entonces la división $X_0|X_1$ es la inducida por e mediante la eliminación de dicho eje y formando los subconjuntos conexos X_0, X_1.

Definición 1.1.15. *Un **cuarteto** es un árbol sin raíz con cuatro hojas etiquetadas con el nombre de la especie. Denotamos el árbol como $ab|cd$ si éste induce la división $\{a,b\}|\{c,d\}$.*

Cualquier árbol filogenético con X el conjunto finito de taxones, induce una colección $Q(T)$ de cuartetos:

$$Q(T) = \{ab|cd : \text{ para algún } X_0|X_1 \text{ con } e \in E(T), a,b \in X_0, \text{ y } c,d \in X_1\}$$

Proposición 1.1.16. *Un cuarteto métrico relacionando los taxones a,b,c,d con longitudes de arista positivas, tiene como vecinos a,b y c,d si y sólo si, cualquiera de las tres desigualdades / igualdades se verifica:*

$$d(a,b) + d(c,d) < d(a,c) + d(b,d) = d(a,d) + d(b,c).$$

Definición 1.1.17. *Una disimilaridad δ en X diremos que satisface la **condición de los cuatro puntos** si para cada elección de nuestros cuatro taxones $x,y,z,w \in X$.*

$$\delta(x,y) + \delta(z,w) \leq \max\{\delta(x,z) + \delta(y,w), \delta(x,w) + \delta(y,z)\}. \tag{1.10}$$

Proposición 1.1.18. *Dado cualquier árbol métrico con longitudes de arista positivas, la métrica de dicho árbol es una disimilaridad que satisface la condición de los cuatro puntos.*

La unicidad que buscábamos nos viene dada por el siguiente teorema:

Teorema 1.1.19. *Sea X un conjunto finito de taxones, $|X| \geq 3$, y δ es una disimilaridad en X con $\delta(x,y) \neq 0$ para cualquier $x \neq y$. Entonces si δ satisface la condición de los cuatro puntos, hay un único árbol métrico cuyo conjunto de taxones es X con longitudes de arista positivas y cuya métrica coincide con δ sobre X.*

Demostración.

La demostración completa de este teorema la podemos encontrar en el libro [AR12]. Ésta procede por inducción introduciendo el concepto de *generalized cherry* en un árbol para n taxones.

□

Vamos a ver los dos algoritmos más importantes a la hora de reconstruir árboles filogenéticos. Por un lado, el algoritmo UPGMA es adecuado cuando la distancia es ultramétrica, mientras que el algoritmo NJ no requiere tanta restricción y, basado en la condición de los cuatro puntos nos da un algoritmo consistente pero más complejo.

1.2. Algoritmo UPGMA

Dada una distancia ultramétrica el algoritmo *Unweighted Pair Group Method Using Arithmetic Averages (UPGMA)* nos recupera el árbol correspondiente. Es importante recalcar que este algoritmo crea un árbol con raíz en el que sitúa siempre a todos los taxones a la misma distancia de ella. Un árbol presentando la característica anterior lo denominamos *árbol molecular*. En términos biológicos, podemos decir que asume la existencia de un reloj molecular evolutivo.

Dada una matriz distancia (d_{ij}), el algoritmo empieza agrupando los dos taxones con la mínima distancia d_{ij}. Añadimos un nuevo nodo en el punto medio de ellos, y estos dos taxones se colocan en el árbol. La distancia del nuevo nodo a otros nodos será la media aritmética. Entonces obtenemos una matriz distancia reducida reemplazando los dos taxones por este nuevo nodo. Repetimos este proceso hasta que todos los taxones se añaden al árbol y el último de ellos será la raíz del árbol.

Escrito en pseudocódigo [JP04]:

UPGMA(d,n)

Formar n grupos, cada uno compuesto por un único elemento
Construir un grafo T asignando a cada hoja un grupo
Asignar altura $h(x_i) = 0$ a cada hoja x_i en este grafo, $i = 1,\ldots,n$
mientras haya más de un grupo:
    Encontrar los dos grupos más cercanos C_1 y C_2
    Fusionar C_1 y C_2 en un nuevo grupo C con $|C_1|+|C_2|$ elementos
    para cada grupo $C^* \neq C$
        $d(C,C^*) = \frac{1}{|C||C^*|}\sum_{i\in C}\sum_{j\in C^*} d_{ij}$
    Añadir un nuevo vértice C a T y conectar a los vértices C_1 y C_2
    $h(C) \leftarrow \frac{d(C_1,C_2)}{2}$
    Asignar longitud $h(C)-h(C_1)$ a la arista (C_1,C)
    Asignar longitud $h(C)-h(C_2)$ a la arista (C_2,C)
    Eliminar filas y columnas de (d_{ij}) correspondientes a C_1 y C_2
    Añadir una fila y columna a (d_{ij}) para el nuevo grupo C
devolver T

Notar que si una distancia satisface la condición de los cuatro puntos pero no es ultramétrica entonces la tasa evolutiva entre linajes puede no ser constante y por tanto al aplicar este algoritmo producirá un árbol equivocado ya que siempre forma un árbol molecular.

1.3. Algoritmo Neighbor-Joining (NJ)

Para empezar, suponer que tenemos un árbol binario con longitudes de rama positivas en el cual los taxones x_1 y x_2 son vecinos mediante el vértice v y unidos de alguna manera al conjunto de taxones restantes $x_3, x_4, ..., x_n$. Dada una matriz de distancias entre taxones δ_{ij} suponemos que ésta es muy próxima a la distancia generada por el árbol $d_{ij} = \delta_{ij}$, luego para todo $i, j = 3, 4, ..., n$ por la condición de los cuatro puntos tenemos que:

$$d_{12} + d_{ij} < d_{1i} + d_{2j}. \tag{1.11}$$

Para un i fijo, hay $n-3$ posibles elecciones de j con $3 \leq j \leq n$ y $j \neq i$, entonces, sumando todas las desigualdades de (1.11) para estos j tenemos que:

$$(n-3)d_{12} + \sum_{j=3 j\neq i}^{n} d_{ij} < (n-3)d_{1i} + \sum_{j=3 j\neq i}^{n} d_{2j}. \tag{1.12}$$

Para simplificar esta expresión, definimos la disimilitud total entre el taxón x_i y los demás x_j como:

$$R_i = \sum_{j=1}^{n} d_{ij} = (n-2)r_i,$$

siendo $r_i = \frac{1}{n-2}\sum_{j=1}^{n} d_{ij}$. Así sumando $d_{i1} + d_{i2} + d_{12}$ a cada lado de la desigualdad (1.12) podemos escribir (1.12) de la siguiente manera:

$$(n-2)d_{12} + R_i < (n-2)d_{1i} + R_2,$$

y restando a esta última expresion $R_1 + R_2 + R_i$ a cada lado de la desigualdad tenemos:

$$(n-2)d_{12} - R_1 - R_2 < (n-2)d_{1i} - R_1 - R_i.$$

Si definimos $D_{1i} = d_{1i} - r_1 - r_i$ con $i = 2, \ldots, n$, hemos visto que $D_{12} < D_{1j}$, $j = 3, \ldots, n,$. Generalizando para x_n y x_m taxones vecinos tenemos:

$$D_{nm} < D_{nk} \text{ para todo } k \neq m,$$

con:

$$D_{ij} = d_{ij} - r_i - r_j. \tag{1.13}$$

Esto nos da el núcleo fundamental del algoritmo NJ: Con los datos, calculamos una tabla con todos los valores D_{ij} usando la ecuación (1.13). Entonces tomamos el par x_i, x_j de taxones con el mínimo valor de D_{ij}. Lo que nos muestra este argumento es que, si x_i y x_j son vecinos en el árbol, entonces el valor de D_{ij} es el valor más pequeño de la tabla D. Sin embargo esto no es suficiente para justificar el algoritmo, se necesita un último teorema para justificar que el menor valor de esta tabla identifica perfectamente al par de hojas vecinas. La demostración de este teorema la podemos encontrar en [AR12]:

Teorema 1.3.1. *Sea X un conjunto finito de taxones y (T, w) el árbol métrico asociado. Suponer que δ es la restricción de d a $X \times X$. Entonces el algoritmo Neighbor-Joining reconstruye T y sus longitudes de rama.*

Siguiendo el mismo esquema de la sección anterior, presentamos este algoritmo mediante agrupaciones. Para cada grupo C conteniendo un determinado número de especies, generalizando r_i definimos $u(C) = \frac{1}{N-2}\sum_{C' \neq C} d(C, C')$ como medida de separación entre C y los demás grupos, donde N es el número de grupos C en cada iteración.

El algoritmo resulta:

NEIGHBOUR-JOINING(d,n)

Formar n grupos, cada uno compuesto por un único elemento
Construir el grafo T asignando a cada hoja un grupo
mientras haya más de un grupo:
    Encontrar los grupos C_1 y C_2 minimizando $d(C_1,C_2)-u(C_1)-u(C_2)$
    Fusionar C_1 con C_2 en un nuevo grupo C
    Calcular $d(C,C^*)=\frac{d(C_1,C^*)+d(C_2,C^*)-d(C_1,C_2)}{2}$ para todos grupos C^*
    Añadir un nuevo vértice C a T y conectarlo a los vértices C_1 y C_2
    Asignar longitud $\frac{1}{2}d(C_1,C_2)+\frac{1}{2}(u(C_1)-u(C_2))$ a la arista (C_1,C)
    Asignar longitud $\frac{1}{2}d(C_1,C_2)+\frac{1}{2}(u(C_2)-u(C_1))$a la arista (C_2,C)
    Eliminar filas y columnas de D correspondientes a C_1 y C_2
    Añadir una fila y una columna a (d_{ij}) para el nuevo grupo C
devolver T

Observar que si aplicamos el algoritmo Neighbour-Joining a distancias ultramétricas nos va a dar un árbol sin raíz pero que puede convertirse perfectamente en el árbol molecular que obtendríamos usando UPGMA simplemente poniendo la raíz a la misma distancia de todos los taxones.

Muy frecuentemente NJ se utiliza aún cuando la función δ no satisface la condición de los cuatro puntos, ya que casi nunca la verifica. Aún más, en la práctica la desigualdad triangular es muy complicada de satisfacer y casi siempre se usa lo que llamamos una "pseudodistancia"d, eliminando entonces la condición iv) de la definición (1.1.6).

Luego, concluimos que a pesar que bajo circunstancias especiales el algoritmo UPGMA puede ser fidedigno, más rápido e incluso preferido al algoritmo NJ, Neighbour Joining abarca un mayor rango de casos y por ello es el que se utiliza comunmente para construir árboles filogenéticos.

Sin embargo pueden ocurrir varias anomalias al aplicar el algoritmo NJ a esta pseudodistancia: se puede producir más de un árbol, pueden producirse ramas de longitud negativa (no por ello desechables, tienen interpretación biológica), es decir, que la función distancia obtenida a partir de estos árboles d quizá no coincida con la distancia δ, original. Dada la matriz de distancias (δ_{ij}), y una vez reconstruido el árbol con matriz de distancias (d_{ij}), podemos preguntarnos cuánto se acerca d a δ.

Basándonos en este planteamiento de minimizar distancias, tenemos el método de reconstrucción de árboles mediante mínimos cuadrados:

$$\rho(d,\delta)=\sum_{1\leq i<j\leq n}(d_{ij}-\delta_{ij})^2,$$

donde d y δ son dos pseudodistancias sobre el mismo conjunto de taxones $X=\{x_1,...,x_n\}$. Además, si T es el árbol que relaciona los taxones de X:

$$ss_d(T)=\rho(d,\delta)$$

Este método de reconstrucción minimizando ss_d se denomina método de mínimos cuadrados. Fitch y Margoliash en [VSL10] proponen un método de mínimos cuadrados que consiste en asignar distintos pesos $w(e_i)$ a cada arista e_i y, utilizando una matriz de incidencia M de dimensiónes $\frac{n(n-1)}{2}\times(2n-3)$, obtiene el árbol óptimo T hallando los pesos que minimizan la ecuación:

$$E=||d-\delta||^2=||M\mathbf{w}-\delta||^{\mathbf{2}}$$

donde d,δ son las matrices de distancia y disimilaridad respectivamente, expresadas en forma de vector columna (notar que al ser simétricas, son vectores $\frac{n(n-1)}{2}\times 1$) y $\mathbf{w}$ es el vector conteniendo todos los pesos de las aristas.

Capítulo 2

Modelos evolutivos

2.1. Máxima parsimonia: Algoritmo Fitch-Hartigan

Los métodos desarrollados hasta ahora no requerían de niguna formalización relacionada con el proceso de mutación del ADN de generación en generación. De acuerdo a esta idea de vincular evolución con ADN, tomamos como datos secuencias para las cuales cada taxón quizá esté en un número finito de estados. Estas secuencias son las cadenas de ADN de cada especie, formadas por carácteres que pueden tener cuatro estados diferentes correspondientes a los cuatro tipos de nucleótidos: adenina (A), guanina (G), timina (T) y la citosina (C).

Si pensamos en un cambio de estado como un evento evolutivo inusual o extraordinario, deberíamos buscar árboles donde se evite que el mismo evento se dé en múltiples ocasiones. Esta idea es la que hay detrás del concepto de compatibilidad .

Definición 2.1.1. *Un carácter es compatible en un árbol si todos los nodos con el mismo estado forman un árbol conexo.*

Definición 2.1.2. *Un árbol T forma una filogenia perfecta si todos los caracteres son compatibles en T.*

Encontrar un T que forme una filogenia perfecta es un problema computacional de tipo NP-duro para un número de taxones, caracteres y posibles estados. Incluso con 4 estados por carácter no es fácil encontrar algoritmos que decidan la existencia y construyan un T que sea una filogenia perfecta. (Véase [VSL10] para una versión extensa del problema y su relación con otros problemas de teoría de grafos)

La soluciones existentes suelen asumir que no pueden aparecer el mismo carácter de forma independiente o que no hay lugar a la reversibilidad o desaparición de un carácter una vez presente. Por todo ello buscamos alternativas algo más flexibles desde el punto de vista matemático y más compatibles con la biología. Empezamos introduciendo un procedimiento para medir el grado de compatibilidad de los datos, cuyo criterio principal será el **principio de máxima parsimonia**:

El mejor árbol inferido a partir de los datos será el que tenga menos cambios entre estados.

Es decir, un buen árbol es aquel que pueda describir la historia evolutiva con el menor número de cambios posibles. A continuación una serie de definiciones y un teorema que recojen esta idea, y que justificará la obtención del árbol con máxima parsimonia.

Definición 2.1.3. *Un **carácter** con un conjunto de estados S en un conjunto X es una función $\chi : X \longrightarrow S$. Si $s = |S|$, decimos que χ es un carácter con s estados.*

Volviendo a nuestro conjunto de datos, para cada taxón del conjunto X hay una secuencia finita de caracteres $\mathscr{C} = \{\chi_1, \chi_2, ..., \chi_m\}$ siendo m la longitud de la secuencia y donde χ_i es el carácter i- ésimo asociado al conjunto de estados S_i. En el caso del ADN tenemos como conjunto de posibles estados $S_i = \{A, C, G, T\}$, $s_i = 4$ para todo $i = 1, \ldots, m$.

Ahora que tenemos nuestros datos compuestos de carácteres χ en X, calcular el número de cambios en T requiere considerar los carácteres $\widetilde{\chi}$ en todo el conjunto de vértices $V(T)$. De acuerdo con esto, decimos que un carácter $\widetilde{\chi}$ en $V(T)$ es una extensión de χ a T si $\widetilde{\chi}(x) = \chi(x)$ para todo $x \in X$. Denotamos con $Ext_T(\chi)$ al conjunto de todas las estensiones de χ a T (con sus elementos representando posibles historias evolutivas que son consistentes con las observaciones de χ) .

Para cada eje $e = \{v, w\} \in E(T)$ y el carácter $\widetilde{\chi}$ en $V(T)$, definimos:

$$\delta(e, \widetilde{\chi}) = \begin{cases} 1 & \text{si } \widetilde{\chi}(v) \neq \widetilde{\chi}(w) \\ 0 & \text{en otro caso} \end{cases}$$

Vista como función de e, con $\widetilde{\chi}$ fija, esta es la función indicadora de aquellas aristas en las cuales ocurre un cambio en la historia evolutiva $\widetilde{\chi}$.

Definición 2.1.4. *El **número total de cambios de estado de** $\widetilde{\chi}$ en un árbol filogenético T es el número de aristas en el cual ocurre un cambio de estado para $\widetilde{\chi}$:*

$$c(\widetilde{\chi}, T) = \sum_{e \in E(T)} \delta(e, \widetilde{\chi}),$$

*El **total de parsimonia de un árbol filogenético** T **para un carácter** χ en X es el mínimo total de cambios de estado alcanzable por una extensión de $\widetilde{\chi}$:*

$$ps_\chi(T) = \min_{\widetilde{\chi} \in Ext_T(\chi)} c(\widetilde{\chi}, T)$$

Decimos que $\widehat{\chi}$ es la mínima extensión de χ para T si:

$$c(\widehat{\chi}, T) = ps_\chi(T)$$

Estas definiciones son para un sólo carácter. Para un secuencia de caracteres tendremos:

Definición 2.1.5. *El **total de parsimonia de un árbol filogenético** T para una secuencia de caracteres $\{\chi_1, \chi_2, ..., \chi_m\}$ de un conjunto X es la suma de las parsimonias de cada carácter.*

$$ps_{\{\chi_1, ..., \chi_m\}}(T) = \sum_{i=1}^{m} ps_{\chi_i}(T)$$

De esta manera, el conjunto de árboles más parsimoniosos para una secuencia de caracteres $\{\chi_1, ..., \chi_m\}$ es la colección de árboles con el mínimo valor total de parsimonia:

$$\{T | ps_{\{\chi_1, ..., \chi_m\}}(T) = \min_{T'} ps_{\{\chi_1, ..., \chi_m\}}(T')\}$$

El problema de encontrar este conjunto es calcular $ps_{\{\chi_1, ..., \chi_m\}}(T')$. Para ello se desarrolló el algoritmo Fitch - Hartigan (1973) cuya descripción completa la podemos encontrar en el Apéndice. Sin embargo, este algoritmo a pesar de calcular la parsimonia de cada árbol, no está claro que obtenga el mínimo de posibles mutaciones necesarias para generar el árbol. Por ello tenemos el siguiente teorema cuya demostración la podemos encontrar en [AR12]:

Teorema 2.1.6. *Sea χ un carácter en X, y T^ρ un árbol binario filogenético con raíz. Entonces el algoritmo Fitch-Hartigan calcula $ps_\chi(T)$. Es más, el conjunto de estados asignados hasta la raíz ρ es exactamente el conjunto de estados que se dan en ρ asociados a la mínima extensión $\widetilde{\chi}$ de χ.*

Pero como se puede observar en las ecuaciones (1.1) y (1.2), el número de topologías crece exponencialmente conforme aumentamos el número de taxones. Por lo que cuando n comienza aumentar, la aplicación del algoritmo a todas las topologías es un proceso mucho más lento que el NJ ó UPGMA aunque la topología resultante sea la mejor que se pueda encontrar de acuerdo a criterios establecidos.

Sin embargo, dado este número elevado de topologías para n taxones, un método muy usual a la hora de encontrar el árbol más parsimonioso dado un árbol T, es el *nearest-neighbor interchange (NNI)* cuyo proceso consiste en eliminar una arista interior del árbol y sustituirla por una nueva, obteniendo dos nuevos árboles (Figura 2.1). El proceso se repite para cada una de las $n-3$ ramas interiores, así que obtiene $2(n-3)$ árboles.

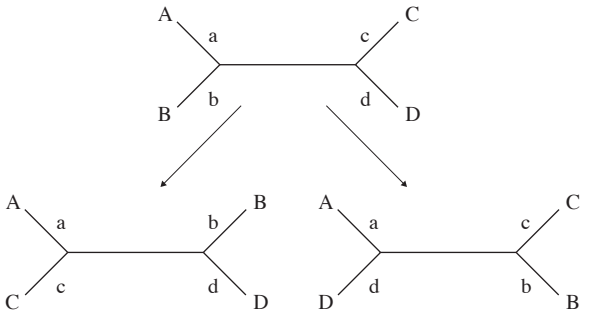

Figura 2.1: Aplicación del algoritmo NNI a un cuarteto

Este algoritmo de búsqueda local, con ayuda del algoritmo Fitch-Hartigan se va acercando progresivamente al mejor árbol calculando la parsimonia de cada árbol obtenido. Recalcar que el método de máxima parsimonia, a parte de ser un problema NP duro, es una aproximación muy simple y no es estadísticamente consistente (con datos suficientes, no garantiza obtener el árbol verdadero), como demostró Joe Felsenstein en 1978 [F81]. El principio de máxima parsimonia puede ser inconsistente en ciertas condiciones debido al problema de "long branch attraction" que aparece cuando se aplica máxima parsimonia sobre caracteres que han evolucionado de forma más rápida que el resto de los analizados. Este defecto metodológico tiende a agrupar estas secuencias en la base del árbol filogenético, independientemente de su verdadera proximidad, ya que no es capaz de diferenciar un cambio rápido de gran magnitud de una divergencia conseguida en un periodo de tiempo mucho más dilatado.

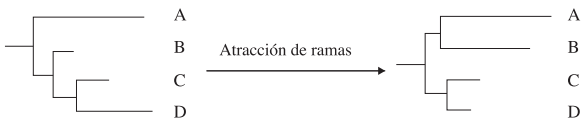

Figura 2.2: Ejemplo del problema de atracción de ramas.

Este problema se puede corregir de diversas formas: aumentando el número de taxones, empleando técnicas de parsimonia ponderada o el más usual y que veremos en la siguiente sección que es recurrir a métodos basados en máxima verosimilitud.

2.2. Introducción y propiedades de cadenas de Markov

El método de la máxima parsimonia tiene en consideración las mutaciones entre las distintas secuencias. Sin embargo, al tratarlas como eventos inusuales no tiene en cuenta las mutaciones múltiples que un lugar de la sucesión puede experimentar hasta convertirse (o no) en otro nucleótido.

Por ello introducimos los modelos evolutivos basados en cadenas de Markov que describirán el proceso de sustitución en una cadena de ADN a través del tiempo. Además, constituyen el engranaje necesario para relacionar los datos (cadenas de nucleótidos) con los procesos de reconstrucción que hemos visto en el capítulo anterior. La cadena de Markov con la que vamos a modelizar este proceso va a tener cuatro estados, correspondientes a los cuatro tipos de nucleótidos S={A,G,C,T}, donde la matriz de transición de probabilidades será de la forma:

$$P(t) = \begin{pmatrix} p_{AA}(t) & p_{AG}(t) & p_{AC}(t) & p_{AT}(t) \\ p_{GA}(t) & p_{GG}(t) & p_{GC}(t) & P_{GT}(t) \\ p_{CA}(t) & p_{CG}(t) & p_{CC}(t) & p_{CT}(t) \\ p_{TA}(t) & p_{TG}(t) & p_{TC}(t) & p_{TT}(t) \end{pmatrix},$$

donde $p_{ij}(t)$ es la probabilidad de cambio en el nucleótido predominante de i a j en un periodo de tiempo t, donde $i, j \in S$. Por comodidad en la notación, también nos referiremos a los estados A, G, C y T mediante los números $1, \ldots, 4$ respectivamente.

A pesar de no ser biológicamente aceptable, de momento, para cada secuencia $(i_1, \ldots, i_m)$, asumiremos que cada lugar i_j se comporta de manera idéntica e independiente $j = 1, \ldots, m$. También debemos hacer otro supuesto ya que el proceso de sustitución de los nucleótidos va de acuerdo a este proceso de Markov *(Propiedad de Markov)*: Si para un tiempo t_0 estábamos en el estado $i \in S$, entonces la probabilidad del evento que en tiempo $t + t_0$ estemos en el estado $j \in S$ depende sólo de i, j y t (exáctamente, es el elemento $p_{ij}(t)$ de la matriz $P(t)$). Luego de acuerdo a ello tenemos las ecuaciones de *Chapman-Kolmogórov*:

$$p_{ij}(t+\tau) = \sum_{k \in S} p_{ik}(t) p_{kj}(\tau) \text{ para todo } i, j.$$

Matricialmente podemos expresar esta relación como:

$$P(t+\tau) = P(t)P(\tau). \tag{2.1}$$

Consideraremos únicamente el caso de cadenas contínuas de Markov regulares, lo que significa que $P(0)$ es la matriz identidad I y que $P(t)$ es diferenciable para todo $t \geq 0$. Varios de los modelos que veremos cumplen la propiedad de reversibilidad del tiempo esto es, que el proceso estocástico para construir el árbol desde una especie hasta el antecesor común es el mismo que desde la raíz hacia las hojas, invirtiendo el flujo del tiempo.

Definición 2.2.1. *Sea π el vector de iniciación de probabilidades de una cadena de Markov contínua y suponer que $\pi_i \neq 0$ para todo i∈S. Definimos la cadena de Markov inversa como la cadena de Markov contínua dada por la matriz de transición de probabilidades $P^*(t)$ con :*

$$p^*_{ij}(t) = \frac{\pi_j p_{ji}(t)}{\pi_i} \text{ para todo } i, j \in S,\ t \geq 0 \tag{2.2}$$

Definición 2.2.2. *Una cadena de Markov cumpliendo las condiciones de la definición (2.2) se dice que es reversible en el tiempo o reversible si $P^*(t)$=$P(t)$ para todo t≥0.*

Si cada componente de π es no nula, para $\pi = \varphi$, siendo φ la distribución de probabilidad estacionaria de la cadena de Markov, la definición anterior equivale a verificar:

$$\varphi_i p_{ij}(t) = \varphi_j p_{ji}(t). \tag{2.3}$$

2.3. Modelos fundamentales

Recordemos la definición de exponencial de una matriz A:

Definición 2.3.1. *Sea A una matriz cuadrada $m \times m$. Entonces e^A es la matriz $m \times m$ dada por la siguiente serie:*

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}. \tag{2.4}$$

Entonces, de acuerdo con la definición (2.4) y bajo las condiciones que acabamos de mencionar tenemos el siguiente teorema:

Teorema 2.3.2. *Tenemos que $P(t)$ es de la forma:*

$$P(t) = e^{tQ}, \tag{2.5}$$

donde $Q = P'(0)$, matriz 4×4.

Demostración. Por (2.1) y la regularidad de la cadena de Markov tenemos que para $t \geq 0, h > 0$:

$$\frac{P(t+h) - P(t)}{h} = \frac{P(t)(P(h) - I)}{h} = \frac{P(t)(P(h) - P(0))}{h} = \frac{(P(h) - P(0))P(t)}{h}$$

Cuando $h \to 0$ tenemos:

$$P'(t) = P'(0)P(t),$$

Como $P'(0)$ es una matriz de coeficientes constantes, la solución del sistema diferencial es $P(t) = e^{tQ}$ con $Q = P'(0)$. □

Esta matriz $Q = (q_{ij})$, $1 \leq i, j \leq 4$, se denomina matriz de cambio instantáneo (rate matrix). Una propiedad importante de Q es que sus filas suman 0 y podemos interpretar estos q_{ij} como la tasa instantánea (de sustituciones en un lugar de la secuencia por unidad de tiempo) en la cual i es sustituido por j.

2.3.1. Modelo de Jukes-Cantor

Este primer modelo está basado en una cadena de Markov en tiempo contínuo sobre espacio discreto y verificando las propiedades mencionadas en la sección anterior. El modelo de Jukes-Cantor se caracteriza por poseer la siguiente matriz de cambio instantáneo:

$$Q = \begin{pmatrix} -\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & -\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & -\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & -\alpha \end{pmatrix}, \tag{2.6}$$

Esta matriz nos indica que las tasas de cambio de base son la misma, $\alpha/3$, por lo que puede no ser un modelo muy realista. Además, observar que la tasa total con la que cualquier base específica cambia a cualquiera de las otras bases es α. Hallemos una expresión explícita de $P(t)$ mediante los valores y vectores propios de Q para conseguir su diagonalización:

$$Q = SDS^{-1},$$

donde:

$$S = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \; D = \operatorname{diag}\left(0, -\frac{4}{3}\alpha, -\frac{4}{3}\alpha, -\frac{4}{3}\alpha\right).$$

Luego la matriz de Markov de Jukes-Cantor es:

$$\begin{aligned} P(t) &= e^{Qt} \\ &= Se^{Dt}S^{-1} \\ &= S \operatorname{diag}\left(1, e^{-\frac{4}{3}\alpha t}, e^{-\frac{4}{3}\alpha t}, e^{-\frac{4}{3}\alpha t}\right) S^{-1} \\ &= \begin{pmatrix} 1-a & a/3 & a/3 & a/3 \\ a/3 & 1-a & a/3 & a/3 \\ a/3 & a/3 & 1-a & a/3 \\ a/3 & a/3 & a/3 & 1-a \end{pmatrix}, \end{aligned}$$

donde:

$$a = a(t) = \frac{3}{4}\left(1 - e^{-\frac{4}{3}\alpha t}\right). \tag{2.7}$$

La distribución estacionaria φ de los estados es:

$$\varphi = \begin{pmatrix} 1/4, & 1/4, & 1/4, & 1/4 \end{pmatrix} P(t) = \begin{pmatrix} 1/4, & 1/4, & 1/4, & 1/4 \end{pmatrix}.$$

Además si suponemos que la distribución en la secuencia ancestral (vector de iniciación de probabilidades) es equiprobable:

$$\pi = \begin{pmatrix} 1/4, & 1/4, & 1/4, & 1/4 \end{pmatrix},$$

entonces, por (2.3) tenemos que este modelo es reversible en el tiempo.

Ahora que tenemos el modelo de ADN definido, podemos desarrollar un distancia basada en dicho modelo. Para ello consideramos la secuencia primigenia S_0 con distribución de base $\pi = (1/4, 1/4, 1/4, 1/4)$ con un proceso de mutación regido por la matriz (2.6). Sea S_1 un descendiente de S_0 después de tiempo t, la distribución conjunta de los distintos carácteres en cada lugar de las dos sucesiones S_0, S_1 será:

$$\widehat{P}(t) = \operatorname{diag}(\pi)P(t) = \begin{pmatrix} (1-a)/4 & a/12 & a/12 & a/12 \\ a/12 & (1-a)/4 & a/12 & a/12 \\ a/12 & a/12 & (1-a)/4 & a/12 \\ a/12 & a/12 & a/12 & (1-a)/4 \end{pmatrix}, \tag{2.8}$$

donde las filas se refieren al estado S_0 y las columnas al estado S_1.

Interpretamos a como la probabilidad de que S_0 y S_1 presenten carácteres diferentes. Notar que puede haber ocurrido mediante una sola sustitución o bien, a través de una sucesión de sustituciones ya que al ser un modelo de Markov contínuo tiene en cuenta todas las posibles maneras de llegar al estado final desde el inicial.

En la práctica, la estimación de a la haremos de forma que la distribución teórica esté lo más próxima a la empírica. Utilizando la información aportada por todos los lugares de la cadena, podemos definir una aproximación de a (también denominada *distancia de Hamming*):

$$\hat{a} = \frac{\text{número de lugares que constan de estados distintos}}{\text{número total de lugares de la secuencia}},$$

Observar que α y t aparecen en forma de producto en (2.7) . La interpretación es muy clara: es el producto de la tasa, medida en unidades de (sustituciones en un lugar de la secuencia / unidad de tiempo) por el tiempo. Luego αt tiene significado en sí mismo, es el número total esperado de sustituciones que ocurren en ese lugar durante todo el periodo de tiempo, incluyendo aquellas mutaciones que están ocultas debido a las múltiples sustituciones.

Podemos ver esta tasa α como constante (de hecho al especificar la matriz de tasas para el modelo ya lo estamos suponiendo) viendo el tiempo no acorde a un reloj convencional sino a uno que pueda acelerarse o frenarse.

Despejando αt de (2.7), podemos estimar el total de mutación:

$$\widehat{\alpha t} = -\frac{3}{4}\ln\left(1-\frac{4}{3}\hat{a}\right).$$

Definimos la distancia de *Jukes-Cantor* entre las cadenas de ADN S_0 y S_1 como:

$$d_{JC}(S_0,S_1) = -\frac{3}{4}\ln\left(1-\frac{4}{3}\hat{a}\right). \tag{2.9}$$

De manera que, dados S_i, S_j con ancestro común S_0 desconocido sabemos que las matrices $P_i(t_i)$ y $P_j(t_j)$ describen el proceso de paso de S_0 a S_i y de S_0 a S_j respectivamente. Por la propiedad de reversibilidad en el tiempo sabemos que S_0 y S_j tendrán la misma distribución que si pensamos como ancestro S_j, así, el proceso conjunto de ir de S_i a S_j a través de S_0 viene descrito por la matriz $P_iP_j = P(t_i+t_j)$, y por tanto la distancia de Jukes-Cantor $d_{JC}(S_i,S_j)$ estima el total de mutación $\alpha(t_i+t_j)$ entre S_i y S_j.

Notar que si tubiésemos secuencias de carácteres de longitud infinita, tendríamos estimada α sin error y la distancia d_{JC} nos daría una matriz de distancias entre especies aditiva y que permitiría reconstruir el árbol métrico con cualquiera de los algoritmos vistos en el capítulo anterior.

Desde un principio hemos utilizado el modelo de Jukes-Cantor que consta de un único parámetro y una distribución de base primigenia uniforme, pero en bioinformática se han desarrollado modelos más complejos.

2.3.2. Extensión del modelo de Jukes-Cantor

El modelo de Kimura [K80] no trata por igual todas las mutaciones, distingue entre transiciones (sustituciónes de una base púrica A,G por otra base púrica o bien una base pirimidínica C,T por otra pirimidínica) y transversiones (sustituciones de una purina por una pirimidina o viceversa), de manera que el modelo Kimura de dos parámetros γ, β define la siguiente matriz de cambio instantáneto:

$$Q = \begin{pmatrix} * & \beta & \gamma & \gamma \\ \beta & * & \gamma & \gamma \\ \gamma & \gamma & * & \beta \\ \gamma & \gamma & \beta & * \end{pmatrix}, \quad * = -2\gamma-\beta,$$

donde β es la tasa de sufrir una transición y 2γ es la tasa de sufrir una transversión. De manera análoga al modelo anterior se puede obtener la matriz de Markov y la distancia generada por este modelo. También la distribución primigenia, es uniforme y asumida como distribución estacionaria en la raíz da un proceso reversible.

También tenemos otros modelos como son el modelo Kimura de tres parámetros, que distingue entre las distintas transversiones, o el modelo de Felsenstein [F81] que también es una generalización de Jukes-Cantor al variar las frecuencias de base, todos ellos son reversibles en el tiempo, pero el más utilizado es el *general time-reversible model* (GTR) donde Q es de la forma:

$$Q = \begin{pmatrix} * & p_G\alpha & p_C\beta & p_T\gamma \\ p_A\alpha & * & p_C\delta & p_T\varepsilon \\ p_A\beta & p_G\delta & * & p_T\eta \\ p_A\gamma & p_G\varepsilon & p_C\eta & * \end{pmatrix}, \tag{2.10}$$

donde $\alpha, \beta, \gamma, \delta, \eta, \varepsilon \geq 0$, donde la diagonal de Q es tal que las filas suman cero y una distribución en la raíz arbitraria $\pi = (p_A, p_G, p_C, p_T)$ con $p_A + p_G + p_C + p_T = 1$. Este modelo engloba a todos los demás, observar que Kimura dos parámetros es el caso particular de base primigenia homogénea y $\alpha = \eta, \beta = \gamma = \delta = \varepsilon$.

Todos estos modelos asumen que cada lugar de la secuencia se comporta idénticamente, sin embargo, esta suposición está lejos de ser biológicamente justificable. Es por ello que en la práctica se utilizan modelos híbridos que permiten no sólo la variabilidad o invariabilidad de los lugares de una secuencia sino también la variación en la velocidad a la que estos nucleótidos mutan.

Dadas n secuencias, para un árbol fijo T y un modelo evolutivo cualquiera GM (general model, donde GM puede ser JC69 (Jukes-Cantor), K80 (Kimura) entre otras variantes) asociado a él, introducimos el parámetro r indicando que con probabilidad r un lugar de la secuencia varía, y con probabilidad $1-r$ se mantiene invariante. Además definimos un vector π_{inv} indicando la distribución de los lugares invariantes, $\pi_{inv} = (q_A, q_G, q_C, q_T)$.

Para calcular la distribución conjunta de este modelo debemos analizar los dos tipos de lugares por separado. Por un lado, para los lugares en los que se permite mutación, como veremos en este capítulo, la distribución de probabilidad $\widehat{P_1}(i_1, \ldots, i_m)$ de observar este array en el lugar $i-$ésimo de las hojas vendrá dada por la fórmula (2.12) que veremos en la siguiente sección.

Mientras que para los lugares invariantes podríamos hacer un cálculo similar utilizando como matriz de Markov la matriz identidad. De hecho, podemos expresar su distribución conjunta mediante la siguiente función:

$$\widehat{P_2}(i_1, \ldots, i_n) = \begin{cases} q_{i_1} & \text{si } i_1 = i_2 = \ldots = i_n, \\ 0 & \text{en otro caso.} \end{cases}$$

Luego la distribución conjunta de este nuevo modelo será:

$$\widehat{P} = r\widehat{P_1} + (1-r)\widehat{P_2}. \tag{2.11}$$

Este modelo que permite la invariabilidad de mutación se denomina modelo GM + I. Notar que las fórmulas de las distancias de los modelos asumían que no había lugares invariantes, por lo que el uso de estas fórmulas requieren que el número de lugares invariables sea despreciable puesto que si no es así, puede haber un sesgo sistemático en las distancias inferidas.

A parte de la variabilidad o no de los lugares de la secuencia, no todos ellos tienen por qué mutar a la misma velocidad. Agrupamos los lugares de cada secuencia de acuerdo a su velocidad de mutación de manera que tenemos g clases o grupos. Introducimos factores de escala λ_j para acelerar o frenar el proceso de cambio de cada clase j con $j = 1, \ldots, g$, y sea $s = (s_1, \ldots, s_g)$ cuyos elementos suman 1 representando el tamaño relativo de los distintos grupos con idéntica velocidad de mutación.

Luego, para cada grupo j, y arista e con longitud t_e tendremos una matríz de cambio $\lambda_j Q$ con matriz de Markov $M_e = e^{t_e \lambda_j Q}$. Ahora podemos calcular de manera directa la distribución conjunta de las sucesiones de ADN. Y por tanto la distribución conjunta de este modelo será:

$$\widehat{P} = \sum_{j=1}^{g} s_j \widehat{P_j}.$$

La distribución de las tasas s en la práctica viene dada por una distribución contínua con función de densidad:

$$s_\alpha(\lambda) = \frac{\alpha^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\alpha\lambda}.$$

con α parámetro de forma. Así tenemos que la distribución de este modelo viene dada por:

$$\widehat{P} = \int_\lambda s(\lambda)\, \widehat{P}_\lambda\, d\lambda.$$

El modelo GM al que se permite variar la velocidad del proceso de mutación se denomina modelo GM $+\Gamma$. A menudo es desable combinar este modelo con el anterior creando el modelo conjunto GM $+\Gamma + \mathrm{I}$ que biológicamente es más acorde al árbol.

2.4. Estimación por Máxima Verosimilitud

Un método muy habitual a la hora de ajustar un modelo evolutivo y encontrar sus parámetros óptimos es máxima verosimilitud. Sin embargo, a pesar de las simplificaciones que asumimos a la hora de modelizar el problema, obtener la máxima verosimilitud presenta una elevada carga computacional.

Definimos la función de verosimilitud para nuestro modelo dotado de un conjunto de parámetros p como:

$$L(p) = L(p|datos) = \mathscr{P}(datos|p).$$

Definición 2.4.1. *Sea un modelo evolutivo acorde a nuestro conjunto de datos, un estimador de máxima verosimilitud (ML) para el conjunto de parámetros p del modelo, es un conjunto de valores $\hat{p}$ que maximizan la función de verosimilitud $L(p|datos)$.*

Antes de generalizar para n taxones, veámoslo para $n = 4$. Suponer que tenemos el árbol T que se muestra en la figura 2.3 y asociado a él un modelo evolutivo con matriz de Markov P_e para cada eje e. Entonces, la probabilidad de observar el vector $(1_j,\ldots,4_j)$ donde i_j es el lugar $j-$ésimo de la sucesión i, $j = 1,\ldots m$, $i = 1,\ldots,4$ será:

$$p(1_j,...,4_j) = \sum_{k=1}^{4}\sum_{w=1}^{4}\sum_{z=1}^{4} \pi_k P_1(k,w) P_2(w,1_j) P_3(w,2_j) P_4(k,z) P_5(z,3_j) P_6(z,4_j).$$

De manera que la función de verosimilitud será:

$$\ln L_T = \sum_{(1_j,\ldots,4_j)\in\{A,G,C,T\}^4} n(1_j,\ldots,4_j) \ln(p(1_j,\ldots,4_j)),$$

donde $n(1_j,\ldots,4_j)$ la frecuencia absoluta del diseño $(1_j,\ldots,4_j)$ observado en la secuencia de datos.

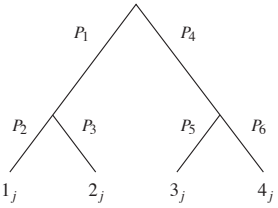

Figura 2.3: Árbol binario de cuatro hojas con raíz.

Para n secuencias, el cálculo es análogo. De manera que fijada una topología T de árbol compuesto por n taxones en X cuyas secuencias tienen longitud m con un modelo evolutivo asociado a él con matriz de Markov P_e para cada eje e y vector de iniciación de probabilidades π, la distribución de los caracteres en las hojas podemos expresarla de la siguiente manera, sea $(i_1,...,i_n)$ vector conteniendo la letra i - ésima de las n secuencias, entonces, probabilidad asociada a $(i_1,...,i_n)$ será:

$$p(i_1,...,i_n) = \sum_{k=1}^{4}\sum_{j=1}^{4}\dots\sum_{s=1}^{4}\pi_k P_1(k,i_1)P_2(k,j)P_3(j,i_2)\dots P_r(s,i_n), \tag{2.12}$$

Entonces para cada árbol T, la función de verosimilitud será:

$$\ln L_T = \sum_{(i_1,\dots,i_n)\in\{A,G,C,T\}^n} n(i_1,\dots,i_n)\ln(p(i_1,\dots,i_n)). \tag{2.13}$$

Notar que los parámetros de nuestro modelo no sólo son $\alpha,\beta,\gamma,\delta,\eta,\varepsilon$ sino también las tres probabilidades de π, el árbol filogenético binario, que es un parámetro no numérico, y las longitudes de sus ejes $\{t_e\}_{e\in E(t)}$.

Una vez obtenido (2.13) debemos encontrar los parámetros que lo maximizan. Como es obvio, el proceso en sí involucra demasiados cálculos para hacerse a mano, e incluso con un ordenador optimizar los parámetros para cada árbol tiene un coste computacional muy elevado ya que calcular ML implica:

1. Calcular cada $n(i_1,\dots,i_n)$, que puede realizarse directamente.
2. Considerar todos los árboles posibles T que relacionan a los taxones. Sabemos por (1.1), que tenemos $(2n-5)!!$ árboles distintos.
3. Para cada árbol, construir la función de verosimilitud, que, en el caso genérico del modelo GTR, tenemos los parámetros que especificamos en el párrafo anterior: $5+3+(2n-3)$ (hemos fijado η como referencia temporal). Calculando la correspondiente matriz de Markov en cada eje e y longitud t_e, $P_e = e^{Qt_e}$ para poder calcular $p(i_1,...,i_n)$.
4. Para la función de verosimilitud construida en el paso anterior, encontrar su máximo.
5. Elegir el árbol T con sus parámetros, que tuviera la mayor verosimilitud.

Recalcar que en el tercer paso, calcular $p(i_1,...,i_n)$ mediante (2.12), a pesar de ser conceptualmente claro, involucra demasiados términos. Para n taxones, hay $n-1$ nodos internos en un árbol con raíz, por lo que habría 4^{n-1} términos en dicha suma.

El coste computacional lleva a plantear métodos heurísticos para calcular esta probabilidad de manera eficaz como es el algoritmo de Felsenstein: *Felsenstein pruning algorithm* [F81]. Este algoritmo ayudándose de que las cadenas de Markov verifican (2.1), utiliza un planteamiento similar al algoritmo Fitch-Hartigan que vimos al principio de este capítulo calcula la verosimilitud ascendiendo desde las hojas y multiplicando las matrices de Markov correspondientes en cada iteración.

Capítulo 3

Aplicación práctica.

En los últimos años, el acelerado desarrollo de la bioinformática ha hecho posible disponer de una gran cantidad de software para la estimación de árboles filogenéticos a partir de secuencias de datos como es el paquete CLUSTAL. Un listado de estos paquetes que ilustran la diversidad de posibilidades y opciones se encuentra en [UW]. Esta página consta de más de cien programas especializados en esta disciplina, que recorren una amplia gama de campos, desde programas generales de filogenética hasta programas especializados en parsimonia, cómputo de distancias o máxima verosimilitud.

En este trabajo optamos por elegir un software libre y de múltiples opciones para el análisis estadístico como es R. Podemos ver en [CR] la abundancia de paquetes que nos ofrece para el tratamiento de secuencias de ADN. Es importante señalar que no todos ellos están en el repositorio principal CRAN, algunos se desarrollan en otros lugares como R-Forge y no han sido todavía incorporados.

Los paquetes `ape` y `geiger` incluyen funciones para la lectura y representación de árboles filogenéticos, funciones para implementación de bases de datos de secuencias de ADN, funciones para manipulación de árboles o datos asociados a éstos, crear longitudes de ramas, obtener información sobre el tamaño y medidas del árbol u otras propiedades. También el paquete `ape` puede ser utilizado para la inferencia filogenética puesto que los procesos de reconstrucción de árboles (NJ, UPGMA,...) también están implementados en este paquete. No obstante los paquetes `phangorn` y `PHYLIP`, entre otros, lo complementan, pudiendo estimar árboles utilizando distancias, parsimonia o máxima verosimilitud.

En este último capítulo analizaremos, mediante las conocimientos adquiridos en las secciones previas y el uso de este software, dos conjuntos de datos bien diferenciados. El primero es un ejemplo basado en secuencia genómica disponible en [UW], hace referencia a catorce especies de mamíferos distintas seleccionadas por Masami Hasegawa de un conjunto de datos de secuencias de nucleótidos recogidas por un grupo de investigadores en Japón para los cuales podremos utilizar los conceptos y modelos introducidos en los dos capítulos previos aplicados a carácteres. El segundo conjunto de datos agrupa un género de pájaros cantores "geospiza" que Darwin, propulsor de la filogenética, descubrió en las Islas Galápagos y cuyas diferencias fundamentales radican en el tamaño y forma del pico, plenamente adaptados a las diferentes fuentes de alimento [W]. Este conjunto usa características numéricas a la hora de diferenciar a las especies.

3.1. Árbol filogenético de `Primates.dna`

Como ya hemos dicho, disponemos de catorce secuencias de ADN de diferentes especies, desde un ratón hasta un humano. Este conjunto de datos puede encontrarse en el paquete de R `DAAGbio`. El objetivo del estudio es encontrar el árbol óptimo y su mejor modelo evolutivo. Subrayar que el conjunto de datos "primates" es un archivo dna que está en formato FASTA, es decir, muestra la secuencia de ADN con limitación de longitud de diez carácteres por identificador de secuencia con un total de 232 nucleótidos por secuencia.

Reconstruimos la matriz de disimilaridad a partir de la distancia de Jukes-Cantor (2.9), siendo $\hat{a}$ la proporción de nucleótidos diferentes entre las dos secuencias. Esta matriz de distancias no es ni ultramétrica ni aditiva, aún así con ella podemos reconstruir el árbol mediante cualquiera de los dos métodos expuestos en el primer capítulo, bien Neighbour-Joining, o UPGMA, sabiendo que este último al no tratarse de una distancia ultramétrica, puede no obtener un árbol adecuado:

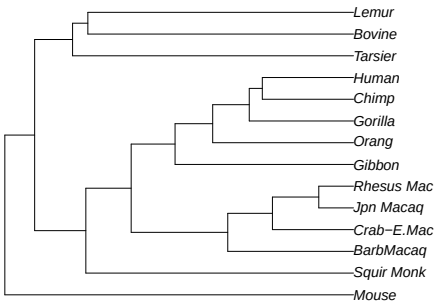

Figura 3.1: Representación en dendograma del árbol filogenético reconstruido mediante UPGMA.

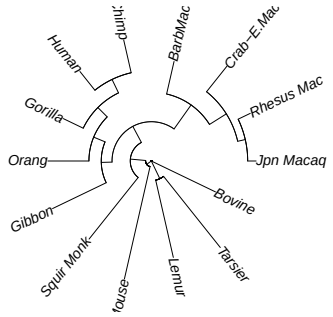

Figura 3.2: Representación en abanico del árbol filogenético reconstruido mediante Neighbour-Joining.

La pregunta que surge inmediatamente es qué árbol se ajusta mejor a nuestros datos, como vimos en el capítulo anterior podemos aplicar el algoritmo Fitch-Hartigan para el cálculo de la parsimonia de cada árbol. Ésto se puede realizar mediante la función `parsimony` de `phangorn`. Aplicando este primer algoritmo obtenemos unos valores de parsimonia para el árbol UPGMA y NJ de 751 y 746 respectivamente. Podemos aplicar el algoritmo heurístico NNI para intentar obtener un árbol con mejor parsimonia:

```
optim.parsimony(arbolNJ, primates, method="fitch",rearrangements="NNI")
Final p-score 746 after  0 nni operations
```

Queda confirmado que el árbol NJ que hemos obtenido es el más parsimonioso para la distancia de Jukes-Cantor y queremos buscar el mejor modelo que se adapte a él comparando mediante máxima verosimilitud. Podemos ajustar un primer modelo, Kimura 2-parámetros mediante la función `pml` de `phangorn`:

```
 loglikelihood: −2952.943

unconstrained loglikelihood: −1230.335

Rate matrix:
      a        c        g        t
a 0.000000 1.000000 3.778059 1.000000
c 1.000000 0.000000 1.000000 3.778059
g 3.778059 1.000000 0.000000 1.000000
t 1.000000 3.778059 1.000000 0.000000

Base frequencies:
0.25 0.25 0.25 0.25
```

Para este modelo tenemos una verosimilitud de $-2952{,}943$ con matriz de cambio instantáneo distinguiendo entre transversiones y transiciones y un vector de inicialización con probabilidades homogéneas. Planteamos un modelo más completo como es el GTR $+\Gamma +$ I, con valores iniciales $r = 0{,}2$ y $k = 4$ intérvalos de la distribucción gamma :

```
 loglikelihood: −2609.587

unconstrained loglikelihood: −1230.335
Proportion of invariant sites: 0.006035243
Discrete gamma model
Number of rate categories: 4
Shape parameter: 3.173903

Rate matrix:
      a         c         g         t
a 0.0000000 0.6476364 33.6872234 0.4062758
c 0.6476364 0.0000000 0.0083054 14.3962899
g 33.6872234 0.0083054 0.0000000 1.0000000
t 0.4062758 14.3962899 1.0000000 0.0000000

Base frequencies:
0.3918023 0.3796106 0.04023627 0.1883509
```

La función pml a partir de los valores iniciales del modelo y buscando la mejor verosimilitud optimiza los pesos de los ejes, el vector de iniciación de probabilidades, la matriz de cambio instantáneo, la proporción de lugares invariantes y el parámetro de forma (incluso optimiza la topología aplicando NNI en el proceso). Notar que este modelo GTR $+\Gamma +$I utiliza una gamma discreta ya que la distribución gamma contínua puede no ajustar bien los datos cuando el tamaño muestral aumenta, de acuerdo al artículo [SFBR03].

Mediante técnicas estadísticas compararemos los dos modelos para ver cuál se ajusta mejor a los datos. Una primera valoración, la realizamos mediante el cociente de razón de verosimilitudes disponible en la función `anova` y que nos da un p-valor menor que 0,001. Permite concluir que hay un incremento de verosimilitud significativo entre estos dos modelos:

```
> anova(fitK80, fitGTR)

 Likelihood Ratio Test Table

  Log lik.  Df  Df change  Diff log lik. Pr(>|Chi|)
1  −2952.9  26
2  −2609.6  35         9         686.71  < 2.2e−16 ***
−−−
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

Alternativamente, el test Shimodaira-Hasegawa basado en el método de remuestreo bootstrap, permite la estimación de la distribución muestral de un estadístico utilizando métodos de remuestreo aleatorios con reemplazo;

```
> SH.test(fitGTR, fitK80)
      Trees    ln L       Diff ln L   p−value
[1,]     1  −2609.586     0.0000     0.5002
[2,]     2  −2952.943   343.3569     0.0000
```

Con el p-valor de 0 se vuelve a confirmar que tenemos un aumento signficativo de la verosimilitud. Por fin, aplicamos el criterio de Información de Akaike (AIC) que combina bondad de ajuste y número

de parámetros necesarios y diseñado para evitar seleccionar modelos sobreajustados. De nuevo, los cálculos indican una preferencia del modelo GTR frente al modelo K80:

```
> AIC(fitK80)
[1] 5957.885
> AIC(fitGTR)
[1] 5289.171
```

Observar que el modelo GTR $+\Gamma$ $+$I consta de 9 parámetros más que el modelo de Kimura, y pese a ello, el AIC sigue siendo menor que el de Kimura. El Criterio de información Bayesiano cuyo término de penalización para el número de parámetros es mayor que en el AIC vuelve a afirmar que el modelo GTR $+\Gamma$ $+$I es el que se ajusta mejor al árbol filogenético que obtuvimos:

```
          df    BIC
fitK80    26 6047.5
fitGTR    35 5411.2
```

Como hemos comprobado. el modelo GTR$+\Gamma$ $+$I ajusta mejor que el de Kimura, pero como vimos en el capítulo anterior disponemos de muchos otros modelos (Jukes- Cantor (JC), Felsenstein (F81), Kimura (K80) y todos sus derivados: JC+Γ, F80+Γ+ I, ...) que podrían representar igual o mejor al conjunto de datos. Por ello una comparación entre todos los modelos es esencial, y pese a que se podría hacer siguiendo el mismo procedimiento comparativo que hasta ahora, la función `modelTest` de `Phangorn` permite cotejar todos los modelos disponibles calculando su AIC, BIC y verosimilitud como se puede ver en el cuadro 3.1. Esta función a pesar que nos permite visualizar de manera rápida todos los modelos, no es tan estricta como pml y por ejemplo, no aplica NNI al comienzo de la optimización para asegurarse que estamos con el árbol de mejor parsimonia.

Por tanto, el modelo óptimo y que mejor refleja los datos entre todos ellos sigue siendo el GTR$+\Gamma$ $+$I, que tiene la mínima verosimilitud, AIC y BIC, pese a tener el número de parámetros mayor, es el más completo desde el punto de vista biológico. Notar que en éste, la proporción de lugares invariantes es muy baja, 0,006, tenemos dos frecuencias de base que destacan notablemente sobre las demás, las correspondientes a los nucleótidos *A* y *C*, y dos tasas de cambio isntantáneo extremadamente elevadas en las bases púricas $A \leftrightarrow G$, seguida de las base pirimidínicas $T \leftrightarrow C$.

Model	df	logLik	AIC	BIC
JC	25.00	-3068.42	6186.83	6273.00
JC+I	26.00	-3062.63	6177.26	6266.87
JC+Γ	26.00	-3066.92	6185.83	6275.45
JC+Γ+I	27.00	-3062.64	6179.29	6272.35
F81	28.00	-2918.17	5892.33	5988.84
F81+I	29.00	-2909.12	5876.24	5976.20
F81+Γ	29.00	-2912.59	5883.17	5983.13
F81+Γ+I	30.00	-2908.52	5877.04	5980.44
K80	26.00	-2952.94	5957.89	6047.50
K80+I	27.00	-2944.51	5943.02	6036.08
K80+Γ	27.00	-2945.00	5944.00	6037.07
K80+Γ+I	28.00	-2942.38	5940.76	6037.27
GTR	33.00	-2642.95	5351.89	5465.64
GTR+I	34.00	-2624.04	5316.07	5433.26
GTR+Γ	34.00	-2613.67	5295.34	5412.53
GTR+Γ+I	35.00	-2610.31	5290.63	5411.26

Cuadro 3.1: Resumen de modelos.

3.2. Filogenésis del género de aves Geospiza.

En el paquete `geiger` vienen cargadas diferentes bases de datos, una de ellas es `geospiza` que consta de una tabla con trece tipos de especies de pájaros y sus cinco medidas de pico, tarso, culmen, ala y peso. Dado que no disponemos de las secuencias de ADN de estas especies y únicamente poseemos rasgos fenotípicos de éstas, vamos a utilizarlos para construir un árbol que relaciona a estas 13 especies.

Las variables están medidas en diferentes escalas, debemos primero estandarizar los datos (restando su media y dividiendo por su desviación típica). Ya tipificados, definimos una matriz de distancias. Una primera aproximación, es utilizar la distancia euclídea y podemos utilizar el método más general de reconstrucción filogenética, el algoritmo Neighbour-Joining.

El resultado se puede ver en la figura (3.3):

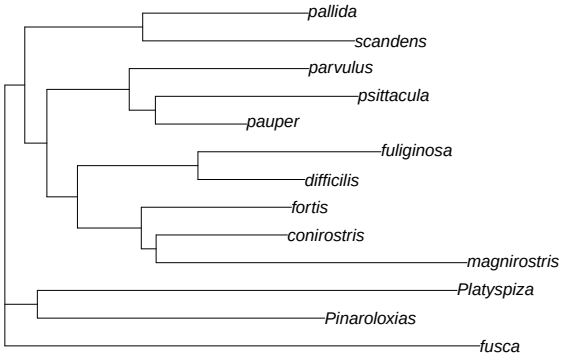

Figura 3.3: Árbol filogenético que relaciona las especies de género geospiza a partir de sus rasgos, mediante el algoritmo NJ.

Sin embargo, la distancia euclídea presenta un grave problema que no deriva directamente de su utilización, sino de la naturaleza de las propias variables. Si las variables utilizadas están correlacionadas, distancia euclídea inflará la disimilaridad entre ellas.

Una posible solución es ponderar la contribución de cada par de variables con pesos inversamente proporcionales a las correlaciones. De manera, que para dos especies $\mathbf{x} = (x_1, \ldots, x_s), \mathbf{y} = (y_1, \ldots, y_s)$ definimos la distancia de Mahalanobis como:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^t \sum^{-1} (\mathbf{x} - \mathbf{y})} \tag{3.1}$$

donde $\sum$ es la matriz de covarianza estimada. Para introducir esta distancia en R, debemos cargar el paquete `StatMatch`, no disponible en CRAN para la última versión de este software, y que contiene la función `mahalanobis.dist` que a partir del conjunto de datos geospiza, obtiene su matriz de distancias. Con esta matriz de datos estadarizados podemos construir el árbol que relaciona estas aves mediante Neighbour-Joining como se puede observar en la figura 3.4.

Como contamos con el árbol de consenso de esta familia de aves (disponible en [RP]) podemos comparar ambos árboles de métricas distintas con el biológicamente correcto. Aplicamos una primera medida de distancia entre dos árboles basada en el número de ramas internas que difieren [P06]:

```
> dist.topo(eucl.NJ, geotree)
[1] 18
```

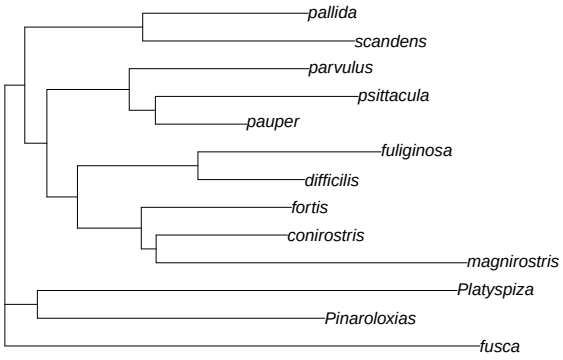

Figura 3.4: Árbol filogenético reconstruido mediante el algoritmo Neighbour-Joining con una matriz de distancias de Mahalanobis.

```
> dist.topo(mhls.NJ, geotree)
[1] 16
```

Por lo que el árbol de la distancia de mahalanobis tiene un número menor de ramas internas no acordes al original, es más, si aplicamos la medida de longitud de rama que propusieron Kuhner y Felsenstein (1994) como la raíz cuadrada de la suma de los cuadrados de las diferencias de las ramas internas obtenemos:

```
> dist.topo(eucl, geotree, method="score")
[1] 1.91005
> dist.topo(mhls.NJ, geotree, method="score")
[1] 1.025441
```

Estas diferencia es casi el doble en la euclídea que en mahalanobis lo que nos indica claramente que el árbol de distancias de Mahalanobis está mucho más cerca al árbol de consenso que el árbol de métrica euclídea [STIC07].

Como podemos observar, pese a que la distancia de Mahalanobis en un principio la hemos visto adecuada para el tratamiento de estas variables de rasgos fenotípicos, en seguida hemos comprobado que, con respecto al árbol original, hay serias diferencias. Es por ello que en bioinformática al igual que existen modelos evolutivos basados en secuencias de ADN, se han desarrollado modelos y técnicas mucho más avanzadas para el tratamiento en concreto de estas variables contínuas que tienen en cuenta el movimiento browniano entre partículas, es decir, el movimiento aleatorio que se observa en ellas.

Es claro que aplicar un modelo de este tipo es el siguiente paso en el análisis de este conjunto de datos. Una introducción sobre filogenética que trata en profundidad el tema de variables contínuas aplicadas a modelos brownianos desde el punto de vista del software R se puede encontrar en la página web [RP]. Sin embargo dada su complejidad, explicar este modelo basado en ecuaciones diferenciales y su implementación en R llevaría varias páginas y de hecho, podría considerarse en sí como un futuro trabajo de fin de máster.

Apéndice A

Anexo

A.1. Algoritmo Fitch-Hartigan

En el capítulo 2 introducimos el algoritmo Fitch-Hartigan para medir la parsimonia de cada árbol filogenético T. Aquí mostramos una explicación detalla de dicho algoritmo [AR12]:

1. Si T no tiene raíz, introducir arbitrariamente una, ρ, para conseguir el árbol con raíz T^ρ.
2. Asignamos a cada vértice $v \in V(T^\rho)$ un par $(U, \widehat{m})$ donde $U \subseteq S$ y $\widehat{m} \in \mathbb{N} \cup \{0\}$ como sigue:
   a) A cada hoja $v \in X$, le asignamos el par $(\chi(v), 0)$
   b) Si dos hijos de v se les ha asignado los pares $(U_1, \widehat{m_1})$ y $(U_2, \widehat{m_2})$ entonces asignamos a v el par:

$$(U, m) = \begin{cases} (U_1 \cup U_2, \widehat{m_1} + \widehat{m_2} + 1), & \text{si } U_1 \cap U_2 = \emptyset \\ (U_1 \cap U_2, \widehat{m_1} + \widehat{m_2}), & \text{en otro caso.} \end{cases}$$

   Repetir hasta que asignamos todos los pares.
3. Si el par (U, m) ha sido asignado a ρ, entonces $ps_\chi(T) = \widehat{m}$.
4. Repetimos este proceso hasta calcular ps_{χ_i} para todo χ_i, $i = 1, \ldots, m$. Entonces el total de parsimonia del árbol T será:

$$ps_{\{\chi_1, \ldots, \chi_m\}}(T) = \sum_{i=1}^{m} ps_{\chi_i}(T) = \sum_{i=1}^{m} \widehat{m}_i$$

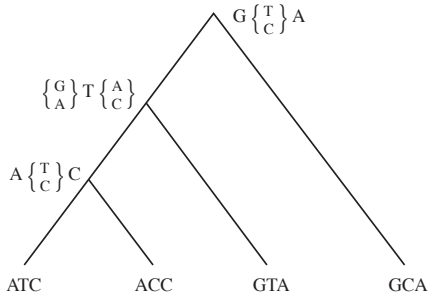

Figura A.1: Aplicación del algoritmo Fitch-Hartigan a un árbol de cuatro hojas.

La figura A.1 de la página anterior muestra un ejemplo del algoritmo de Fitch-Hartigan aplicado en este caso a un árbol de cuatro hojas.Veamos cómo calcular la parsimonia para el primer carácter. Asignamos a cada secuencia de izquierda a derecha $\widehat{m_i} = 0$, $i = 1,\ldots,4$. El vértice v_5 común a las secuencias *ATC* y *ACC* tiene $\widehat{m_5} = \widehat{m_1} + \widehat{m_2} + 0 = 0$ al tener ambas secuencias el primer carácter *A*. Ahora, el vértice v_6 con hijos v_5 y *GTA* tiene $\widehat{m_6} = \widehat{m_5} + \widehat{m_3} + 1 = 0 + 0 + 1$ ya que $\{A\} \cap \{G\} = \emptyset$. Por último, para la raíz, como $\{A,G\} \cap G \neq \emptyset$ tendrá una parsimonia de $ps_{\chi_1} = \widehat{m_6} + \widehat{m_4} + 0 = 1 + 0 + 0 = 1$. Reiterando para los caracteres restantes se tiene que la parsimonia total de este árbol es $ps_{\{\chi_1,\ldots,\chi_m\}}(T) = \sum_{i=1}^{4} ps_{\chi_i}(T) = 1 + 2 + 1 = 4$.

A.2. Detalles técnicos del análisis de datos.

En esta sección, se exponen brevemente los comandos de R utilizados para el estudio de los dos conjuntos de datos en el último capítulo.

A partir del conjunto de datos primates.dna calculamos la matriz de distancias, reconstruimos y representamos el árbol filogenético mediante los dos métodos de reconstrucción más usuales NJ y UPGMA:

```
> library(ape)
> library(phangorn)
> primates = read.phyDat("primates.dna", format="phylip", type="DNA")
> d = dist.dna(as.DNAbin(primates), model="JC69")
> d
> arbolUPGMA = upgma(d)
> plot(arbolUPGMA)
> arbolNJ = NJ(d)
> plot(arbolNJ, type="fan")
```

Notar que la matriz d tiene dimensión 14×14, presentamos sus valores más significativos :

$$\begin{array}{ll} d_{8,3} = 0{,}9905056 & d_{8,7} = 0{,}2676608 \\ d_{3,7} = 0{,}9282938 & d_{13,7} = 0{,}7439654 \\ d_{12,8} = 0{,}7557304 & d_{13,12} = 0{,}3255053 \\ d_{7,8} = 0{,}2676608 & d_{13,8} = 0{,}8174937 \\ & d_{7,12} = 0{,}5771917 \end{array}$$

Efectivamente, como dijimos en el capítulo tres no es ultramétrica ya que $d_{8,3} \not\leq \max(d_{8,7}, d_{3,7})$ ni tampoco aditiva pues $d_{13,7} + d_{12,8} \not\leq \max(d_{13,12} + d_{7,8}, d_{13,8} + d_{7,12})$. Cabe destacar de este código que la función dist.dna es muy completa y ofrece un gran abanico de posibilidades.

Algunos de sus argumentos más importantes son:

dist.dna(x, model, gamma, pairwise.deletion, base.freq, ...)

x Una matriz o una lista conteniendo las secuencias de ADN (Debe ser de clase DNAbin).

model Una cadena de caracteres especificando qué distancia evolutiva se usará: JC69 (Jukes-Cantor), K80 (Kimura), F81 (Felsenstein) entre los 17 disponibles.

gamma El valor del parámetro gamma usado para aplicar corrección a las distancias.

pairwise.deletion variable lógica que indica si eliminar aquellos lugares de la cadena en los que falta información.

base.freq vector de frecuencias de base usado en los cálculos (Si se puede introducir). Por defecto esta función las calcula a partir de el conjunto de secuencias.

Calculamos la parsimonia mediante la función parsimony de phangorn con el método de Fitch-Hartigan:

```
> parsimony(arbolUPGMA, primates, method="fitch")
[1] 751

> parsimony(arbolNJ, primates, method="fitch")
[1] 746

> optim.parsimony(arbolNJ, primates, method="fitch",rearrangements="NNI")

Final p-score 746 after 0 nni operations
Phylogenetic tree with 14 tips and 12 internal nodes.
Tip labels:
        Mouse, Bovine, Lemur, Tarsier, Squir Monk, Jpn Macaq, ...
Unrooted; no branch lengths.
```

Este último comando como mencionamos en el capítulo 3 aplica al árbol, en este caso el arbol obtenido mediante Neighbour-Joining (arbolNJ), el método heurístico Nearest Neighbour Interchange midiendo el grado de compatibilidad del árbol obtenido mediante el método de máxima parsimonia Fitch-Hartigan.

El paquete phangorn tiene distintas funciones que permiten no sólo ajustar el modelo evolutivo que se desee sino también calcular su verosimilitud. Centramos nuestra atención en las siguientes:

```
pml(tree, data, bf, Q, inv, k, shape, model, ...)

optim.pml(object, model, optNni, optBf, optQ, optInv, optGamma, optEdge,
   , control, subs , ...)
```

La función pml calcula el modelo evolutivo y su máxima verosimilitud dado el árbol filogenético y la secuencia de datos. La función optim.pml optimiza los diferentes parámetros del modelo. Explicamos algunos de los argumentos más relevantes de estas funciones:

object Un objeto de clase pml

k Número de intérvalos de la distribución gamma discreta.

shape Parámetro de forma de la distribución gamma α.

gamma El valor del parámetro gamma usado para aplciar corrección a las distancias.

inv Porcentaje de lugares que permanecen invariantes r.

model Al igual que en la función anterior, especifica el modelo que se va a aplicar.

opt(Nni, Bf, Q, Inv, Gamma, Edge) Optimiza la topología (volviendo a aplicar NNI), el vector de iniciación de probabilidades, la matriz de cambio instantáneo, los lugares invariantes, el parámetro de corrección gamm y las longitudes de los ejes respectivamente.

control, subs, maxit, ... Otros argumentos que permiten profundizar en la optimización del modelo ajustando el número de iteraciones o la manera de optimizar la matriz de cambio instantáneo.

Utilizando las funciones descritas y como hicimos en el último capítulo, ajustamos un primer modelo, Kimura 2-parámetros:

```
> fit = pml(arbolNJ, data=primates)
> fitk80 = optim.pml(fit, TRUE, model="K80")

optimize edge weights: −3074.938 −−> −3068.417
optimize rate matrix: −3068.417 −−> −2956.24
optimize edge weights: −2956.24 −−> −2953.216
optimize topology: −2953.216 −−> −2953.216
0
optimize rate matrix: −2953.216 −−> −2952.972
optimize edge weights: −2952.972 −−> −2952.946
optimize rate matrix: −2952.946 −−> −2952.943
optimize edge weights: −2952.943 −−> −2952.943
optimize rate matrix: −2952.943 −−> −2952.943
optimize edge weights: −2952.943 −−> −2952.943
optimize rate matrix: −2952.943 −−> −2952.943
optimize edge weights: −2952.943 −−> −2952.943

> fitk80

loglikelihood: −2952.943

unconstrained loglikelihood: −1230.335

Rate matrix:
      a        c        g        t
a 0.000000 1.000000 3.778059 1.000000
c 1.000000 0.000000 1.000000 3.778059
g 3.778059 1.000000 0.000000 1.000000
t 1.000000 3.778059 1.000000 0.000000

Base frequencies:
0.25 0.25 0.25 0.25
```

Ahora, ajustamos un modelo más general, el modelo General Time Reversible (GTR), optimizando, al igual que en el de Kimura, todos los parámetros. Debido a que esta optimización ocupa un gran número de líneas, presentamos un pequeño extracto de ésta:

```
> fitGTR = update(fit, k=4, inv=0.2)
> fitGTR = optim.pml(fitGTR, TRUE,TRUE, TRUE, TRUE, TRUE)

optimize topology: −2641.21 −−> −2641.2
optimize base frequencies: −2610.23 −−> −2610.225
optimize rate matrix: −2610.225 −−> −2609.886
optimize invariant sites: −2609.886 −−> −2609.886
optimize shape parameter: −2609.886 −−> −2609.878
optimize edge weights: −2609.878 −−> −2609.587

> fitGTR

 loglikelihood: −2609.587

unconstrained loglikelihood: −1230.335
```

```
Proportion of invariant sites: 0.006035243
Discrete gamma model
Number of rate categories: 4
Shape parameter: 3.173903

Rate matrix:
      a         c         g         t
a 0.0000000 0.6476364 33.6872234 0.4062758
c 0.6476364 0.0000000 0.0083054 14.3962899
g 33.6872234 0.0083054 0.0000000 1.0000000
t 0.4062758 14.3962899 1.0000000 0.0000000

Base frequencies:
0.3918023 0.3796106 0.04023627 0.1883509
```

Destacar por último que las funciones empleadas para la evaluación del modelo necesitan obligatoriamente argumentos de tipo pml. Presentamos también la lista completa de todos los modelos que se pueden obtener mediante la función modelTest.

```
> anova(fitk80, fitGTR)
 Likelihood Ratio Test Table

      Log lik.  Df  Df change  Diff log lik. Pr(>|Chi|)
1    -2952.9   26
2    -2609.6   35         9          686.71    < 2.2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

> SH.test(fitGTR, fitK80)
       Trees    ln L       Diff ln L   p-value
[1,]       1   -2609.586    0.0000    0.5002
[2,]       2   -2952.943  343.3569    0.0000

> AIC(fitK80)
[1] 5957.885
> AIC(fitGTR)
[1] 5289.171

> mt = modelTest(primates)
> mt
     Model df    logLik      AIC      BIC
1       JC 25 -3068.417 6186.834 6273.003
2     JC+I 26 -3062.628 6177.255 6266.870
3     JC+G 26 -3066.916 6185.832 6275.447
4   JC+G+I 27 -3062.642 6179.285 6272.347
5      F81 28 -2918.167 5892.333 5988.842
6    F81+I 29 -2909.121 5876.241 5976.196
7    F81+G 29 -2912.585 5883.170 5983.126
8  F81+G+I 30 -2908.519 5877.038 5980.440
9      K80 26 -2952.943 5957.885 6047.501
10   K80+I 27 -2944.508 5943.017 6036.079
11   K80+G 27 -2945.002 5944.003 6037.065
```

```
12 K80+G+I 28 -2942.382 5940.764 6037.272
13     HKY 29 -2647.767 5353.534 5453.489
14   HKY+I 30 -2629.834 5319.668 5423.070
15   HKY+G 30 -2618.512 5297.024 5400.426
16 HKY+G+I 31 -2615.152 5292.303 5399.152
17     SYM 30 -2813.914 5687.827 5791.229
18   SYM+I 31 -2811.727 5685.454 5792.303
19   SYM+G 31 -2804.771 5671.542 5778.391
20 SYM+G+I 32 -2804.680 5673.361 5783.656
21     GTR 33 -2642.946 5351.893 5465.635
22   GTR+I 34 -2624.036 5316.071 5433.260
23   GTR+G 34 -2613.669 5295.338 5412.527
24 GTR+G+I 35 -2610.313 5290.626 5411.262
```

Para finalizar este Anexo, presentamos los comandos utilizados en el segundo conjunto de datos geospiza:

```
> library(StatMatch)
> library(ape)
> library(phangorn)
> geodata<−read.table("geospiza.txt")
> medians = apply(geodata,2,median)     #Cálculo de las medianas de los rasgos
> mads = apply(geodata,2,mad)     #Cálculo de MAD (La media de la desviación típica)
> geodata.std<−as.matrix(geodata−outer(rep(1,13),medians)) %∗% diag(1/mads) #tipifiación robusta
> geo.eucl = dist(geodata.std)     #cálculo de la distancia euclídea
> geo.NJ = NJ(geo.eucl)     #cálculo del árbol NJ aplicado a la distancia euclídea
> geo.maha = mahalanobis.dist(geodata.std)     #cálculo de la distancia de Mahalanobis
> geo2.NJ = NJ(geo.maha)     #cálculo del árbol NJ aplicado a la distancia de Mahalanobis
> geotree <− read.nexus("geospiza.nex")     #árbol de consenso
> geotree<− drop.tip(geotree, "olivacea")     #eliminación del nodo olivacea, no perteneciente
 al conjunto de Geospiza
> dist.topo(geo.NJ, geotree)     #Comparamos ambos árboles mediante el número de ramas
[1] 18

> dist.topo(geo2.NJ, geotree)
[1] 16

> dist.topo(geo.NJ, geotree, method="score")     #Comparamos el árbol de distancia
euclídea con el de consenso
[1] 1.91005

> dist.topo(geo2.NJ, geotree, method="score")     #Comparamos el árbol de distancia
de Mahalanobis con el de consenso
[1] 1.025441
```

Bibliografía

[AR12] Elizabeth S. Allman, John A.Rhodes, *Lecture Notes: The Mathematics of Phylogenetics*, IAS/Park City Mathematics Institute , June-July, 2005, University of Alaska Fairbanks, Spring 2009, 2012.

[VSL10] Martin Vingron, Jens Stoye, Hannes Luz, *Lecture Notes: Algorithms for Phylogenetic Reconstructions*, Winter 2009/2010.

[JP04] Neil C.Jones, Pavel A. Pevzner, *An introduction to bioinformatics algorithms*, Massachusetts Institute of Technology, 2004.

[I04] Alexander Isaev, *Introduction to Mathematical Methods in Bioinformatics*, Universitext, Springer, 2004.

[EG01] Warren J.Ewens, Gregory R.Grant, *Statistical Methods in Bioinformatics*, Statistics for Biology and Health, Springer, 2001.

[P06] Emmanuel Paradis, *Analysis of Phylogenetics and Evolution with R*, Springer, 2006.

[C12] Marta Casanellas, *Técnicas algebraicas para la evolución de especies*, La Gaceta de la RSME, Vol. 15 (2012), Núm. 3, Págs. 521-536.

[SFBR03] Edward Susko, Chris Field, Christian Blouin, Andrew J.Roger , *Estimation of Rates-Across-Sites Distributions in Phylogenetic Substitution Models*,Department of Mathematics and Statistics, Canadian Institute for Advanced Research, Department of Biochemistry and Molecular Biology , Dalhousie University, Halifax, 594 - 603, 2003.

[STIC07] Victor Soria-Carrasco, Gerard Talavera, Javier Igea y Jose Castresana *The K tree score: quantification of differences in the relative branch length and topology of phylogenetic trees*, Department of Physiology and Molecular Biodiversity, Institute of Molecular Biology of Barcelona, CSIC, Barcelona 2007.

[NM87] Saitou, N., Nei, M, *The neighbor-joining method: a new method for reconstructing phylogenetic trees*, 1987.

[TC69] Jukes T, Cantor C, *Evolution of protein molecules*, Munro, H. N. (ed) Mamalian Protein Metabolism. Academic Press, New York, 1969.

[K80] Kimura M., *A simple method for estimating evolutionary rate in a finite population due to mutational production of neutral and nearly neutral base substitution through comparative studies of nucleotide sequences*, J.Mol. Biol., 16, 111-120, 1980.

[F81] Felsenstein J., *Evolutionary trees from DNA sequences: a maximum likelihood approach*, J.Mol. Evol., 17, 368-376, 1981.

[RP] R- Phylogenetics, http://www.r-phylo.org/

[W] Wikipedia The Free Encyclopedia, http://en.wikipedia.org/

[CR] The Comprehensive R Archive Network http://cran.r-project.org/

[IR] Inside R, a community Site for R, http://www.inside-r.org/

[UW] Department of Genome Science, University of Washington, http://evolution.genetics.washington.edu/