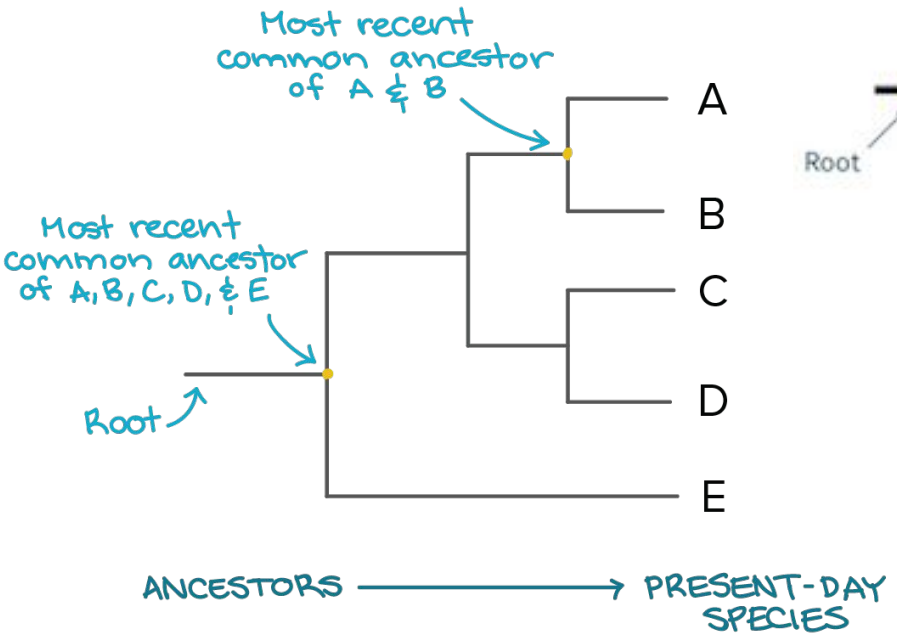
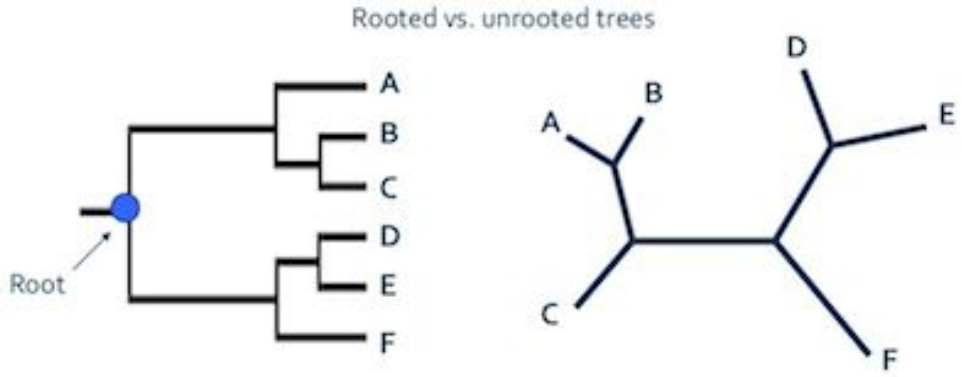
Comparación De Algoritmos Para La Generación De Árboles Filogenéticos

Adriana Michel Ávila García
Fernando Márquez Pérez

Pandas, ¿Osos o mapaches?

Árboles filogenéticos





Neighbor-Joining

Iniciar con un árbol en forma de estrella donde cada punta son las secuencias.

Repetir hasta que el árbol esté completado (ya no haya más pares para unir):

Calcular una matriz especial de distancias Q basada en las distancias de las secuencias recibidas.

Encontrar el par (f,g) distintos de la matriz con la distancia más corta y unirlas a través de un nuevo nodo u.

Conectar el nuevo nodo al nodo central.

Calcular las distancias de f y g a u y actualizar a Q.

Calcular las distancias de los demás nodos a u y actualizar a Q.

Funciones utilizadas

Matriz Q:

$$Q(i,j) = (n-2)d(i,j) - \sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k)$$

Distancias de f y g a u:

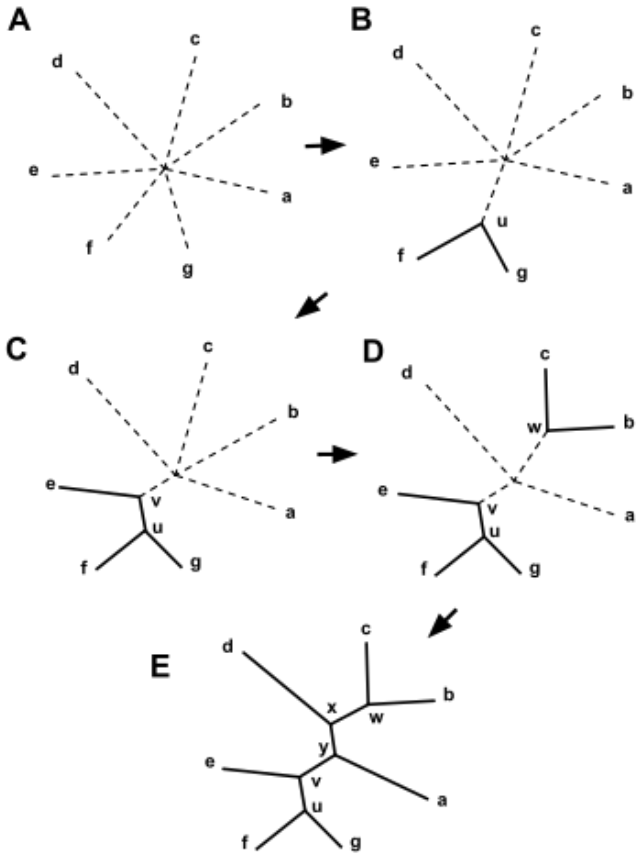
$$\delta(f,u) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}\left[\sum_{k=1}^{n} d(f,k) - \sum_{k=1}^{n} d(g,k)\right]$$

$$\delta(g,u) = d(f,g) - \delta(f,u)$$

Distancias de los demás vértices a u:

$$d(u,k) = \frac{1}{2}[d(f,k) + d(g,k) - d(f,g)]$$

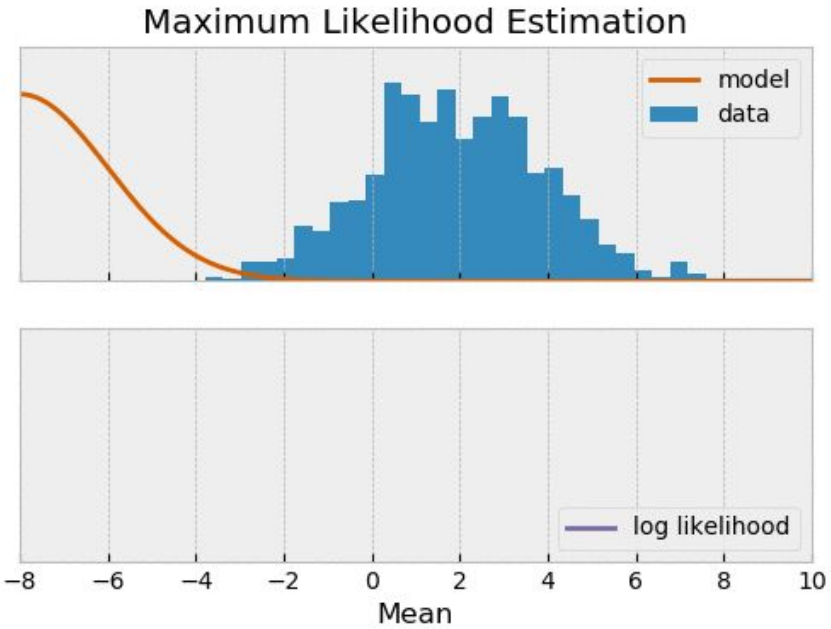
Ejemplo


A
d
c
b
e
a
f
g
B
d
c
b
e
a
u
f
g
C
d
c
b
e
v
a
u
f
g
D
d
c
w
b
e
v
a
u
f
g
E
d
c
x
b
w
y
e
v
a
u
f
g

Maximum likelihood

Algoritmo que busca maximizar el likelihood del árbol lo más posible.

Likelihood: La probabilidad de que el modelo actual generé las secuencias deseadas.



Algoritmo general

Tomamos valores iniciales para los parámetros de nuestro modelo, que incluyen pero no se limitan a: La topología del árbol, la longitud de las ramas, frecuencia de los nucleótidos, etc.

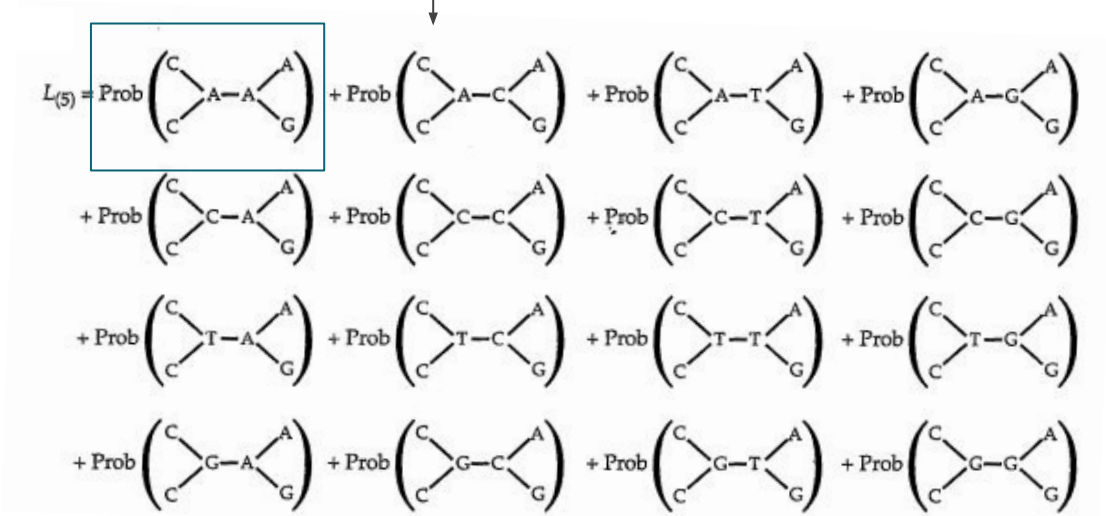
Obtenemos del modelo la matriz de probabilidades de que cualquier par de bases haya mutado (converga en el tiempo en el árbol).

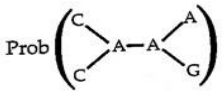
Con estas calculamos el likelihood del modelo como multiplicación de los likelihoods individuales de cada columna en el alineamiento, aunque usualmente se prefiere calcular la suma del logaritmo de estos valores para evitar valores tan pequeños.

Los likelihoods individuales se calculan como la suma de las probabilidades de las “historias” (que son los nucleótidos raíz que desconocemos para cada)

Al final alteramos ligeramente los valores de los parámetros y calculamos el likelihood hasta maximizarlo tanto como sea posible.

	1	2	3	4	5	6	7	8	9	...*n*
OTU1	A	A	G	A	C	T	T	C	A	...N
OTU2	A	G	C	C	C	T	T	C	T	...N
OTU3	A	G	A	T	A	T	C	C	A	...N
OTU4	A	G	A	G	G	T	C	C	T	...N





$= P_{Aeq} * P_{AC} * P_{AC} * P_{AA} * P_{AA} * P_{AG}$

Secuencias, alineamiento y árboles

Obtuvimos las secuencias de ADN y de aminoácidos para el gen del citocromo b de cada una de las especies mencionadas

Utilizamos el programa MEGA para realizar los alineamientos, por lo que usamos el algoritmo MUSCLE.

Una vez alineadas las secuencias, utilizamos la opción de MEGA para generar árboles filogenéticos a partir de estas.

Todos los árboles de aminoácidos se generaron con el modelo de Poisson y los de AND con el modelo evolutivo de Jukes-Cantor, que supone que la sustitución de una base por cualquier otra tiene la misma probabilidad.

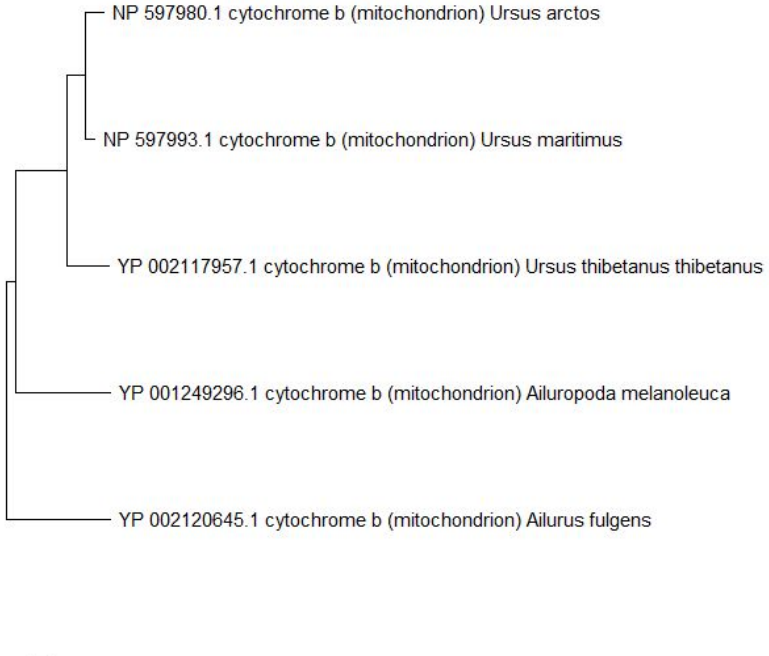
Métodos

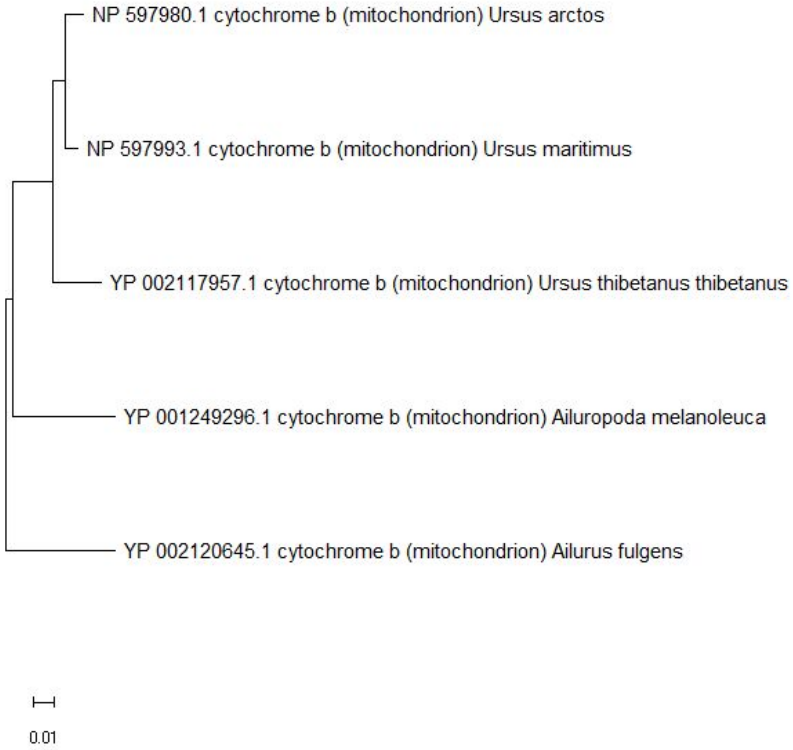
Decidimos comparar el gen cytochrome b de las siguientes especies:

- Panda gigante (Ailuropoda melanoleucapanda)
- Panda rojo (Ailurus fulgens)
- Oso pardo (Ursus arctos)
- Oso tibetano (Ursus thibetanus thibetanus)
- Oso polar (Ursus maritimus)

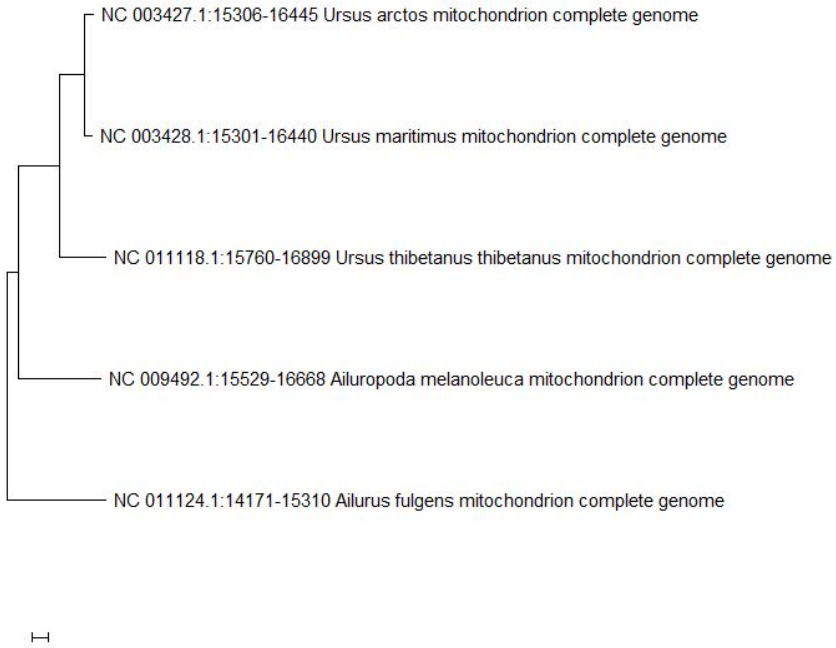
Cytochrome b es una proteína que se encuentra en la mitocondria de células eucariotas, y forma parte de la cadena de transporte de electrones.

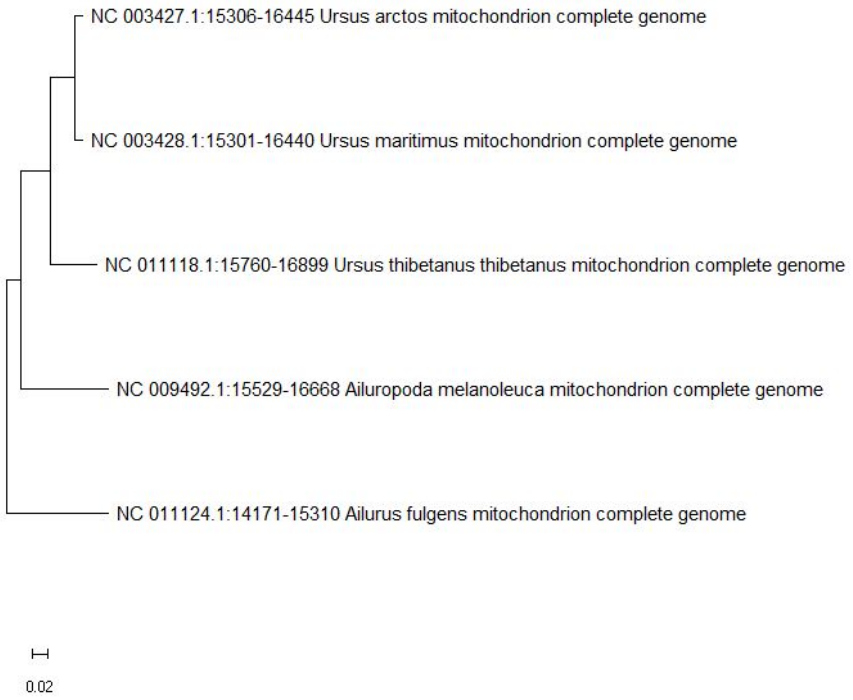
Resultados (aminoácidos)





Resultados (nucleótidos)





Conclusiones

En general notamos que todos los árboles generados presentan la misma estructura independiente del algoritmo utilizado para generarlos y de si son aminoácidos o nucleótidos. Era algo que podíamos esperar debido a la pequeña magnitud de secuencias y de tamaños que utilizamos. Todos los modelos cumplen con asociar a los pandas gigantes con la familia de los osos antes que con el panda rojo.

Si bien los resultados no lo reflejan, mencionamos que ambos algoritmos se utilizan en diferentes situaciones: Cuando hay muchos datos es preferible Neighbor-Joining y cuando se busca precisión es más común usar Maximum Likelihood.