Filogenética molecular (II)

Bioinformática, 29-3-17

Kevin Yip-CSE-CUHK
(Universidad china de Hong-Kong)

HOY …

1. Distancia evolutiva y modelos de mutación
2. Árboles: Las estructuras jerárquicas relacionando diferentes objetos biológicos
    1. Formatos de archivo
    2. reconstrucción de árboles filogenéticos
    3. Métodos basados en secuencias
        - **máxima parsimonia**
        - Máxima verosimilitud
    4. métodos basados en distancias
        - UPGMA
        - Unión de vecinos

máxima parsimonia

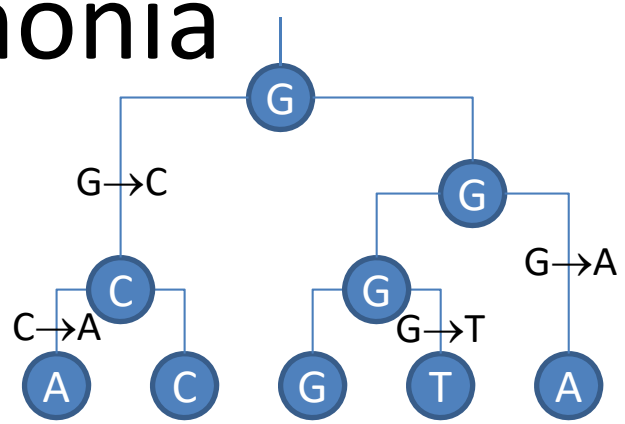
- Suponemos: Un árbol es probable que sea correcto **si implica pocas mutaciones**
- Razón fundamental:
  - Las mutaciones son poco frecuentes
  - "Navaja de Occam": **La explicación más simple es probablem. la correcta**
- Problema gral.:
  - Dado un conjunto de secuencias, encontrar una topología de árbol con raíz y las secuencias ancestrales del árbol de forma que el número total de mutaciones en el árbol sea mínimo
  - NP duro: no hay algoritmos en tiempo polinómico

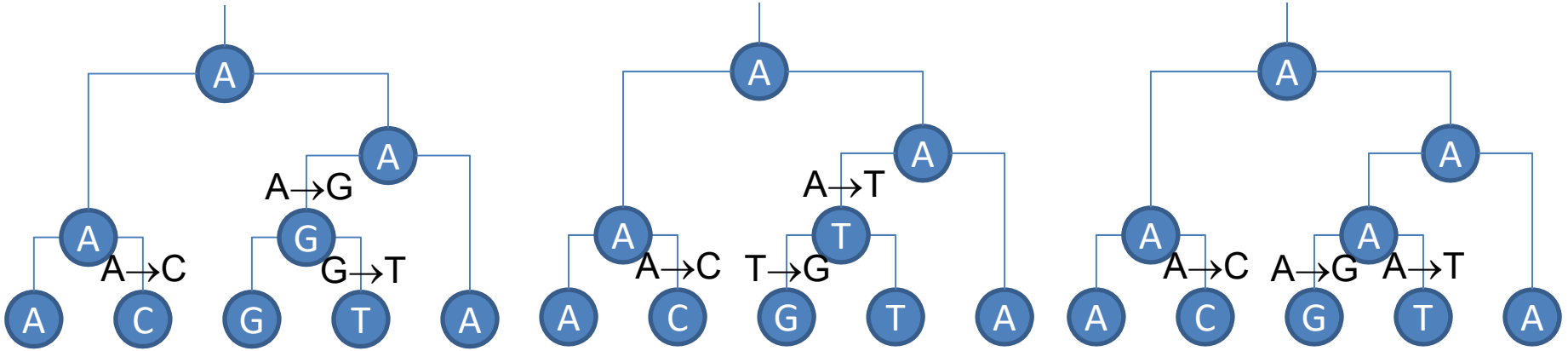
máxima parsimonia

- Problema restringido:
  - **Dado** un conjunto de secuencias y una **topología de árbol** con raíz
  - Encontrar las secuencias ancestrales del árbol de forma que el número total de mutaciones en el árbol sea mínimo
- Ahora nos centraremos en el problema restringido

Ejemplo de parsimonia

- Vamos a considerar una sola posición
  - Asumiendo que las posiciones son independientes, sólo necesitamos un algoritmo para una
  - Veremos un ejemplo con más posiciones
- En el árbol de la derecha, el número de mutaciones es 4
  - ¿Es el mínimo (es decir, la solución más parsimoniosa)?
  - Para esta topología del árbol, el número mínimo de mutaciones es 3. Hay tres conjuntos de estados ancestrales que resultan en este número de mutaciones, que se muestran en los tres árboles de debajo



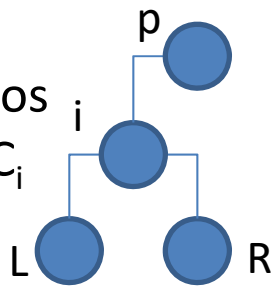


problema de parsimonia

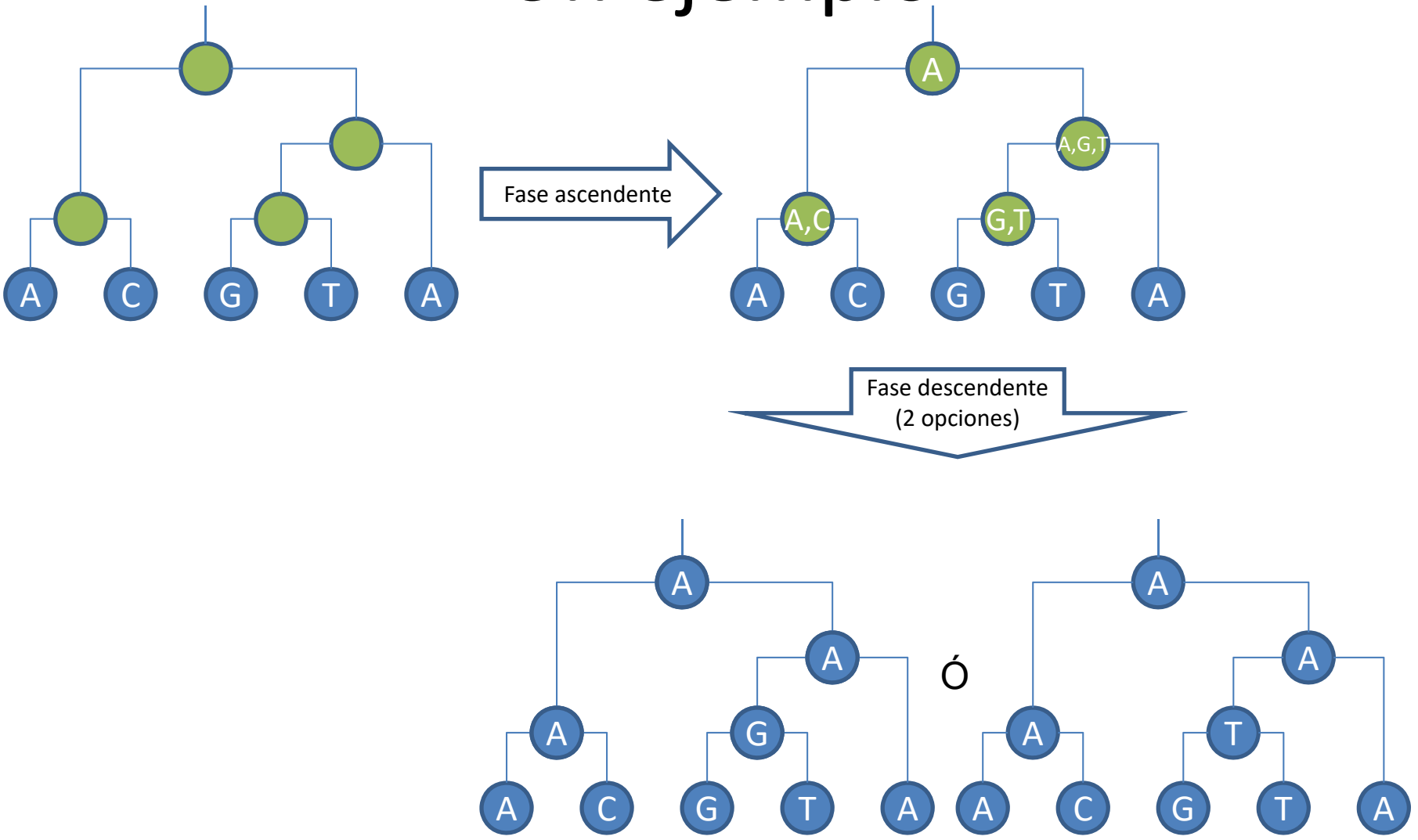
- ¿Cómo asignar estados ancestrales para minimizar el número total de mutaciones?
- Ideas: dado un nodo,
  - Si ambos hijos tienen el mismo estado, probablemente es bueno adoptar ese estado
  - Si los hijos tienen dos estados diferentes, probablemente es bueno adoptar uno de ellos
  - Retrasar la decisión de la elección exacta hasta que el padre también ha expresado una preferencia

El algoritmo: versión simple

- El algoritmo de Fitch: Si sólo se necesita una solución
  - Para cada nodo interno i con los padres y los hijos p, L y R, vamos a determinar su conjunto de preferencias S_i y su carácter final C_i que reduzca al mínimo el número total de mutaciones
  - Pasos:
    1. Para cada nodo hoja i, S_i es el carácter de la hoja i
    2. fase ascendente: Para cada i nodo interno,
       Si $(S_L \cap S_R) = \{\}$ // L y R no están de acuerdo: coger ambos
       $S_i := S_L \cup S_R$
       else // L y R están de acuerdo en algo: tomar el acuerdo
       $S_i := S_L \cap S_R$
    3. fase descendente: En primer lugar elegir cualquier $C_{raíz}$ en $S_{raíz}$.
       Luego, para cada i otro nodo interno,
       si $C_p \in S_i$ // p está de acuerdo con i en algo: cogerlo
       $C_i := C_p$
       else // p no está de acuerdo con i: usar las preferencias de i
       $C_i :=$ Elegir uno de S_i



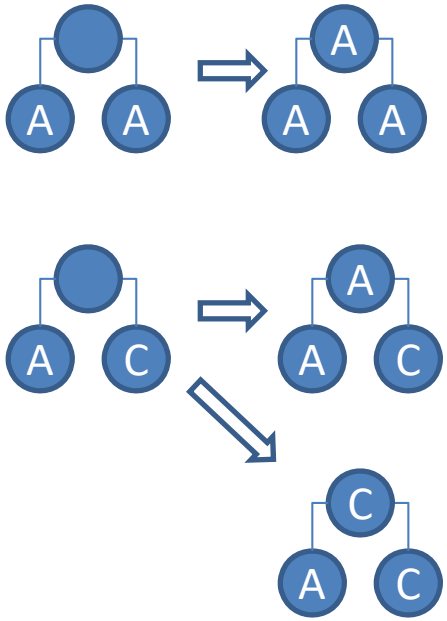
Un ejemplo


conjunto de preferencias
carácter final elegido
A
C
G
T
A
Fase ascendente
A
A,G,T
A,C
G,T
A
C
G
T
A
Fase descendente
(2 opciones)
A
A
A
G
A
C
G
T
A
Ó
A
A
A
T
A
C
G
T
A

¿Por qué funciona?

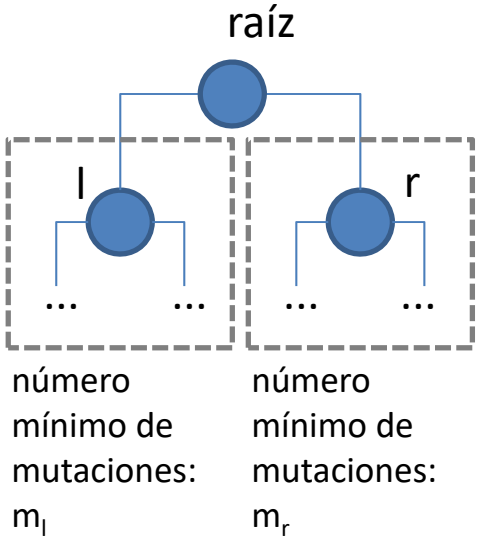
- Demostración por inducción
  - Cuando hay dos hojas, sólo hay dos casos:
    - Tienen el mismo carácter
      - número mínimo real de mutaciones: 0
      - El algoritmo da el mismo número
    - Tienen diferentes caracteres
      - número mínimo de mutaciones en: 1
      - El algoritmo también da el mismo número

    Por lo tanto el algoritmo es óptimo

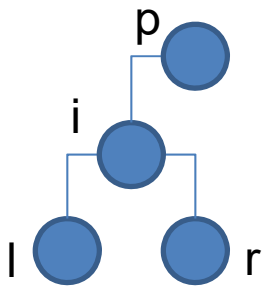


¿Por qué funciona?

- Supongamos que el algoritmo es capaz de minimizar el número de mutaciones para árboles con k o menos hojas
- Ahora, por un árbol con hojas k + 1,
  - Se compone de una raíz conectado a dos sub-árboles con raíces l y r, ambos con k o menos hojas
  - Dos casos:
    - Si $S_l \cap S_r \neq \{\}$, El algoritmo da una solución con $m_l + m_r$ mutaciones, que es óptima debido a la hipótesis de inducción
    - Si $S_l \cap S_r = \{\}$, El algoritmo da una solución con $m_l + m_r + 1$ mutaciones, que también es óptima ya que una mutación adicional debe ser introducida entre la raíz y uno de sus hijos

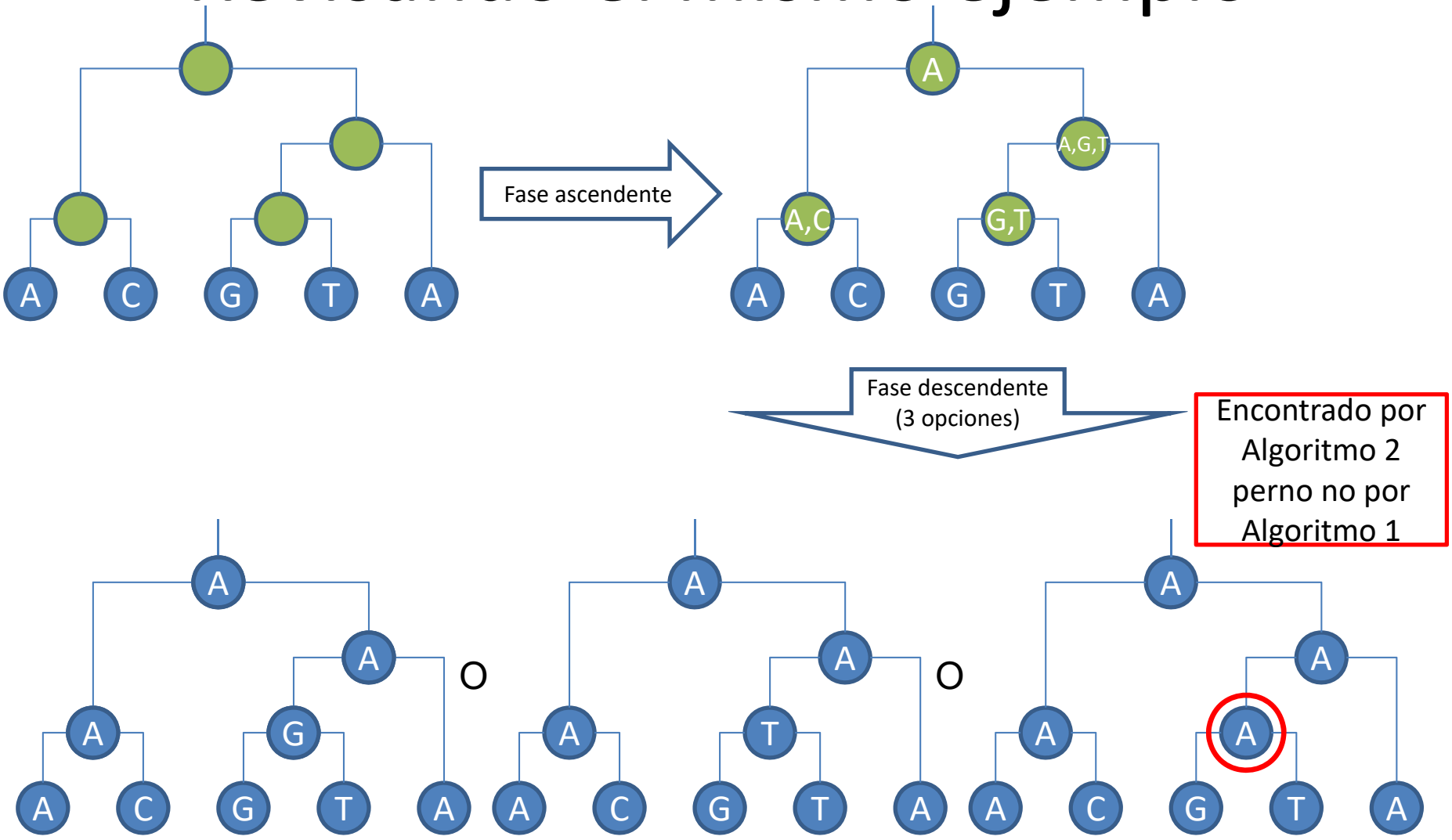


El algoritmo: versión extendida

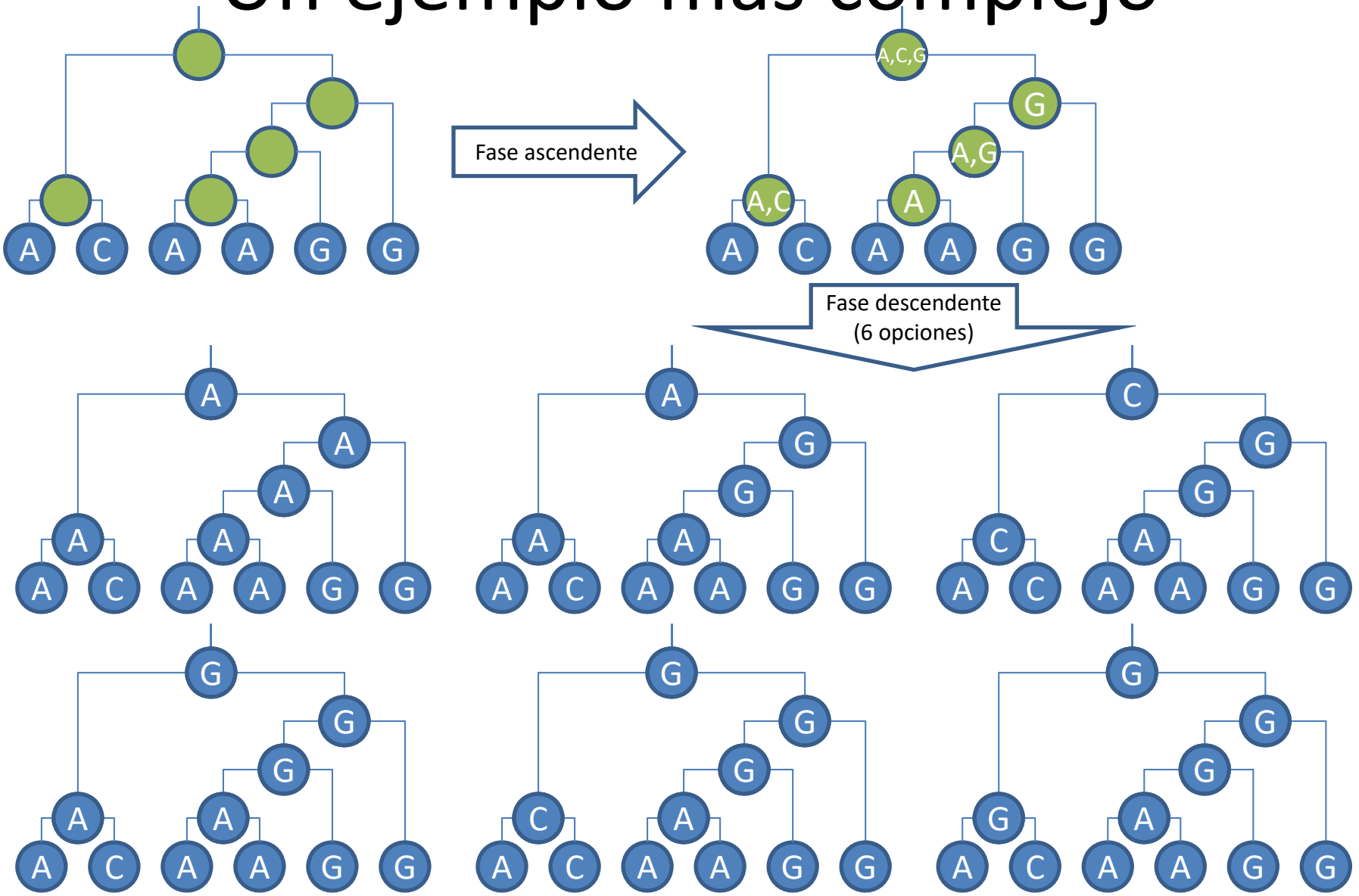


- Si necesita todas las soluciones
  - Pasos:
    1. Para cada nodo hoja i, S_i es el carácter de la hoja
    2. fase ascendente (igual que antes): Para cada nodo interno i,
       Si ($S_l \cap S_r$) = {} // L y R no están de acuerdo: hay que tomar ambos conjuntos
       $S_i := S_l \cup S_r$
       else // L y R están de acuerdo en algo: tomarlo
       $S_i := S_l \cap S_r$
    3. fase descendente: Primera selección $C_{raíz}$ desde $S_{raíz}$. Luego, para cada i otro nodo interno **(Diferente estrategia - voto mayoritario): elegiremos C_i a partir de los caracteres que existen en el mayor número de conjuntos entre {C_p}, S_l y S_r. Además, cada vez que hay múltiples opciones, elegimos una cada vez de para enumerar todas las soluciones óptimas.**
  - Podemos demostrar que este algoritmo da todas las soluciones óptimas
  - Un caso especial de algoritmo de programación dinámica

Revisando el mismo ejemplo



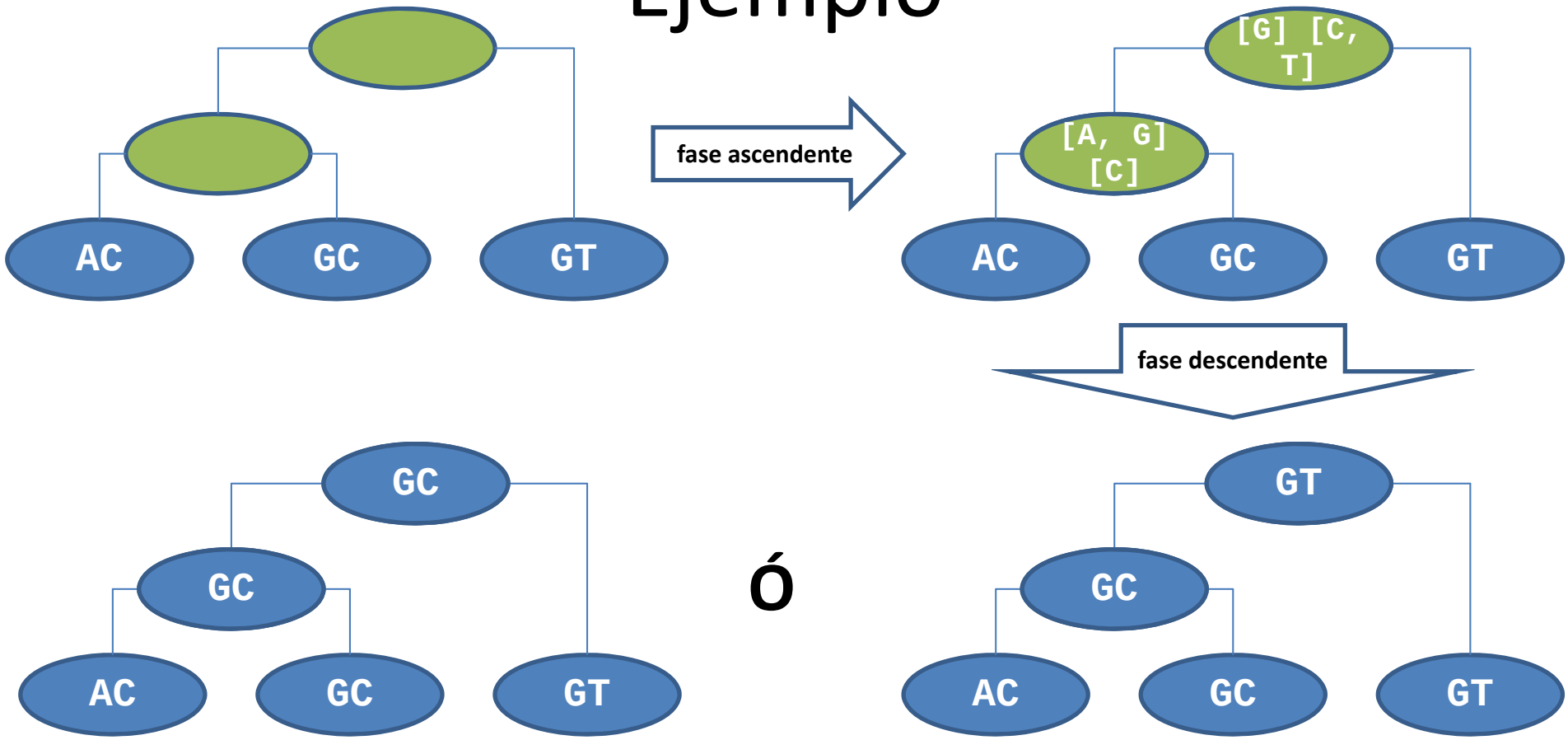
Un ejemplo más complejo



múltiples posiciones

- En una situación real, tenemos que hacer frente a secuencias que contienen más de una posición
- Simplemente aplicamos el algoritmo anterior a las diferentes posiciones de forma independiente
  - Es como suponer que posiciones diferentes mutan de forma independiente

Ejemplo



- Mínimo: 1 sustitución para la posición 1, 1 sustitución de la posición 2
- máxima parsimonia: 2 árboles que pueden alcanzar este mínimo

HOY …

1. Distancia evolutiva y modelos de mutación
2. Árboles: Las estructuras jerárquicas relacionando diferentes objetos biológicos
    1. Formatos de archivo
    2. reconstrucción de árboles filogenéticos
    3. Métodos basados en secuencias
        - máxima parsimonia
        - **Máxima verosimilitud**
    4. métodos basados en distancias
        - UPGMA
        - Unión de vecinos

Máxima verosimilitud

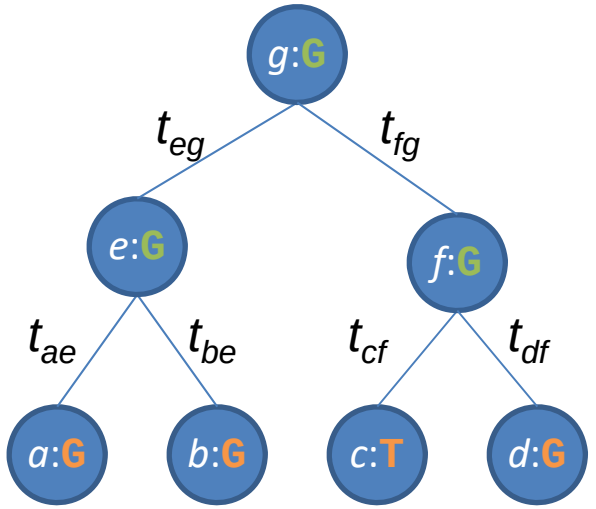
- Probabilidad: Probabilidad de producir los datos observados por un modelo determinado los parámetros del modelo, Pr ($x|\theta$)
  - x: Datos observados
    - Las secuencias de entrada, que se suponen alineadas
    - Una vez más, consideramos una sola posición aquí. La probabilidad para el conjunto de las secuencias es el producto de la probabilidad de las posiciones individuales, ya que se suponen independientes
  - θ: Los parámetros del modelo (ver página siguiente)
- máxima verosimilitud: Encontrar el valor de θ tal que Pr ($x|\ \theta$) se maximiza

Los parámetros del modelo

- Existen diferentes posibilidades
  - En todos los casos, *x* son las secuencias de entrada
- Gran problema de verosimilitud
  - θ: topología de árbol, las tasas de mutación y los tiempos de divergencia
  - Muy difícil
- Pequeño problema de verosimilitud
  - Dada la topología de árbol
  - θ: las tasas de mutación y los tiempos de divergencia
  - Hay soluciones heurísticas eficaces que por lo general (pero no siempre) producen resultados óptimos

el cálculo de la verosimilitud

- Supongamos que se nos da lo siguiente, como se muestra en la figura:
  - topología de árbol
  - Los datos observados, x = {a:G,b:G,c:T,d:G}
  - secuencias ancestrales
  - Los parámetros, θ = {<tasas de mutación>, t_{ae},t_{be},t_{cf},t_{df},t_{eg},t_{fg}}
- Probabilidad =Pr(g:G) Pr(e:G|g:G, t_{eg}) Pr(f:G|g:G, t_{fg}) Pr(a:G|e:G, t_{ae}) Pr(b:G|e:G, t_{be}) Pr(c:T|f:G, t_{cf}) Pr(d:G|f:G, t_{df})
  - Hemos aprendido cómo calcular estas probabilidades condicionales para Jukes-Cantor



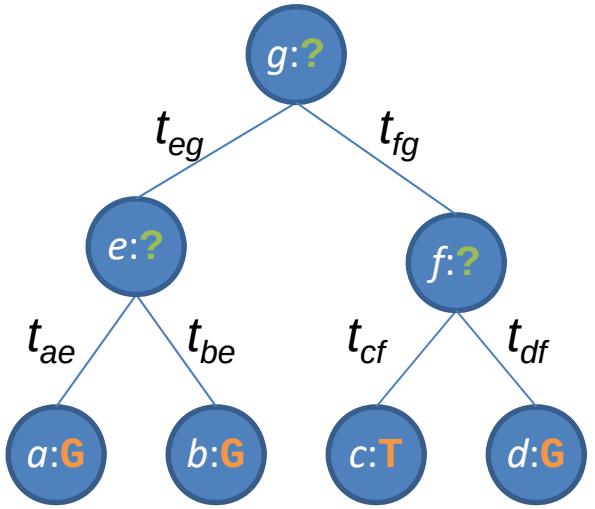
etiquetas de nodo
secuencias observadas
secuencias ancestrales

los tiempos de divergencia

el cálculo de la verosimilitud

- En el pequeño problema de verosimilitud, que sólo da la topología del árbol, pero no los estados ancestrales
- Necesidad de probarlos todos (Suma de $4^3 = 64$ términos) :probabilidad =

Pr(g:A)
Pr(e:A|g:A, t_{eg}) Pr(f:A|g:A, t_{fg})
Pr(a:G|e:A, t_{ae}) Pr(b:G|e:A, t_{be})
Pr(c:T|f:A, t_{cf}) Pr(d:G|f:A, t_{df})
+
Pr(g:C)
Pr(e:A|g:C, t_{eg}) Pr(f:A|g:C, t_{fg})
Pr(a:G|e:A, t_{ae}) Pr(b:G|e:A, t_{be})
Pr(c:T|f:A, t_{cf}) Pr(d:G|f:A, t_{df})
+
...
+
Pr(g:T)
Pr(e:T|g:T, t_{eg}) Pr(f:T|g:T, t_{fg})
Pr(a:G|e:T, t_{ae}) Pr(b:G|e:T, t_{be})
Pr(c:T|f:T, t_{cf}) Pr(d:G|f:T, t_{df})



Posibles estados ancestrales:

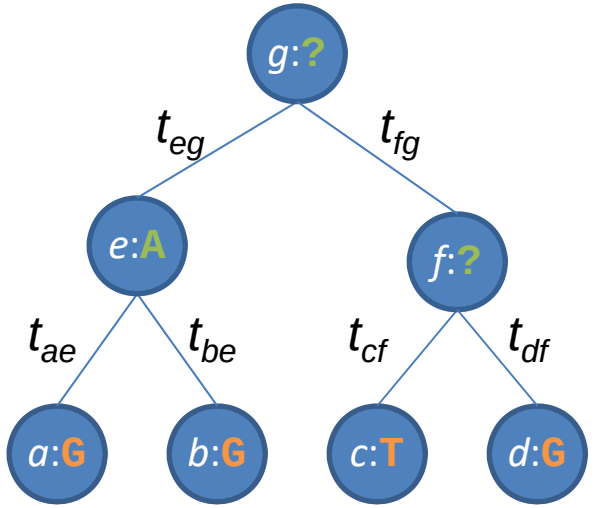
e	A	A	A	A	A		T
f	A	A	A	A	C	...	T
g	A	C	G	T	A		T

eficiencia computacional

- ¿Tiempo de cálculo necesario?
- Para nuestro ejemplo:
  - 3 nodos internos $\Rightarrow 4^3 = 64$ posibles conjuntos de estados ancestrales
  - Para cada conjunto de estados ancestrales, necesitamos multiplicar 7 términos (porque hay 7 nodos en el árbol)
- En general:
  - Si hay n secuencias de entrada, hay n-1 Nodos internos $\Rightarrow 4^{n-1}$ posibles conjuntos de estados ancestrales
  - Para cada conjunto de estados ancestrales, necesitamos multiplicar $n+n-1 = 2n-1$ términos
- Es poco práctico llevar a cabo este número exponencial de operaciones - A continuación, la forma de resolver el problema …
  - ¡Programación dinámica!

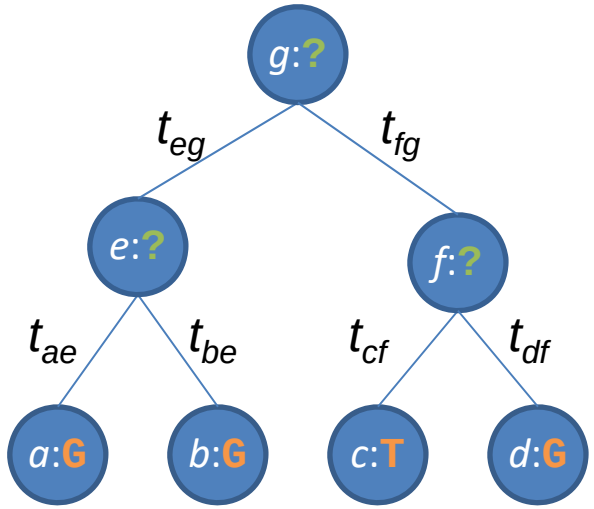
El cálculo de la verosimilitud de manera eficiente

- Una observación importante: una vez que se determina la raíz de un sub-árbol, la probabilidad de este sub-árbol no depende de otros nodos en el árbol entero
- Por ejemplo, una vez que el nodo *e* se decide a tomar carácter A, La verosimilitud de el sub-árbol de nodos *a*,*b* y *e* es $\Pr(e{:}\mathrm{A}|g, t_{eg})$ $\Pr(a{:}\mathrm{G}|e{:}\mathrm{A}, t_{ae})\Pr(b{:}\mathrm{G}|e{:}\mathrm{A}, t_{be})$
  - Si el carácter en el nodo *g* no cambia, el valor de la expresión anterior no cambiará sin importar qué carácter toma *f*
  - Por lo tanto este valor puede ser reutilizado



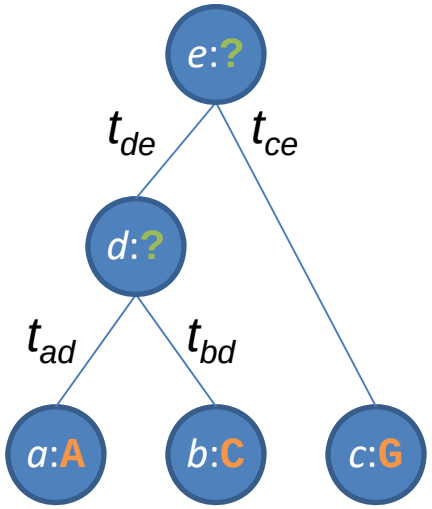
El cálculo de la verosimilitud de manera eficiente

- Definir la tabla V, donde la entrada V (*i*,*c*) es la verosimilitud del subárbol con raíz *i* cuando el **padre** de *i* tiene carácter *c*
  - verosimilitud =
    Pr(*g*:A) V(*e*,A) V(*f*,A) +
    Pr(*g*:C) V(*e*,C) V(*f*,C) +
    Pr(*g*:G) V(*e*,G) V(*f*,G) +
    Pr(*g*:T) V(*e*,T) V(*f*,T)
  - V(*e*, A) =
    $\Pr(e{:}A \mid g{:}A, t_{eg})\ V(a,A)\ V(b,A) +$
    $\Pr(e{:}C \mid g{:}A, t_{eg})\ V(a,C)\ V(b,C) +$
    $\Pr(e{:}G \mid g{:}A, t_{eg})\ V(a,G)\ V(b,G) +$
    $\Pr(e{:}T \mid g{:}A, t_{eg})\ V(a,T)\ V(b,T)$
  - V(*a*,A) = $\Pr(a{:}G \mid e{:}A, t_{ae})$
  - V(*a*,C) = $\Pr(a{:}G \mid e{:}C, t_{ae})$
  - ...



- La tabla V contiene O (*n*) entradas. Calcular el valor de cada entrada requiere un número constante de operaciones $\Rightarrow$ tiempo lineal

Ejemplo



- Asumamos:
  - Las cuatro bases son igualmente probables en la raíz
  - La mutación modelo de Jukes-Cantor es correcta
  - tasa de mutación por unidad de tiempo, α = 0,1
  - Cada rama del árbol representa una unidad de tiempo
- V (*i*,*x*) es la verosimilitud del subárbol con raíz *i* cuando el **padre** de *i* tiene carácter *x*

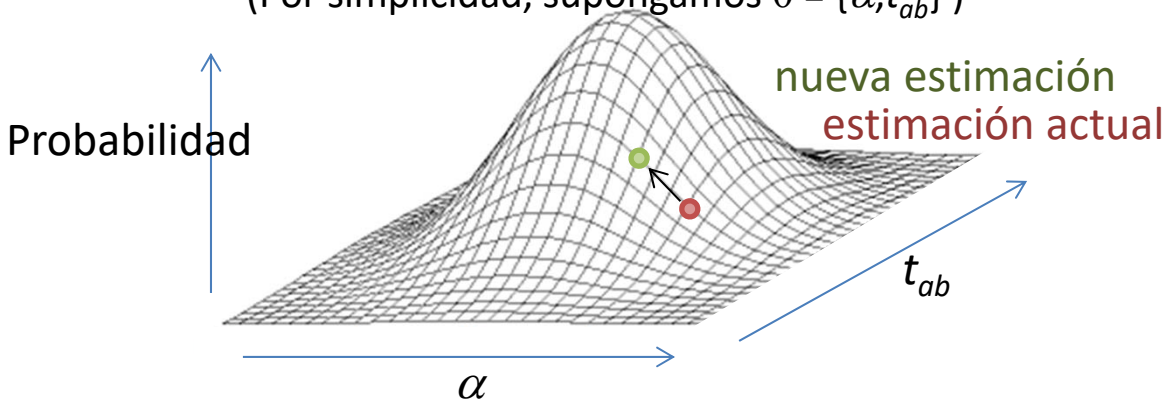
V(*i*, *x*)	*x*=A	*x*=C	*x*=G	*x*=T
i=*a*	0.7	0.1	0.1	0.1
i=*b*	0.1	0.7	0.1	0.1
i=*c*	0.1	0.1	0.7	0.1
i=*d*	0.7(0.7)(0.1) + 0.1(0.1)(0.7) + 0.1(0.1)(0.1) + 0.1(0.1)(0.1) = 0.058	0.1(0.7)(0.1) + 0.7(0.1)(0.7) + 0.1(0.1)(0.1) + 0.1(0.1)(0.1) = 0.058	0.1(0.7)(0.1) + 0.1(0.1)(0.7) + 0.7(0.1)(0.1) + 0.1(0.1)(0.1) = 0.022	0.1(0.7)(0.1) + 0.1(0.1)(0.7) + 0.1(0.1)(0.1) + 0.7(0.1)(0.1) = 0.022

- verosimilitud total: 0,25 (0,058) (0,1) + 0,25 (0,058) (0,1) + 0,25 (0,022) (0,7) + 0,25 (0,022) (0,1) = 0,0073

La solución del problema pequeño

- Entonces, ¿cómo encontrar los valores óptimos de los parámetros?
  - Comience con una estimación al azar θ
  - Aplicar un algoritmo “hill climbing"
    - Cambiar el valor de un parámetro de manera que se incremente la probabilidad
    - Repetirlo para cada parámetro, a su vez, por múltiples iteraciones
    - Alcanzará máximo si hay un solo "pico" - Esto es cierto en muchas situaciones reales, aunque se pueden construir casos en los que no es cierto

(Por simplicidad, supongamos $\theta = \{\alpha, t_{ab}\}$)



Fuente de la imagen: http://www.absoluteastronomy.com/topics/Hill_climbing

HOY …

1. Distancia evolutiva y modelos de mutación
2. Árboles: Las estructuras jerárquicas relacionando diferentes objetos biológicos
    1. Formatos de archivo
    2. reconstrucción de árboles filogenéticos
    3. Métodos basados en secuencias
        - máxima parsimonia
        - Máxima verosimilitud
    4. **métodos basados en distancias**
        - UPGMA
        - Unión de vecinos

Motivación

- En los anteriores algoritmos basados en secuencias, las secuencias exactas se utilizan en la reconstrucción de los árboles filogenéticos
- En un método basado en distancia, sólo se consideran las distancias entre pares de secuencias
  - Bueno si las secuencias son largas, y sólo nos preocupa la estructura de árbol, pero no las secuencias ancestrales
  - Las distancias se pueden calcular por métodos basados en la alineación de secuencias (pero hay distancias calculables sin alineamiento)
  - Una vez que las distancias por pares se han calculado, no se utilizarán las secuencias originales

UPGMA

- Unweighted Pair Group Method with Arithmetic Mean
- Algoritmo:
  1. Calcular la distancia entre cada par de secuencias
  2. Tratar a cada secuencia como un grupo por sí mismo
  3. Combinar los dos grupos más cercanos. La distancia entre dos grupos es la distancia media entre todas sus secuencias (excepto que d (C_i, C_i) = 0):

     $$\mathrm{d}(\mathrm{C_i}, \mathrm{C_j}) = \frac{1}{|\mathrm{C_i}||\mathrm{C_j}|} \sum_{r \in \mathrm{C_i}, s \in \mathrm{C_j}} \mathrm{d}(r, s)$$

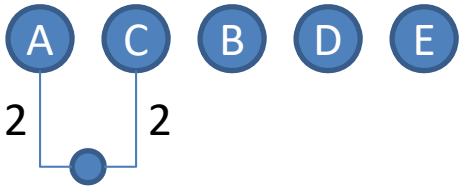
     (Nótese que d (*r*,*s*) es la distancia entre r y s en la matriz de distancia de entrada)
  4. Repetir 2 y 3 hasta que sólo quede un cluster

Ejemplo

	A	B	C	D	E
A	0	8	4	6	8
B	8	0	8	8	4
C	4	8	0	6	8
D	6	8	6	0	8
E	8	4	8	8	0

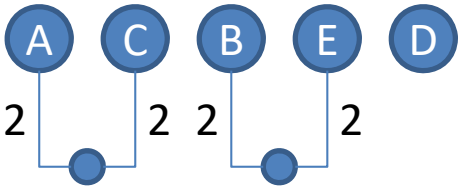
{A},{C}

	A,C	B	D	E
A,C	0	8	6	8
B	8	0	8	4
D	6	8	0	8
E	8	4	8	0



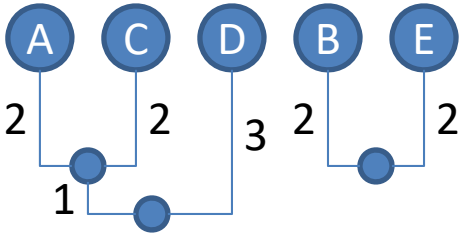
{B},{E}

	A,C	B,E	D
A,C	0	8	6
B,E	8	0	8
D	6	8	0



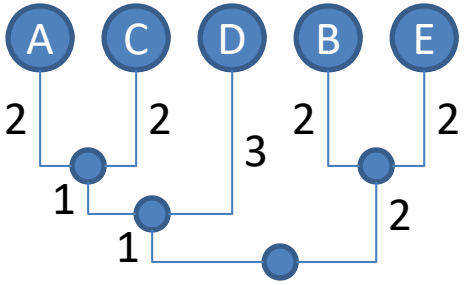
{A,C}, {D}

	A,C,D	B,E
A,C,D	0	8
B,E	8	0



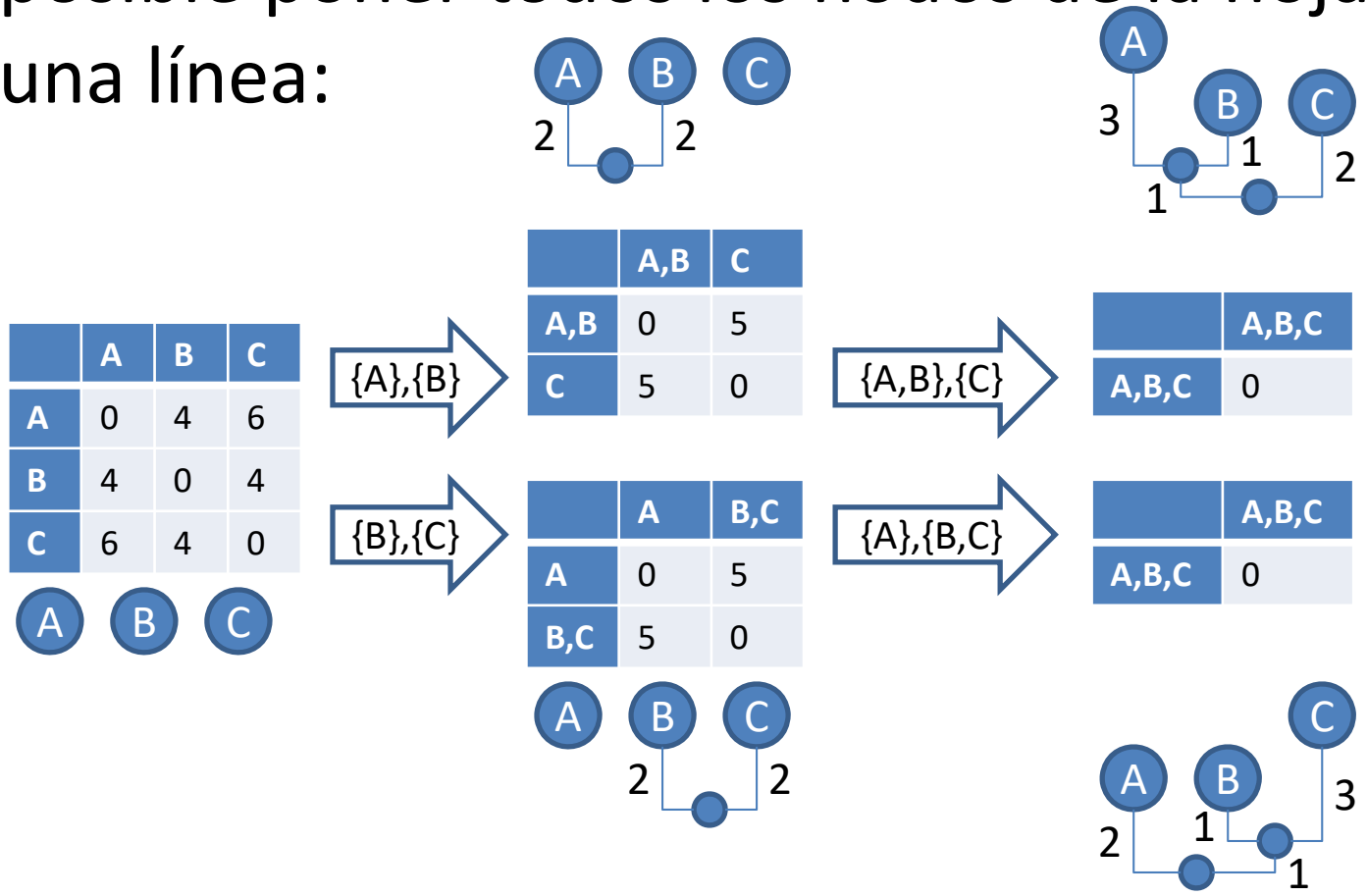
{A,C,D}, {B,E}

	A,B,C,D,E
A,B,C,D,E	0



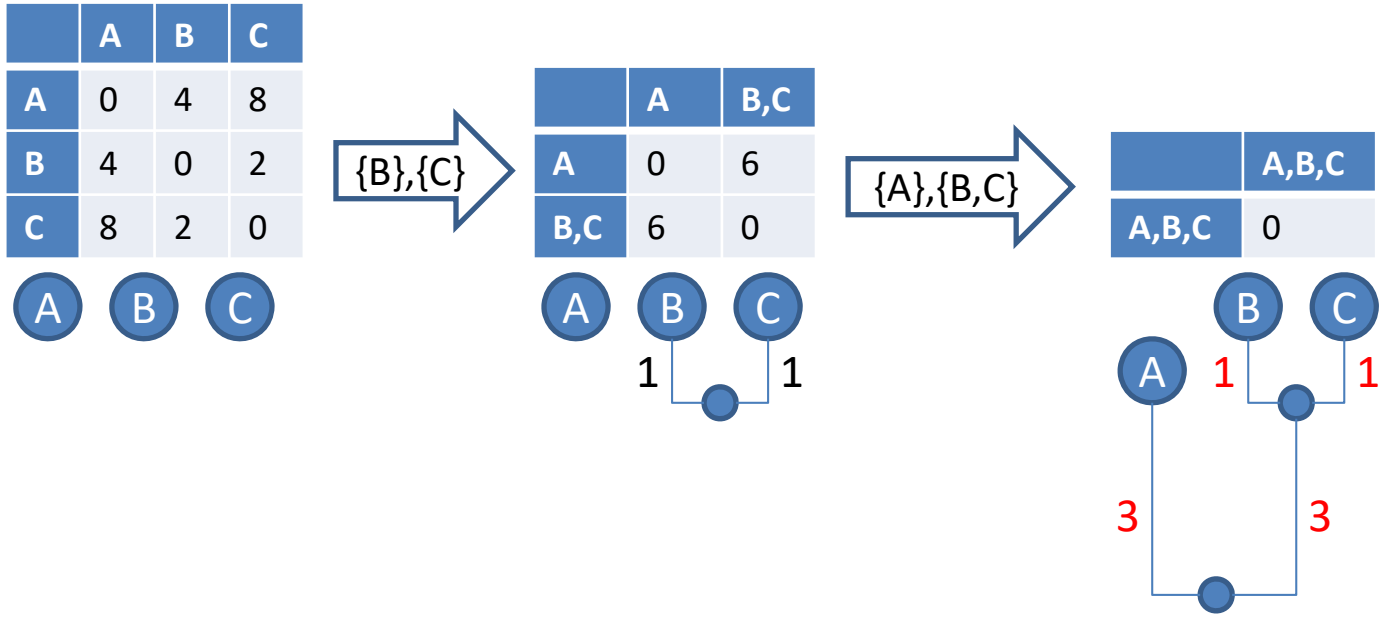
Unicidad

- No siempre es único, también no siempre es posible poner todos los nodos de la hoja en una línea:



Longitudes de rama

- No siempre es posible asignar longitudes de rama de acuerdo a las distancias:



Aquí las brach lengths sólo reflejan las distancias de clúster, no las distancias de secuencia

HOY …

1. Distancia evolutiva y modelos de mutación
2. Árboles: Las estructuras jerárquicas relacionando diferentes objetos biológicos
    1. Formatos de archivo
    2. reconstrucción de árboles filogenéticos
    3. Métodos basados en secuencias
        - máxima parsimonia
        - Máxima verosimilitud
    4. métodos basados en distancia
        - UPGMA
        - **Unión de vecinos**

Unión de vecinos

- En UPGMA, cada vez que se fusionan los dos grupos más cercanos según su distancia (criterio nº 1):

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{r \in C_i, s \in C_j} d(r, s)$$

- Sería bueno elegir el grupo que también está muy lejos de otros grupos (criterio nº 2), medidos por:

$$u(C_i) = \sum_j d(C_i, C_j)$$

- En el algoritmo unión de vecinos, los dos grupos que fusionar es el par que minimiza

$Q(i,j) = (r-2)d(C_i, C_j) - u(C_i) - u(C_j)$, donde r es el número actual de grupos (y Q (i, i) ≡ 0 para todo i)

  - La fórmula considera ambos criterios, mientras que el factor (r-2) es para equilibrar sus pesos relativos

Unión de vecinos

- El algoritmo:
  1. Comienza con cada secuencia como un clúster. Todos ellos están conectados a un centro, formando una estrella.
  2. Encontrar grupos i y j conectados al centro donde Q (i, j) es mínimo entre todos los pares de cluster
  3. Insertar un nuevo nodo interno C_k
     - Conectarlo a C_i, C_j y el centro
     - Asignar la longitud $\frac{d(C_i, C_j)}{2} + \frac{u(C_i) - u(C_i)}{2(r-2)}$ a C_iC_k
     - Asignar la longitud $\frac{d(C_i, C_j)}{2} + \frac{u(C_i) - u(C_i)}{2(r-2)}$ a C_jC_k
     - Para cada nodo C_l, $d(C_k, C_l) = [d(C_i, C_l) + d(C_j, C_l) - d(C_i, C_j)] / 2$ (todos los $d(C_x, C_y)$ son valores ya calculados).
  4. Repetir 2 y 3 hasta que todas las longitudes de rama se asignen
     - El resultado final será un árbol sin raíces

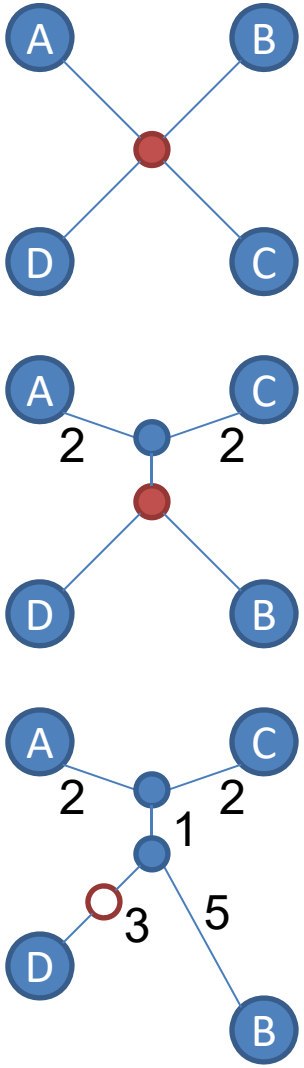
Ejemplo

$Q(i,j) = (r-2)d(C_i,C_j) - u(C_i) - u(C$

Distancia entre A y el nuevo nodo: d(A,C)/2 + [u(A) – u(C)] / [2(r-2)] = 4/2 + (18-18) / [2(2)] = 2

{A}, {C}

{A,C}, {B}



d	A	B	C	D
A	0	8	4	6
B	8	0	8	8
C	4	8	0	6
D	6	8	6	0

u	
A	18
B	24
C	18
D	20

Q	A	B	C	D
A	0	-26	-28	-26
B	-26	0	-26	-28
C	-28	-26	0	-26
D	-26	-28	-26	0

d	A,C	B	D
A,C	0	6	4
B	6	0	8
D	4	8	0

u	
A,C	10
B	14
D	12

Q	A,C	B	D
A,C	0	-18	-18
B	-18	0	-18
D	-18	-18	0

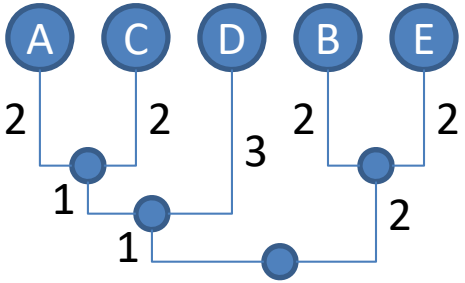
d	A,B,C	D
A,B,C	0	3
D	3	0

u	
A,B,C	3
B	3

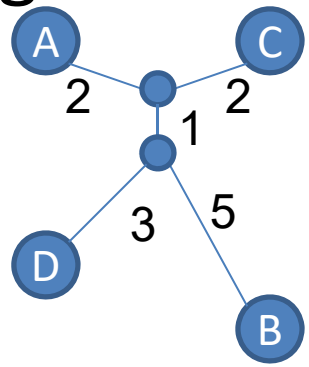
En el último paso, quitamos el centro y escribimos la distancia

Comparando los resultados

- UPGMA: (con un nodo más, E)



- Neighbor Joining:



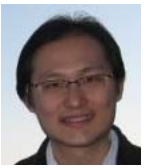
¿Qué método utilizar?

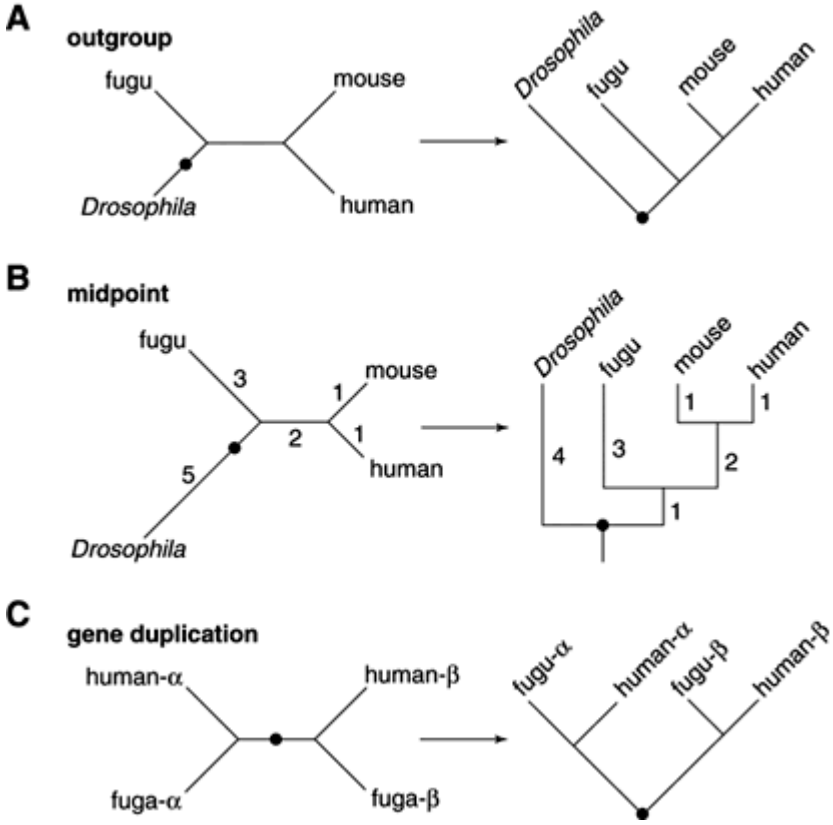
- No hay una respuesta definitiva
  - Hay diferentes campos
- En general, es bueno utilizar métodos que
  - No requieran hipótesis fuertes
  - Sean robustos (no producen resultados drásticamente diferentes cuando las entradas se cambian sólo un poco)
    - Construir múltiples árboles utilizando diferentes parámetros, a continuación, combinar
    - Construir árboles con diferentes subconjuntos de secuencias, a continuación, combinar
    - Utilizar métodos probabilísticos
  - computacionalmente eficientes
- Hay muchos otros algoritmos que no cubrimos, incluidos aquellos que consideran modelos de mutación.

Enraizamiento un árbol sin raíces

- ¿Cómo encontrar la raíz de un árbol sin raíces?
  - Por lo general, mediante el uso de un valor externo, algo que debe ser separado primero
  - Hay algunos otros métodos





Crédito de la imagen: Wikipedia, http://blog.ohinternet.com/wp-content/uploads/2011/03/fugu.jpg, http://www.currentprotocols.com/protocol/bi0601

observaciones

- Los diferentes tipos de DNA tienen diferentes tasas de mutación
  - árbol de genes frente a árbol de especies
- Algunos DNAs no se heredan según la ruta habitual
  - Por ejemplo, las bacterias pueden adquirir nuevo DNA tomado de plásmido (transferencia horizontal de genes)
  - Necesitamos un grafo filogenético general que permita múltiples padres y bucles
- La reconstrucción de árboles filogenéticos se beneficiaría de tener una alineación de secuencias múltiples precisa, y viceversa
  - Algunos métodos realizan los dos iterativamente

Resumen

- Los modelos de mutación nos permiten estimar formalmente el número de mutaciones ocurridas basado en los datos observados
  - modelo Jukes-Cantor de un parámetro
- Los arboles filogenéticos capturan eventos de separación y cuándo sucedieron
- formatos de archivo comunes para los árboles

Resumen

- Existen dos tipos principales de métodos de reconstrucción del árbol:
  - Basados en secuencia
    - máxima parsimonia
    - Máxima verosimilitud
  - basados en distancias
    - UPGMA
    - Unión de vecinos

Epílogo

CASO DE ESTUDIO

Caso de estudio: las clasificaciones inesperadas

- En los viejos tiempos, los biólogos clasifican las especies en función de sus características de alto nivel
  - Si una especie posee características que lo hacen los organismos similares a varios otros tipos de especies, puede ser difícil clasificar
  - Cuando se disponga de características moleculares (por ejemplo, secuencias de DNA), se pueden utilizar para clasificar las especies de una manera sistemática
    - Se descubrió que algunas clasificaciones anteriores son incompatibles con la evidencia molecular

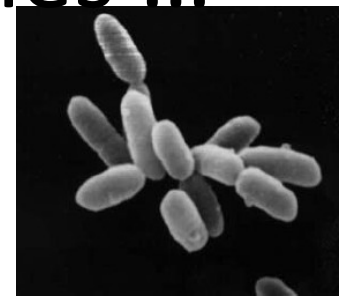
Caso de estudio: clasificaciones …

- Ejemplo 1: Mamíferos
  - Los murciélagos parecen pájaros, los delfines se ven como los peces, pero ambos son en realidad mamíferos
- Reino: Animalia (animales)
  Superphylum: Deuterostomados
  Filo: Chordata
  Subphylum: Vertebrata (animales con columna vertebral)
  Infraphylum: Gnathostomata (vertebrados con mandíbulas)
  Clase: Chondrichthyes (peces cartilaginosos)
  Superclase: Osteichthyes (peces óseos)
  Superclase: Tetrapoda (cuatro ramificac vertebrados)
  Clase: Aves (aves)
  Clase Mammalia (mamíferos)

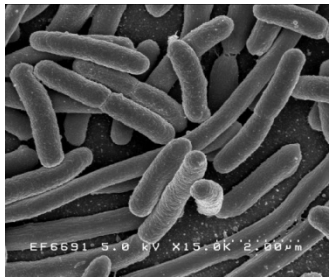
Fuente de la imagen: Wikipedia

Caso de estudio: clasificaciones …

- Ejemplo 2: Los tres dominios
- Todas las especies de la tierra pertenecen a uno de los tres dominios
  - arqueas
    - Unicelulares, sin núcleo
    - Por lo general, viven en lugares con condiciones extremas (por ejemplo, alta temperatura o la salinidad - "extremófilos")
  - Las bacterias
    - Unicelulares, sin núcleo
  - eucariotas
    - Muchos son multicelular, con núcleo



Halobacterias sp. cepa NRC-1, un archaaeon



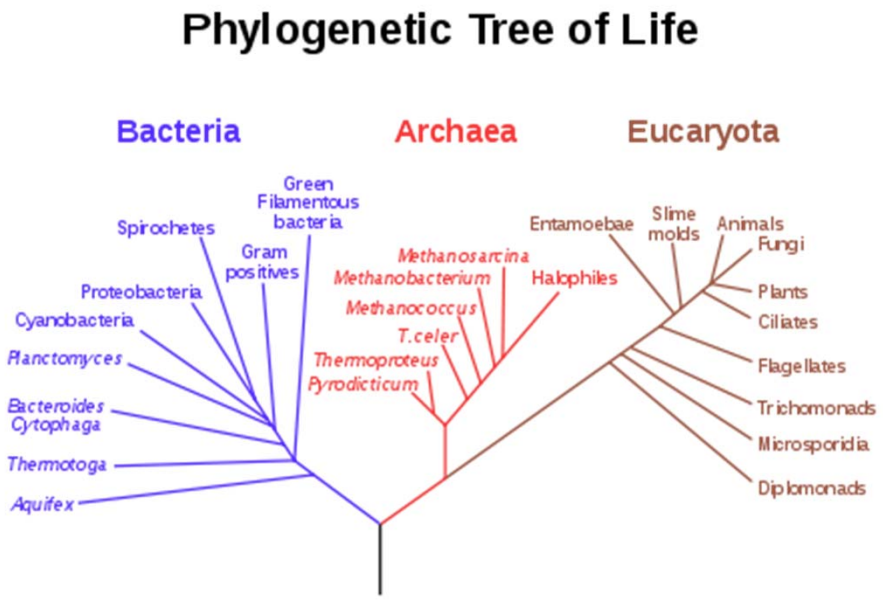
Escherichia coli, una beacterium

Varias especies eucariotas

Fuente de la imagen: Wikipedia

Caso de estudio: clasificaciones …

- Parece razonable suponer que los eucariotas se separaron primero de los otros dos
- Sin embargo, basándose en la secuencia de RNA ribosomal, algo tan importante que evoluciona lentamente, las arqueas están más cerca de los eucariotas que las bacterias



Fuente de la imagen: Wikipedia