RAxML

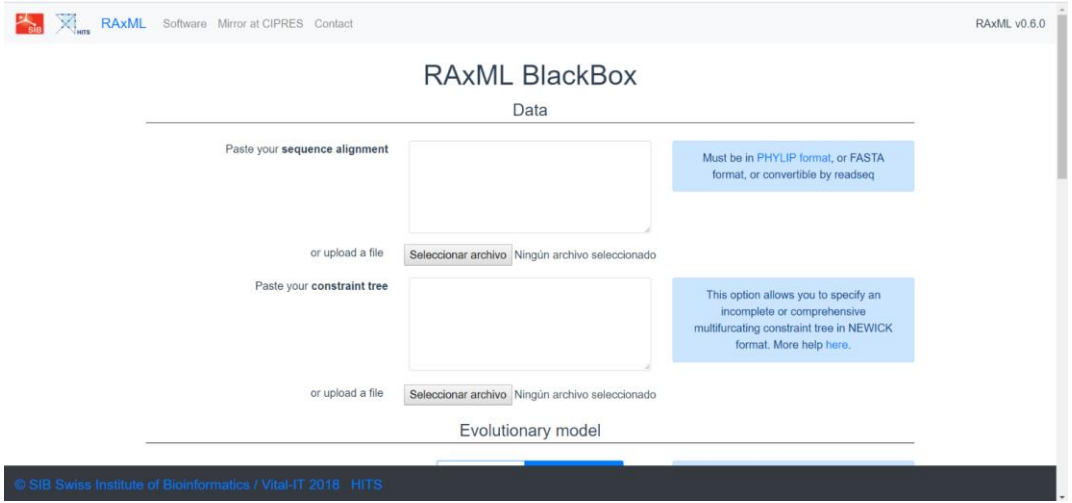
Jorge Yagüe - 698976
Eduardo Alonso - 728502

Instalación

- Manual y documentación: https://embnet.vital-it.ch/raxml-bb/
- Código fuente en GitHub: https://github.com/stamatak/standard-RAxML
- Para Windows:
  - WindowsExecutables_v8.2.10
  - WindowsExecutables_v8.2.4
- Para Linux:
  - Compilar la versión requerida siguiendo las instrucciones del fichero README
- Versiones distintas:
  - Estándar: raxmlHPC
  - Varios hilos de ejecución: raxmlHPC-PTHREADS
  - Distintos procesadores: SSE3 (7-8 años) y AVX (5-6 años)

Instalación

- Versión online:
  - https://raxml-ng.vital-it.ch/
  - Introducir alineamiento en formato FASTA y seleccionar opciones



Uso

- Mediante línea de comando
- Algunas opciones:
  - -s fichero con las secuencias alineadas
  - -m modelo evolutivo a utilizar
  - -n nombre del fichero de salida
  - -T número de hilos a utilizar (solo versión PTHREADS)
  - -p semilla para la aleatoriedad
  - -f algoritmo a ejecutar (Hill-Climbing por defecto)
  - -h parámetros disponibles

Modelos evolutivos

- Se van a utilizar solo modelos para nucleótidos
- CAT: recomendable solo para más de 50 secuencias
  - 4 veces más rápido que GAMMA
  - No es útil para alineaciones de pocas secuencias ya que no puede estimar bien
  - GTRCAT: el árbol final es evaluado con GTRGAMMA
  - GTRCATI: el árbol final es evaluado con GTRGAMMAI
- GAMMA: válido para pocas secuencias
  - Mayor coste en tiempo y memoria
  - GTRGAMMA
  - GTRGAMMAI: igual que GTRGAMMA, pero estima la proporción de partes invariables

Modelos evolutivos

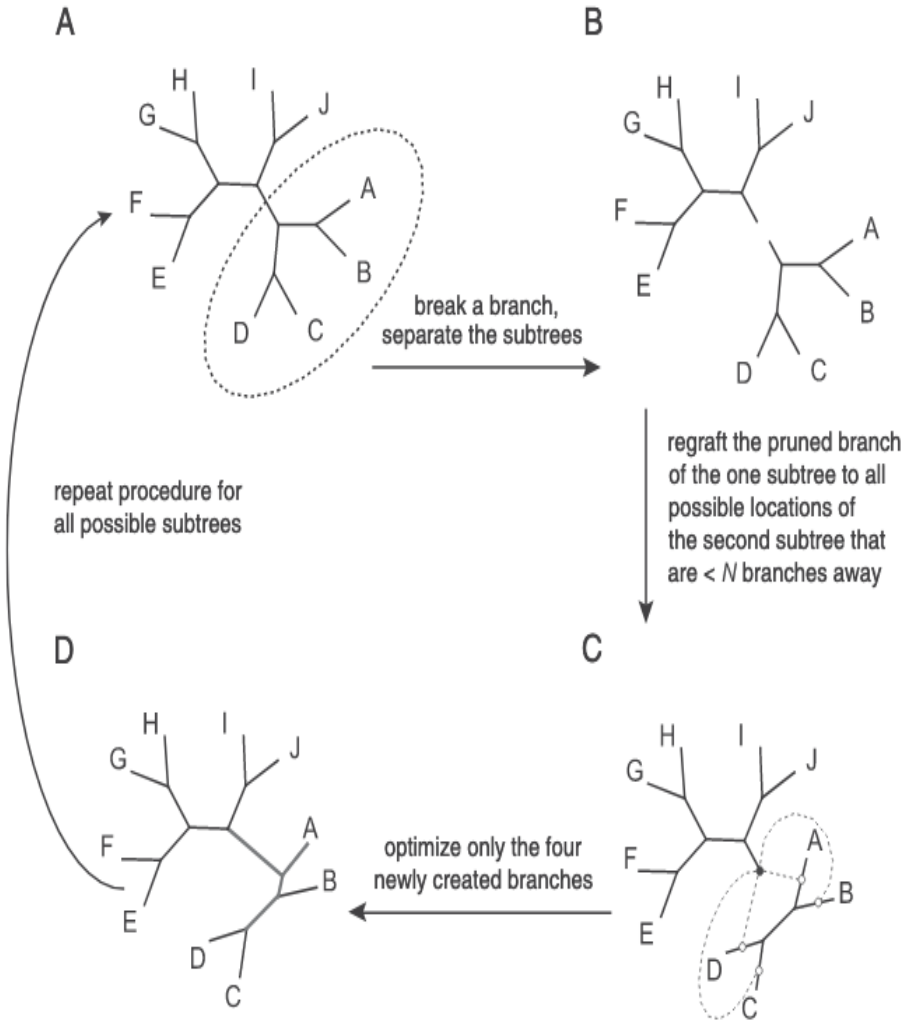
```
NUCLEOTIDES:

  "-m GTRCAT[X]"       : GTR + Optimization of substitution rates + Optimization of site-specific
                         evolutionary rates which are categorized into numberOfCategories distinct
                         rate categories for greater computational efficiency.  Final tree might be evaluated
                         under GTRGAMMA, depending on the tree search option.
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
  "-m GTRCATI[X]"      : GTR + Optimization of substitution rates + Optimization of site-specific
                         evolutionary rates which are categorized into numberOfCategories distinct
                         rate categories for greater computational efficiency.  Final tree might be evaluated
                         under GTRGAMMAI, depending on the tree search option.
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
  "-m ASC_GTRCAT[X]"   : GTR + Optimization of substitution rates + Optimization of site-specific
                         evolutionary rates which are categorized into numberOfCategories distinct
                         rate categories for greater computational efficiency.  Final tree might be evaluated
                         under GTRGAMMA, depending on the tree search option.
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
                         The ASC prefix willl correct the likelihood for ascertainment bias.
  "-m GTRGAMMA[X]"     : GTR + Optimization of substitution rates + GAMMA model of rate
                         heterogeneity (alpha parameter will be estimated).
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
  "-m ASC_GTRGAMMA[X]" : GTR + Optimization of substitution rates + GAMMA model of rate
                         heterogeneity (alpha parameter will be estimated).
                         The ASC prefix willl correct the likelihood for ascertainment bias.
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
  "-m GTRGAMMAI[X]"    : Same as GTRGAMMA, but with estimate of proportion of invariable sites.
                         With the optional "X" appendix you can specify a ML estimate of base frequencies.
```

Estrategia de búsqueda

El primer paso de la estrategia de búsqueda empleada por RAXML es la generación de un árbol inicial. Este árbol se construye agregando las secuencias una por una en orden aleatorio, e identificando su ubicación óptima en el árbol bajo el criterio de optimización de parsimonia.

El segundo paso de la estrategia de búsqueda implica un método conocido como ‘lazy subtree rearrangement’ (LSR). Todos los subárboles posibles de un árbol se recortan y se vuelven a insertar en todas las ubicaciones posibles, siempre que el número de ramas que separa los puntos recortados y de inserción sea menor que N ramas



Construcción árbol máxima verosimilitud

Podemos hacer un análisis de máxima verosimilitud en RAXML escribiendo:

```
RAXMLHPC -s protein.phy -n A1 -m PROTGAMMAWAG
```

La opción `-s protein.phy` especifica el archivo de datos de secuencia a analizar

La opción `-n A1` especifica el apéndice del nombre del archivo que se agregará a todos los archivos de salida producidos en esta ejecución, que se incluirá en el formato RAXML_filename.A1

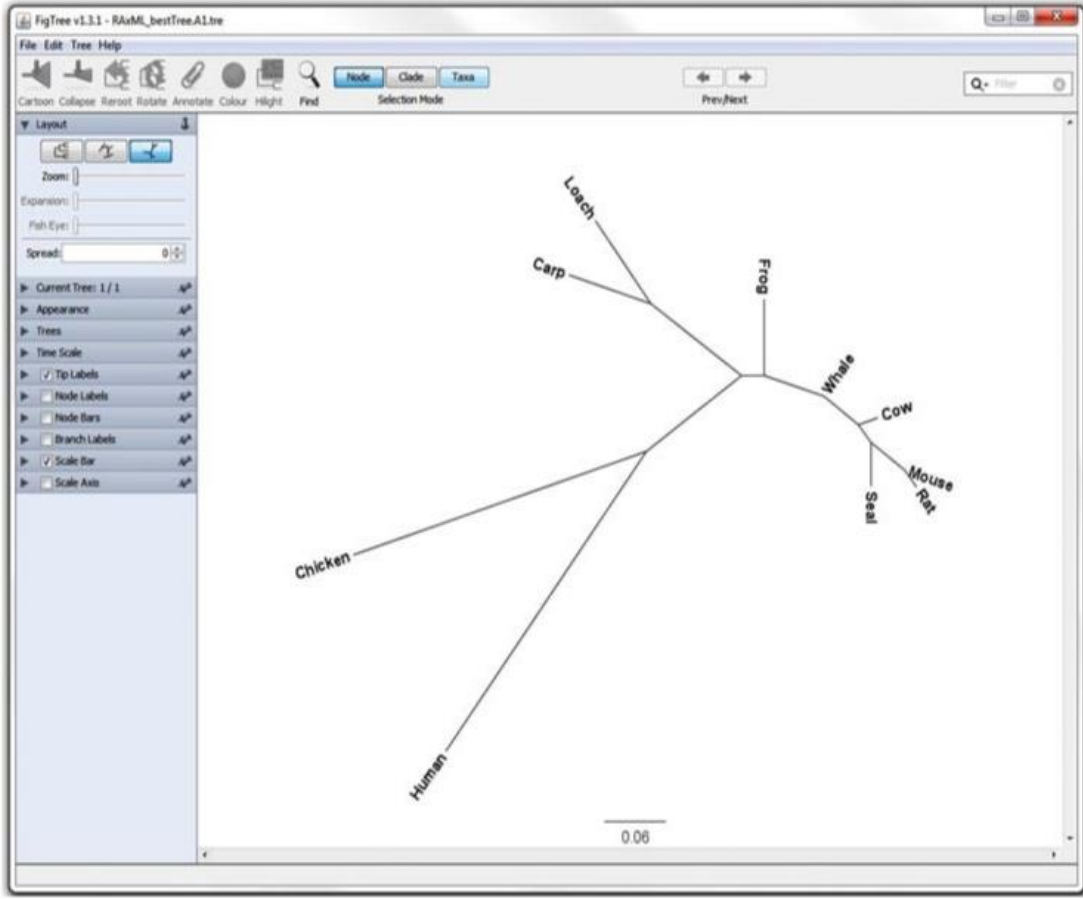
La opción `-m PROTGAMMAWAG` especifica a tres parámetros asociados con el modelo de evolución de secuencia empleado: primero, que estamos utilizando datos de proteínas (PROT); segundo, que estamos teniendo en cuenta la heterogeneidad de la velocidad entre los sitios en nuestra alineación mediante el uso de la distribución gamma (GAMMA); y tercero, que estamos empleando la matriz de sustitución de aminoácidos de Whelan y Goldman (WAG).

Si quisiéramos elegir una matriz de sustitución de aminoácidos diferente (por ejemplo, RTREV), sería necesario simplemente reemplazar la parte WAG de la opción -m con RTREV. Para usar las frecuencias básicas empíricas extraídas de la alineación, todo lo que se necesita es agregar la letra F a la opción -m

Visualización del árbol de máxima verosimilitud

El examen del contenido del directorio donde se ejecutaron los diferentes análisis muestra que RAXML genera varios archivos diferentes de varios análisis diferentes. Estos archivos brindan información detallada y resultados sobre el análisis (RAXML_info.A1), el árbol de máxima probabilidad (RAXML_bestTree.A1), el árbol de parsimonia inicial (RAXML_parsimonyTree.A1), etc.

Normalmente, los archivos de salida más útiles son los archivos de árbol, que están escritos en el formato NEWICK.



Raíces de árboles filogenéticos

Los árboles filogenéticos pueden tener o no raíz. Los árboles filogenéticos con raíz tienen dirección, ya que todos los linajes representados en el árbol se originan del mismo ancestro común. En contraste, los árboles sin raíces, no nos informan sobre los dirección de la evolución.

Si queremos usar la secuencia de Carp como el grupo externo, el comando RAXML debería ser:

RAXMLHPC -s protein.phy -n A5 -m PROTGAMMAWAGF -o Carpa

El grupo externo también puede consistir en más de una secuencia. Por ejemplo, utilizar las dos secuencias de peces (Carp y Loach) como grupo externo, en cuyo caso el comando RAXML debería ser:

RAXMLHPC -s protein.phy -n A6-m PROTGAMMAWAGF -o Carp, Loach

Tener en cuenta que si las secuencias especificadas como grupo externo no forman un grupo monofilético, no podrá ubicarlas todas como grupo externo. En este caso, imprimirá una advertencia en el archivo de información

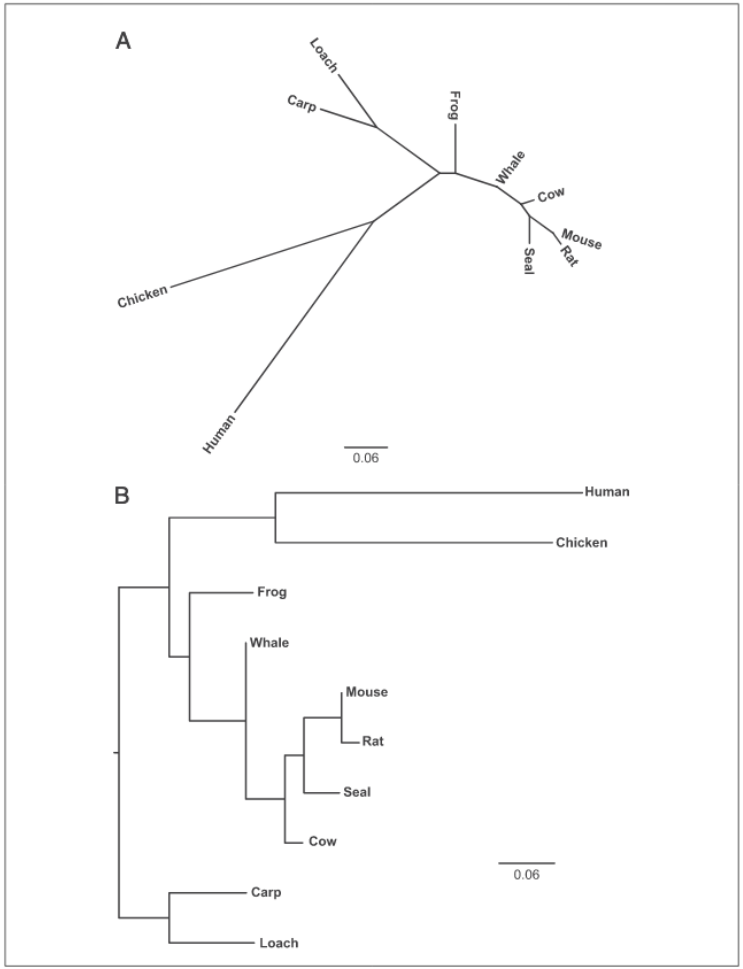


Figure 19.11.4 Examples of an unrooted (**A**) and a rooted (**B**) phylogenetic tree for the example data set. The unrooted tree on the top panel, if rooted on the branch leading to the `Carp` and `Loach` sequences, corresponds to the rooted tree on the bottom. Note that in the unrooted tree neighboring sequences are not necessarily closely related.

Boostrap

Para realizar un análisis de bootstrap en RAXML, es necesario especificar dos opciones adicionales.

RAXMLHPC -s protein.phy -n A7 -m PROTGAMMAWAGF --b 0123 -# 100

La primera es la opción -b, donde n puede ser cualquier entero positivo, y que especifica la semilla de números aleatorios requerida para el análisis bootstrap. El uso del mismo número de inicialización aleatorio en diferentes ejecuciones del mismo conjunto de datos dará lugar a la generación de conjuntos de datos duplicados idénticos de arranque, por lo que puede ser conveniente seleccionar un nuevo número de inicialización aleatorio cada vez que se ejecute un análisis.

La segunda opción es la opción -# n, donde n puede ser cualquier entero positivo, y que especifica el número de réplicas de bootstrap a realizar.

El número típico de réplicas de bootstrap realizadas varía mucho entre los estudios, y puede variar de cien a miles de réplicas. RAXML permite al usuario estimar automáticamente cuándo se ha realizado un número apropiado de réplicas mediante el uso de varios criterios de detención diferentes. La lógica que subyace a todos estos criterios es la misma; después de cada 50 réplicas de arranque, el programa realiza 100 divisiones aleatorias del conjunto de réplicas de arranque en dos mitades y calcula las estadísticas, que varían según el criterio implementado. Para ello se añade autoFC a la opción -#

```
RAXMLHPC -s protein.phy -n A8-m PROTGAMMAWAGF --b 0123 -#autoFC
```

Bootstrap

Podemos indicar a RAXML que dibuje valores bootstrap en el árbol de máxima verosimilitud usando el siguiente comando:

RAXMLHPC -n A9 -m PROTGAMMAWAGF -f b -t bestTree.A4 -z bootstrap.A8

La opción -f b especifica el análisis que se realizará (dibuja los valores de bootstrap en un árbol determinado)

La opción -t bestTree.A4 especifica el árbol en el que queremos que se muestren los valores,

La opción -z bootstrap.A8 Especifica el archivo que contiene los árboles generados mediante bootstrapping.

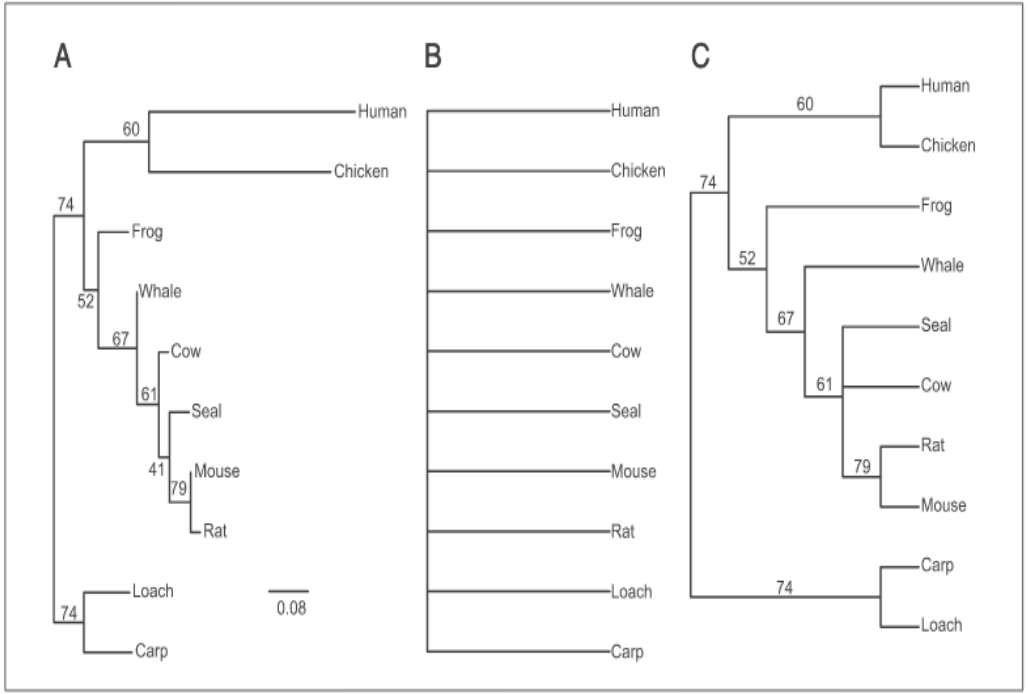


Figure 19.11.5 Different ways of visualizing bootstrap support values on phylogenetic trees. (**A**) Bootstrap support values depicted on the maximum likelihood tree. (**B**) The strict consensus tree, which is completely unresolved because none of the groupings was present in all bootstrap replicate trees. (**C**) Bootstrap support values depicted on the majority rule tree.

Bootstrap: Árboles de consenso

We can also use the Bootstrap replicates to build consensus trees, RAxML supports strict, majority rule, and extended majority rule consenus trees:

- strict consensus: **raxmlHPC -m GTRCAT -J STRICT -z RAxML_bootstrap.T14 -n T16**
- majority rule: **raxmlHPC -m GTRCAT -J MR -z RAxML_bootstrap.T14 -n T17**
- extended majority rule: **raxmlHPC -m GTRCAT -J MRE -z RAxML_bootstrap.T14 -n T18**

Bootstrap acelerado

Para facilitar un análisis más rápido, el programa RAXML también contiene un algoritmo de arranque rápido que es al menos un orden de magnitud más rápido que el estándar, mientras que es igualmente preciso (Stamatakis et al., 2008). Para ejecutar un análisis utilizando este algoritmo, el usuario simplemente necesita cambiar la opción -b a la opción -x, donde puede ser cualquier entero positivo. Después de reemplazar la opción -b n con la opción -x n, el comando RAXML debería verse así:

RAXMLHPC -s protein.phy -n A12-m PROTGAMMAWAGF --x 0123 - # 100or,

Si se debe implementar el criterio de detención basado en la frecuencia, debería tener el siguiente aspecto:

RAXMLHPC -s protein.phy -n A13-m PROTGAMMAWAGF --x 0123 - # autoFC

Comparar árboles filogenéticos

En este caso, podemos usar la prueba SH para evaluar si el árbol de máxima verosimilitud es significativamente mejor que el árbol de vertebrados tradicional usando el siguiente comando RAXML:

```
RAXMLHPC -f h -s protein.phy-n A15 -m PROTGAMMAWAGF -t bestTree.A4 -z verte-brate.tree
```

En este comando, la opción ‘-f h’ especifica que queremos realizar una prueba SH.

El árbol de máxima verosimilitud se especifica mediante la opción -t bestTree.A4.

El árbol de vertebrados se especifica mediante la opción -z verte-brate.treeoption. Este último archivo se puede crear escribiendo la filogenia de vertebrados para las secuencias utilizadas en el formato NEWICK

Análisis particionados

Una tarea común es realizar análisis particionados. Necesitamos pasar la información sobre particiones a RAxML a través de un archivo de texto simple que se pasa a través del parámetro -q. Para una partición simple de nuestro conjunto de datos de ADN podemos escribir:

raxmlHPC -m GTRGAMMA -p 12345 -q simpleDNApartition.txt -s dna.phy -n T21

El archivo simpleDNApartition divide la alineación en dos regiones de la siguiente manera:

DNA, p1 = 1-30

DNA, p2 = 31-60

Si queremos hacer una partición más elaborada mediante la posición del primer, segundo y tercer codón, podemos ejecutar:

raxmlHPC -m GTRGAMMA -p 12345 -q dna12_3.partition.txt -s dna.phy -n T23

El archivo de partición ahora se ve así:

DNA, p1 = 1-60 \ 3,2-60 \ 3

DNA, p2 = 3-60 \ 3

Fichero de salida

El análisis produce un solo archivo de salida (RAXML_info.A15) que informa los resultados de la prueba SH en sus últimas líneas:

`Model optimization, best Tree:-411.163389` informa la puntuación de probabilidad del mejor árbol

`Tree: 0 Likelihood:-423.777863` informa la puntuación de probabilidad del árbol de vertebrados.

`D(LH):-12.614474 SD: 6.714754` informa la diferencia D (LH) en las puntuaciones de probabilidad entre los dos árboles y su desviación estándar SD.

`Significantly Worse: No (5%),No (2%), No (1%)` Finalmente, la última línea informa si esta diferencia de probabilidad entre el mejor árbol y el árbol de vertebrados es significativa en el nivel del 5%, 2% y 1%.

Dado que el resultado informado es No para los tres niveles de importancia, se puede concluir que el árbol filogenético de máxima probabilidad estimado a partir de este conjunto de datos no difiere significativamente de la filogenia de vertebrados estándar.