Tutoral de uso de phyml3 para selección de modelos y búsquedas intensivas de árboles

Pablo Vinuesa, CCG-UNAM, Cuernavaca, México

2019-07-28

Contents

Tutoral de uso de phyml3

Estas prácticas fueron diseñadas para los Talleres Internacionales de Bionformática, TIB2019-T3. Este tutorial forma parte del repositorio GitHub - TIB-filoinfo

Objetivos

En esta práctica aprenderemos a usar phyml para seleccionar modelos de sustitución, paramétricos (DNA) y empíricos (prot), usando LRTs y análisis de AICs con la ayuda scripts sencillos de R para despliegue gráfico de resultados y análisis numérico de valores de lnL para los modelos en competición.

Además aprenderemos a hacer búsquedas intensivas bajor el mejor modelo seleccionado, usando múltiples árboles de semilla, así como el despliegue y edición de los mejores árboles encontrados con FigTree.

NOTAS DE INSTALACION DE PHYML DESDE EL REPOSITORIO GitHub

1. Descarga la ultima version estable desde el repositorio GitHub de phyml wget -c https://github.com/stephaneguindon/phyml/archive/v3.3.20190321.tar.gz
2. desempaqueta y ve al direcotorio tar -xvzf v3.3.20190321.tar.gz cd phyml-v3.3.20190321
3. Para compilarlo, ejecuta:

./autogen.sh

./configure

make

4. El binario aparecerá en el subdirecorio src cd src

./phyml –help

5. Sugiero copiar phyml a un directorio en el $PATH sudo cp phyml /usr/local/bin

USO DE PHYML y seleccion de modelos de sustitucion nucleotidica

Nota: Phyml requiere alineamientos múltiples de DNA o proteína en formato phylip.

Opciones mas usadas de PhyML v3

```
# para ver la ayuda: phyml --help

# 1) Un ejemplo de corrida del set de primates bajo el modelo definido en PAUP como: rclass=(a b c c b
# -i input_file
# -d nt|aa
# -b -4 usa SH-like branch support values (-b 100 corre 100 repl de bootstrap)
# -m  PARA NT: (modelo de sustitucion HKY85 (default) | JC69 | K80 | F81 | F84 | TN93 | GTR o definido
# -m PARA AA LG (default) WAG | JTT | MtREV | Dayhoff | DCMut | RtREV | CpREV | VT | Blosum62 | MtMam |
# -f frec. de nt [e empirica| m estimada por ML]
# -t ts/tv
# -v prop inv [0,1] o estimarla con e
# -c no. clases para discretizar la funcion gamma
# -a e (estima gamma) -a digito (fija su valor)
# -s (define la busqueda o search) NNI SPR BEST (mejor de ambas)
# --rand_start (inicia busquedas de arboles de adic. sec. aleatorizada)
# --n_rand_starts no. (no. de arboles semilla)
```

Ejemplos de invocación de PhyML v3 para alineamientos de secuencias de DNA y proteína

```
# A: Ejemplo de corrida para NT; recureda que los modificatodres de tasa indicados con -m se aplican en
phyml -i primates.phy -m HKY85 -c 4 -a e -o tlr -s BEST --rand_start --n_rand_starts 5

# B: Ejemplo de corrida para AA
phyml -i 10_GDP_eucar_prokar_clu0_prof2prof.faa  -d aa -m WAG -c 4 -a e -v e -s BEST
```

SELECCION DE MODELOS USANDO PHYML

I. LRTs para selección de modelos de nucleótidos

Notas: para acelerar el procedimiento de seleccion de modelos, podemos usar un arbol aproximado, tipo NJ, calculado rapidamente bajo el modelo mas sencillo, sin optimizar parametros. Esta filogenia la usaremos para para los pasos siguientes, evitando tener que volver a estimar topologia y longitudes de rama, dejando que phyml optimice solo los valores de los parametros del modelo bajo evaluacion:

```
# visualizemos el alineamiento para confirmar que es tal y que está en formato phylip (requerido por ph
head primates.phy && tail primates.phy

phyml -i primates.phy -m 000000 -o n

# renombremos el archivo
mv primates.phy_phyml_tree.txt primates_JC.ph

# podemos visualizar el arbol con:
figtree primates_JC.ph &
```

```
### >>>>> ahora veamos la seleccion de modelos (JC vs K2P, JC vs F81; JC vs HKY; HKY vs HKY+G, y HKY+G v
# codificacion de tasas de sustitucion relativas:
#  012345 (ac ag at cg ct gt)
#  A  == G    Ti (purinas)
#  ||  X ||   Tv
#  C  == T    Ti (pirimidinas)
# Comencemos evaluacion de modelos; anota el valor de -lnL para cada caso

# 1. Modelo JC
phyml -i primates.phy -d nt -m JC69 -c 1 -b 0 -u primates_JC.ph -o lr

# Anotemos el valor de Log-likelihood para este modelo
cat *stats.txt

# o usando grep:
grep Log-likelihood *stats.txt
grep Log-likelihood *stats.txt > JC.fit

# 2. Modelo K80; veamos lo que tarda al usar el "user-tree" vs. tener que estimarlo
phyml -i primates.phy -d nt -m K80 -c 1 -u primates_JC.ph -o r

grep Log- *stats.txt > K80.fit

# 3. Modelo F81
#  phyml -i primates.phy -d nt -m F81 -c 1
phyml -i primates.phy -d nt -m 111111 -f e -c 1 -u primates_JC.ph -o r

grep Log- *stats.txt > F81.fit

# 4. Modelo HKY85
#  phyml -i primates.phy -d nt -m HKY85 -c 1
phyml -i primates.phy -d nt -m 121121 -f e -c 1 -u primates_JC.ph -o r

grep Log- *stats.txt > HKY85.fit

# 5. Modelo HKY85+G
#  phyml -i primates.phy -d nt -m HKY85 -c 4 -a e
phyml -i primates.phy -d nt -m 121121 -f e -c 4 -a e -u primates_JC.ph -o r

grep Log- *stats.txt > HKY85G.fit

# 6. Modelo HKY85+G+I
#  phyml -i primates.phy -d nt -m HKY85 -c 4 -a e -v e
phyml -i primates.phy -d nt -m 121121 -f e -c 4 -a e -v e -u primates_JC.ph -o r

grep Log- *stats.txt > HKY85GI.fit
```

Análisis gráfico y numérico de resultdos usando herramientas del shell y R

```
# veamos el listado de archivos *.fit, en formato de una linea
ls -1 *fit
ls -1 *fit > fit.order
cat fit.order
```

```
# peguemos todos los archivos *.fit, uno detras del otro, en el orden que nos muestra ls -1 *.fit
cat *fit > model_fit.list
paste fit.order model_fit.list
paste fit.order model_fit.list | awk '{print $1,$4}'
paste fit.order model_fit.list | awk 'BEGIN{OFS=","}{print $1, $4}' | sed 's/\.fit//' > model_fits.csv
paste fit.order model_fit.list | awk 'BEGIN{OFS="\t"}{print $1, $4}' | sed 's/\.fit//' > model_fits.tsv

# limpiemos un poco el directorio de trabajo
rm *list *fit *order

# 6. Evaluacion del merito relativo de los modelos HKY85+G vs HKY85+G+I
#    mediante la prueba de cocientes de verosimilitud (LRT), en R
R

# 6.1 leamos los datos en una estructura tipo dataframe
system("ls")
dfr <- read.csv(file="model_fits.csv", header = FALSE)

# ver la estructura del data frame
str(dfr)

# imprimelo a pantalla
dfr

# aniadimos un header (nombramos las columnas con las variables)
names(dfr) <- c("model", "lnL")
dfr

# 6.2 veamos un plot sencillo y rapdio; que puedes concluir?
plot(dfr$model, dfr$lnL, xlab="substitution models", ylab="ln-L values", main="comparing model fits")


# 6.3 Entre los dos modelos que parecen empatados, veamos si existen o no diferencias significativas,
#     corriendo una prueba estadistica formal: dado que los modelos HKY+G u HKY+G+I estan anidados,
#     podemos correr un simple LRT (prueba de razones de verosimilitudes): LRT = 2(lnL1 - lnL 0)

dfr

# LRT = 2(   lnL1    -    lnL0  )
xsq <- 2*( -5778.111 - -5781.239) # 2 veces la diferencia de lnL entre el modelo nulo - alternativo
xsq # [1] 6.256


pchisq(xsq,1)      # 0.9876227;  este es el valor critico de la X2 para el estadistico LRT con 1 grado
1-(pchisq(xsq,1))  # 0.01237734 esta es la p asociada a dicho valor crítico, SIGNIFICATIVA con alpha =

q()         # para salir de R
            # decimos que NO quieren salvar sesion


# Finalmente, estimemos la filogenia de máxima verosimilitud bajo el modelo seleccionado:
# propongo que seamos conservadores y nos quedemos con HKY+G
# Nótese que usamos sólo la parameterización del modelo (-m HKY85 -c 4 -a e)
```

```
# dejando que phyml ahora optimice todos los parámetros, incluyendo ahora
# topología, longitudes de ramas y tasas -o tlr

phyml -i primates.phy -d nt -m HKY85 -c 4 -a e -o tlr

# veamos las estimas de los parámetros
cat primates.phy_phyml_stats.txt

# despliega el árbol
figtree primates.phy_phyml_tree.txt &
```

II. AICs para selección modelos de aminoácidos

```
# Evaluemos un subconjunto de los modelos empiricos (matrices de sustitución) implementados en phyml pa
# - Amino-acid based models : LG (default) | WAG | JTT | MtREV | Dayhoff | DCMut | RtREV | CpREV | VT |
#                       Blosum62 | MtMam | MtArt | HIVw |  HIVb | custom

# primero seleccionemos la mejor matriz de base y además ajustando con ajuste empírico de frecuencias y
#   una distribución gamma para modelar heterogeneidad de tasas entre sitios
# NOTA: copia todo el bloque, hasta pasado el done, y pégalo en tu consola

# primero estimemos un árbol rápido
phyml -i primates_21_AA.phy -d aa -m BLOSUM62 -c 1

# y renombrémoslo, para su uso como usertree
mv primates_21_AA.phy_phyml_tree.txt primates_Blosum62.ph

# Corramos un bucle for para hacer ésto más eficiente
for mat in BLOSUM62 DAYHOFF JTT LG MTMAM WAG; do
     echo " >>> running: phyml -i primates_21_AA.phy -d aa -m $mat -u primates_Blosum62.ph -c 1 -v 0 -o
     phyml -i primates_21_AA.phy -d aa -m $mat -u primates_Blosum62.ph -c 1 -o r &> /dev/null
     grep Log- *21_AA*stats.txt > ${mat}.fit

     echo " >>> running: phyml -i primates_21_AA.phy -d aa -m $mat -c 4 -a e -u primates_Blosum62.ph -o
     phyml -i primates_21_AA.phy -d aa -m ${mat} -c 4 -a e -u primates_Blosum62.ph -o r &> /dev/null
     grep Log- *21_AA*stats.txt > ${mat}+G.fit

     echo " >>> running: phyml -i primates_21_AA.phy -d aa -m $mat -f e -c 1 -u primates_Blosum62.ph -o
     phyml -i primates_21_AA.phy -d aa -m $mat -f e -c 1 -u primates_Blosum62.ph -o r &> /dev/null
     grep Log- *21_AA*stats.txt > ${mat}+f.fit

     echo " >>> running: phyml -i primates_21_AA.phy -d aa -m $mat -u primates_Blosum62.ph -f e -a e -o
     phyml -i primates_21_AA.phy -d aa -m $mat -u primates_Blosum62.ph -f e -a e -c 4 -o r &> /dev/null
     grep Log- *21_AA*stats.txt > ${mat}+f+G.fit
done
```

Análisis gráfico y numérico de resultados usando herramientes del filtrado y R

```
# veamos el listado de archivos *.fit, en formato de una linea
ls -1 *fit
ls -1 *fit > fit.order
cat fit.order

# peguemos todos los archivos *.fit, uno detras del otro, en el orden que nos muestra ls -1 *.fit
```

```
cat *fit > model_fit.list
paste fit.order model_fit.list
paste fit.order model_fit.list | awk '{print $1,$4}'
paste fit.order model_fit.list | awk 'BEGIN{OFS=","}{print $1, $4}' | sed 's/\.fit//' > model_fits.csv
paste fit.order model_fit.list | awk 'BEGIN{OFS="\t"}{print $1, $4}' | sed 's/\.fit//' > model_fits.tsv

# limpiemos un poco el directorio de trabajo
rm *list *fit *order

cat model_fits.tsv | sort -nrk2

# 6. Evaluacion del merito relativo de los modelos HKY85+G vs HKY85+G+I
#    mediante la prueba de cocientes de verosimilitud (LRT), en R
R

# 6.1 leamos los datos en una estructura tipo dataframe
system("ls")
dfr <- read.csv(file="model_fits.csv", header = FALSE)

# ver la estructura del data frame
str(dfr)

# imprimelo a pantalla
dfr

# aniadimos un header (nombramos las columnas con las variables)
names(dfr) <- c("model", "lnL")
head(dfr)

# veamos un plot sencillo y rapdio; que puedes concluir?
opar <- par()
plot(dfr$model, dfr$lnL, ylab="ln-L values", main="comparing model fits", las=2)
par(opar)

# mejor un dotchart()
dfrord <- dfr[order(dfr$lnL),]
dotchart(dfrord$lnL, labels = dfrord$model, main = "sorted model scores", xlab="lnL score")

# veamos los resultados ordenados por valor decreciente de lnL
dfr[order(dfr$lnL, dfr$model, decreasing = TRUE), ]
#          model       lnL
#10      JTT+f+G -4779.895
#12        JTT+G -4816.827
#9         JTT+f -4863.585
#22      WAG+f+G -4871.378
#18    MTMAM+f+G -4879.238
#11          JTT -4901.528
# ... truncado

# ¿Qué podemos concluir de esta salida?

# Computo de AICmin y deltaAICi
# recuerda: AICi = 2Ki - 2(lnLi); donde Ki = numero parametros libres del modelo i; lnLi = lnL del model
```

```
AIC_JTTfg <- 2*19 -2*(-4779.895)
AIC_JTTG  <- 2*1 -2*(-4816.827)
AIC_JTTf <- 2*19 -2*(-4863.585)

cat(AIC_JTTfg, "\n", AIC_JTTG, "\n", AIC_JTTf, "\n")

# compute and print the deltaAICs for the 2 top models
AICmin <- AIC_JTTfg
deltaJTTG <- AIC_JTTG - AICmin
deltaJTTf <- AIC_JTTf - AICmin

cat ("deltaAIC", "\t", "value", "\n", "deltaJTTG", "\t", deltaJTTG,  "\n", "deltaJTTf", "\t", deltaJTTf

# > 37.864

# >>> CONCLUSION: el análisis de AIC claramente muestra que por mucho el mejor modelo de los evaluados

# finalmente, hagamos una busqueda intensiva bajo este modelo, usando el algoritmo BEST the rearreglo d
phyml -i primates_21_AA.phy -d aa -m JTT -c 4 -a e -f e -s BEST --rand_start --n_rand_starts 10

# despliega y edita el árbol
figtree primates_21_AA.phy_phyml_tree.txt &
```