Pràctica amb ordinador 3. Filogènia mitocondrial

Principals Transicions Evolutives

2023-04-20

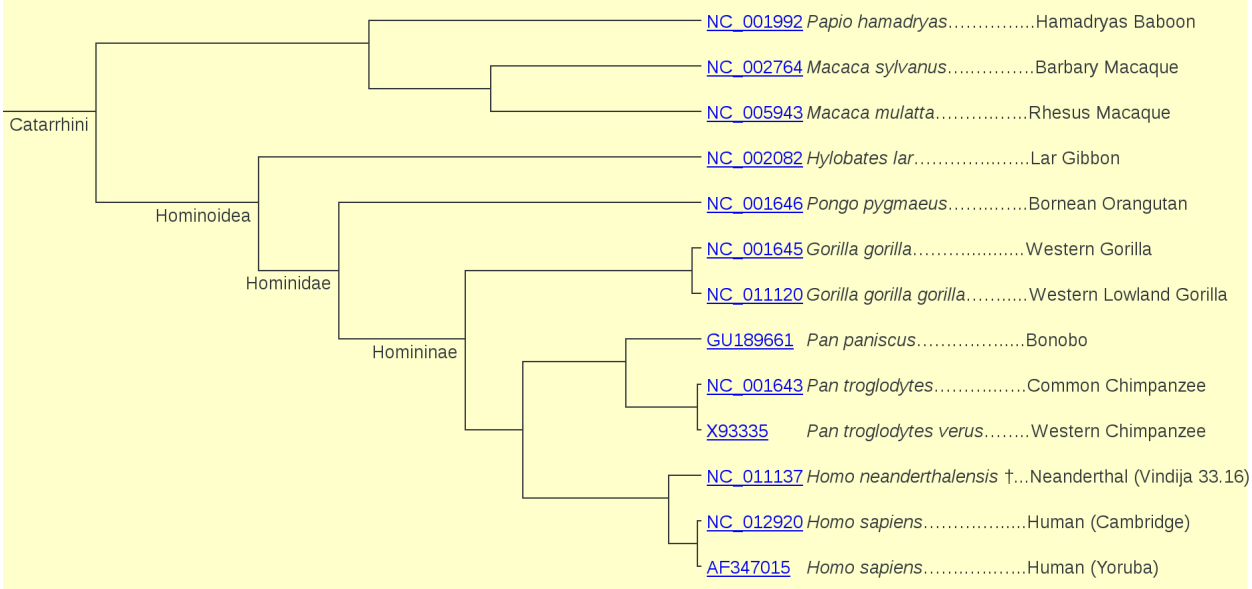
Preparació de l'ordinador

En aquesta pràctica utilitzarem els paquets de R `DECIPHER` (Wright 2016) i `phangorn` (Schliep 2011), que pots instal · lar amb els comandaments següents:

```
install.packages('phangorn')
if (!requireNamespace('BiocManager', quietly=TRUE))
    install.packages('BiocManager')
BiocManager::install('DECIPHER')
```

Objectiu

L'objectiu d'esta pràctica és reproduir la filogènia dels primats catarrins a partir de les seqüències mitocondrials completes de 13 individus de 9 espècies diferents, tal com apareix a la figura següent, procedent del web de PhyloTree: http://www.phylotree.org/resources/mtDNA_human_relatives.htm.



El procediment hauria d'incloure els passos següents:

1. Descarregar les seqüències nucleotídiques dels genomes mitocondrials.
2. Alinear-les amb el mètode implementat al paquet `DECIPHER`.
3. Obtenir la filogènia mitjançant algun mètode ràpid, com ara el *Neighbor-Joining*.

Descàrrega de les seqüències.

La taula 1 indica els números d'accés de les seqüències que necessitem i l'espècie a la qual pertany cada seqüència.

Taula 1: Números d'accés i espècies de les seqüències mitocondrials seleccionades.

Número d'accés	Espècie
Y18001	Papio hamadryas
AJ309865	Macaca sylvanus
AY612638	Macaca mulatta
X99256	Hylobates lar
D38115	Pongo pygmaeus
D38114	Gorilla gorilla
X93347	Gorilla gorilla gorilla
GU189661	Pan paniscus
D38113	Pan troglodytes
X93335	Pan troglodytes verus
AM948965	Homo sapiens neanderthalensis
J01415	Homo sapiens (Cambridge)
AF347015	Homo sapiens (Yoruba)

Pots descarregar les seqüències en format FASTA, una a una, seguint els enllaços de la Taula 1. Caldria guardar-les totes al mateix arxiu de text pla, respectant el format FASTA. Alternativament, podem utilitzar els comandaments següents per descarregar en la carpeta de treball un arxiu FASTA amb les 13 seqüències:

```
Numeros <- c('Y18001', 'AJ309865', 'AY612638', 'X99256', 'D38115',
             'D38114', 'X93347', 'GU189661', 'D38113', 'X93335',
             'AM948965', 'J01415', 'AF347015')
API <- 'https://www.ebi.ac.uk/ena/browser/api/fasta/'
URL <- paste0(API, paste(Numeros, collapse = ','))
download.file(URL, destfile = 'primats.fasta')
```

Executa el codi anterior i comprova que a la carpeta de treball se t'ha descarregat un arxiu anomenat `primats.fasta`. Podràs obrir-lo amb un editor de text, com el bloc de notes en MS Windows. Però no és recomanable editar manualment els arxius de dades.

Alineament

Per alinear les 13 seqüències, necessitem: carregar el paquet `DECIPHER`, llegir les seqüències i guardar-les en un objecte dins la sessió de R, i aplicar la funció `AlignSeqs()` per crear un nou objecte amb les seqüències alineades. Executa els comandaments següents:

```
library('DECIPHER')
senseAlinear <- readDNAStringSet('primats.fasta')
senseAlinear
```

```
## DNAStringSet object of length 13:
##      width seq                                               names
##  [1] 16521 GTTTATGTAGCTTAAACATACCC...ACACAACTACACCCGCACTAGC ENA|Y18001|Y18001...
##  [2] 16586 GTTTATGTAGCTTAAACCCACCC...CACAATTGTCACTCACACCCCT ENA|AJ309865|AJ30...
##  [3] 16564 GATCACGGGTCTATCACCCTATT...CCTTAAATAAGACATCTCGATG ENA|AY612638|AY61...
##  [4] 16472 GTTTATGTAGCTTAACTACCCAA...AGCCTATCCCCAAAGAGTCCCC ENA|X99256|X99256...
```

```
##  [5] 16389 GTTTATGTAGCTTATTCCATCCA...AACCCAAAAGACACCCCGCACA ENA|D38115|D38115...
##  ...   ... ...
##  [9] 16554 GTTTATGTAGCTTACCCCCTCAA...ACCCCAAAGACACCCCTACACA ENA|D38113|D38113...
## [10] 16561 GTTTATGTAGCTTACCCCCTCAA...ACCCCAAAGACACCCCTACACA ENA|X93335|X93335...
## [11] 16565 GATCACAGGTCTATCACCCTATT...CCTTAAATAAGACATCACGATG ENA|AM948965|AM94...
## [12] 16569 GATCACAGGTCTATCACCCTATT...CCTTAAATAAGACATCACGATG ENA|J01415|J01415...
## [13] 16571 GATCACAGGTCTATCACCCTATT...CCTTAAATAAGACATCACGATG ENA|AF347015|AF34...
```

Ara, l'objecte `senseAlinear` conté en la memòria de treball les 13 seqüències mitocondrials. Observa com en invocar el nom de l'objecte, ens apareix un resum del seu contingut. Si executes `names(senseAlinear)`, observaràs que els noms de les seqüències són innecessàriament llargs. Abans d'alinear, és convenient acurtar els noms, cosa que podem fer re-assignant el valor dels noms, respectant l'ordre en el qual estan les seqüències:

```
names(senseAlinear) <- c('Papio hamadryas', 'Macaca sylvanus', 'Macaca mulatta',
           'Hylobates lar', 'Pongo pygmaeus', 'Gorilla gorilla',
           'Gorilla gorilla gorilla', 'Pan paniscus', 'Pan troglodytes',
           'Pan troglodytes verus', 'H. sapiens neanderthalensis',
           'H. sapiens (Cambridge)', 'H. sapiens (Yoruba)')
```

```
alineades <- AlignSeqs(senseAlinear)
alineades
```

Observa que la funció `AlignSeqs()` no modifica l'objecte original (`senseAlinear`), sinó que en crea un altre. Necessitem assignar (`<-`) el resultat de la funció a un nou objecte (`alineades`) per tal de retenir en la memòria de treball l'alineament. Observaràs que la funció `AlignSeqs()` produeix alguns missatges sobre el procés. Podries suprimir eixos missatges amb l'argument opcional `verbose = FALSE`. Observa també que el nou objecte `alineades` té un aspecte diferent al de `senseAlinear`. Per tal de veure l'alineament complet, executa el comandament següent:

```
BrowseSeqs(alineades)
```

Hauria d'obir-se una nova pestanya en el navegador on es mostra l'alineament. Observa'l detingudament. Per què creus que algunes seqüències semblen tenir un fragment addicional en un extrem o en l'altre?

L'objecte `alineades` només existeix en la memòria de treball, i pot desaparèixer quan es tanque la sessió. Si vols conservar l'alineament que has generat en un arxiu FASTA, pots utilitzar el comandament següent:

```
writeXStringSet(alineades, 'alineades.fasta')
```

En l'aula virtual, o bé en aquest enllaç pots descarregar un arxiu FASTA amb un alineament alternatiu. Una volta descarregat, pots obrir-lo amb la funció `readDNAStringSet()`, i visualitzar-lo amb `BrowseSeqs()`. En què es diferencia de l'alineament que has obtingut tu? Continua treballant amb l'alineament que consideres més adient.

Reconstrucció filogenètica

Una vegada disposem d'un alineament, podem inferir les relacions filogenètiques entre les 13 seqüències. Hi ha molts mètodes diferents: màxima parsimònia, màxima versemblança, UPGMA, etc. Utilitzarem el mètode de Neighbor-Joining, basat en la matriu de distàncies genètiques entre les seqüències, perquè és un mètode ràpid.

Primer, carreguem el paquet `phangorn`:

```
library('phangorn')
```

La funció `NJ()` de `phangorn` que produirà l'arbre filogenètic s'aplica sobre una **matriu de distàncies**. Per calcular la matriu de distàncies entre les 13 seqüències nucleotídiques, podem aplicar la funció `dist.dna()`, la qual necessita com a arguments un alineament *guardat com a objecte de classe* ***DNAbin*** i un model d'evolució

molecular. Primer, creem un objecte de classe `DNAbin` amb el mateix alineament que tenim guardat en l'objecte `alineades` (o en el que hages triat), i que és de classe `DNAStringSet`:

```
aln <- as.DNAbin(alineades)
aln
```

En segon lloc, hauríem de triar un model d'evolució molecular, dels quals n'hi ha molts. Disposem de la funció `modelTest()` per determinar quin és el model d'evolució que millor s'ajusta a les nostres dades. Però tarda uns quants minuts en comprovar tots els models. Per estalviar aquest temps, la taula 2 mostra els cinc millors models d'evolució molecular del mitocondri dels primats. És a dir, els cinc models amb un *criteri d'informació bayesià* (BIC) menor. El criteri d'informació d'Akaike (AIC) és una altra manera de comparar els models.

Taula 2: Selecció de models d'evolució molecular que millor s'ajusten a l'alineament.

Model	logLikelihood	AIC	BIC
TPM2u+G(4)+I	-73032.87	146125.7	146357.5
TIM2+G(4)+I	-73029.03	146120.1	146359.6
TVM+G(4)+I	-73029.47	146122.9	146370.2
GTR+G(4)+I	-73026.01	146118.0	146373.0
HKY+G(4)+I	-73052.22	146162.4	146386.5

El model TPM2u+G(4)+I és el model de tres paràmetres de Kimura, també congut com K81 (Posada 2008), amb variació entre llocs de la taxa de substitucions (+G(4)) i amb una proporció de llocs invariables (+I). En qualsevol cas, la funció `dist.dna()` no ens ofereix totes les opcions (consulta `help(dist.dna)`). Calcularem les distàncies amb quatre models diferents i comprovarem quina repercusió té el model d'evolució molecular en la inferència filogenètica. Executa els comandaments següents:

```
distancies.K81  <- dist.dna(aln, model = 'K81',  gamma = 1.19)
distancies.T92  <- dist.dna(aln, model = 'T92',  gamma = 1.19)
distancies.F84  <- dist.dna(aln, model = 'F84',  gamma = 1.19)
distancies.TN93 <- dist.dna(aln, model = 'TN93', gamma = 1.19)
```

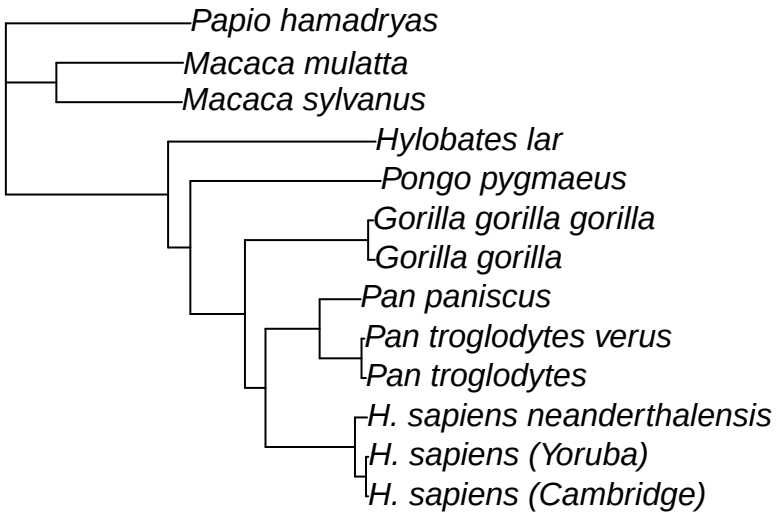
La distància genètica entre dues seqüències és una estimació del nombre mig de substitucions que s'han acumulat *en cada posició*, des de la seua divergència.

Procedim amb la inferència filogenètica pel mètode del Neighbor-joining. El comandament següent crea un arbre filogenètic, guardat en l'objecte `NJ.K81`. Executa'l i fes el mateix per obtenir els arbres corresponent a les altres tres distàncies.

```
NJ.K81 <- NJ(distancies.K81)
```

Pots utilitzar l'ordre `plot(NJ.K81)` per visualitzar l'arbre. El mètode Neighbor-joining produeix arbres *desarrelats*. Podem crear un arbre arrelat definint l'outgroup amb la funció `root()`:

```
NJ.K81.arrelat <- root(NJ.K81,
                       outgroup = c('Papio hamadryas', 'Macaca mulatta', 'Macaca sylvanus'))
plot(NJ.K81.arrelat)
```



Qüestionari

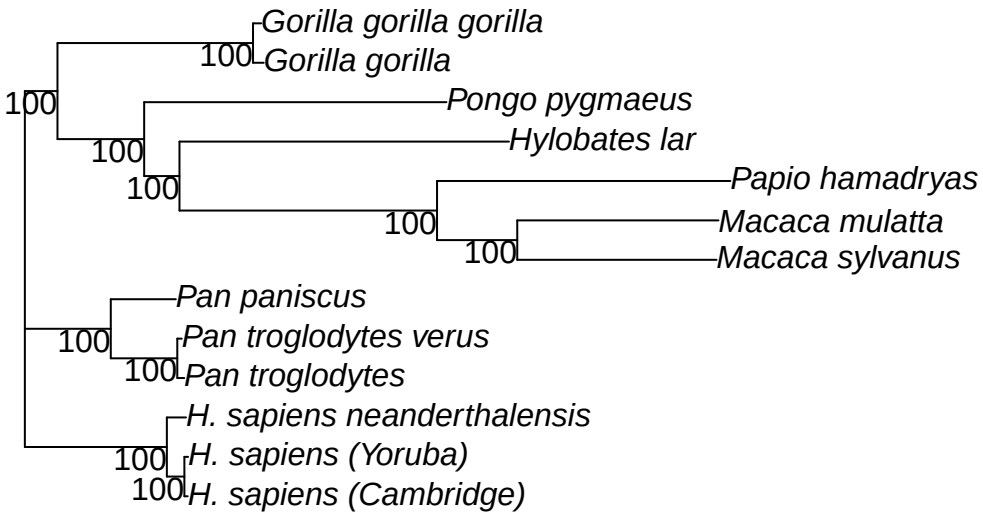
1. Assegura't d'*arrelar* tots els arbres amb el mateix outgroup abans de comparar-los. Determina quines distàncies produeixen arbres més semblants.
2. Consulta l'ajuda de la funció `plot.phylo()`, que és de fet la que s'executa quan invoquem `plot()` sobre un objecte de classe `phylo`, com `NJ.K81`. Veuràs que es poden canviar moltes coses. Comprova l'efecte de modificar els valors dels paràmetres `type`, `use.edge.length` i `align.tip.label`.

Bootstrap

El **bootstrap** és una tècnica estadística basada generar pseudo-rèpliques d'una mostra, per tal d'avaluar la robustesa d'una inferència o d'un estadístic. En el cas de la reconstrucció filogenètica les pseudo-rèpliques són alineaments alternatius generats per re-mostreig (amb repetició) de les columnes o posicions de l'alineament original. A partir de cada pseudo-rèplica es genera l'arbre filogenètic igual que l'original. Per últim es fa un recompte de la proporció d'arbres de bootstrap en què apareix cada una de les branques internes de l'arbre original. Aquesta proporció sol afegir-se en forma de percentatge a la representació gràfica d'un arbre. No informa de la validesa de l'arbre, sinó de la robustesa o el grau de confiança que mereix cada branca interna (clade) de l'arbre.

Intenta afegir valors de bootstrap a les branques dels arbres que has generat, seguint l'exemple següent:

```
funcio    <- function(x) NJ(dist.dna(as.DNAbin(x), model = 'K81'))
bootstrap <- bootstrap.phyDat(as.phyDat(aln), funcio, bs = 500)
plotBS(NJ.K81, bootstrap)
```



Bibliography

Posada, D. 2008. "jModelTest: Phylogenetic Model Averaging." *Molecular Biology and Evolution* 25 (7): 1253–56. https://doi.org/10.1093/molbev/msn083.

Schliep, K. P. 2011. "phangorn: Phylogenetic Analysis in R." *Bioinformatics* 27 (4): 592–93. https://doi.org/10.1093/bioinformatics/btq706.

Wright, Erik S. 2016. "Using DECIPHER V2.0 to Analyze Big Biological Sequence Data in R." *The R Journal* 8 (1): 352–59. https://doi.org/10.32614/RJ-2016-025.