Manipulación de árboles filogenéticos

March 15, 2016

Seminario: Análisis Comparativo Filogenético

Prof: Marcelo Araya-Salas, PhD

Escuela de Biología

Universidad de Costa Rica

I-2016

Primero instalamos y cargamos los paquetes que vamos a usar

```
#instalar paquetes (SOLO SI AUN NO ESTAN INSTALADOS!!)
install.packages("ape")
install.packages("phytools")

#cargar paquetes
library("ape")
library("phytools")
```

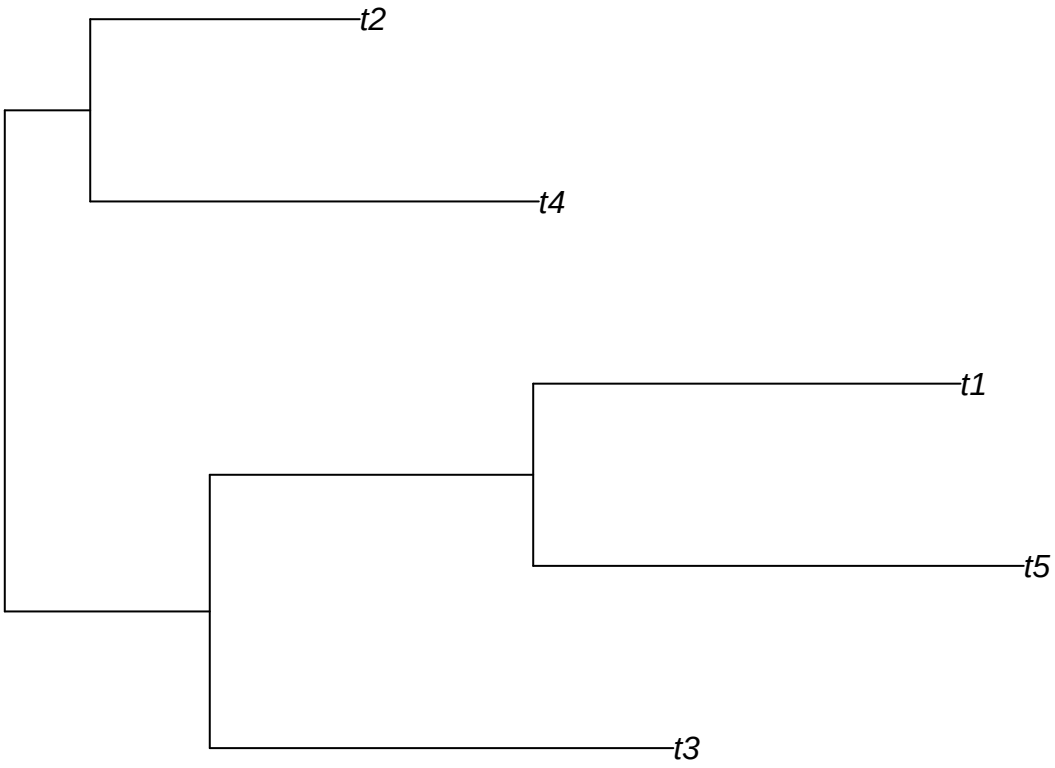
Recuerden que siempre pueden ver la ayuda asociada a un paquete (o función)

```
?ape
```

Elementos de un árbol filogenético clase `phylo` –

Generemos un árbol aleatoriamente con la función `rtree`. Esta función puede generar un árbol con un determinado número de especies (`n`)

```
#primero ajustar los margenes del grafico
opar <- par
par(mar = rep(1, 4))

#crear árbol
set.seed(10)
arb <- rtree(n = 5, rooted = TRUE)

#graphicar árbol
plot(arb)
```



Los árboles son guardados como objetos de clase `phylo` en el paquete `ape`. Hay varias otras clases para guardar árboles que son usadas por otros paquetes, todas con un formato similar. Por ahora trabajaremos con la clase `phylo`.

```
class(arb)
```

```
## [1] "phylo"
```

Una de las ventajas de trabajar con árboles filogenéticos en `R` es que se puede acceder fácilmente a todos los elementos. Con la función `str` podemos ver la estructura de los objetos, en este caso nuestro árbol aleatorio `tr`

```
str(arb)
```

```
## List of 4
##  $ edge       : int [1:8, 1:2] 6 7 7 8 8 6 9 9 7 1 ...
##  $ tip.label  : chr [1:5] "t3" "t5" "t1" "t4" ...
##  $ edge.length: num [1:8] 0.272 0.616 0.43 0.652 0.568 ...
##  $ Nnode      : int 4
##  - attr(*, "class")= chr "phylo"
##  - attr(*, "order")= chr "cladewise"
```

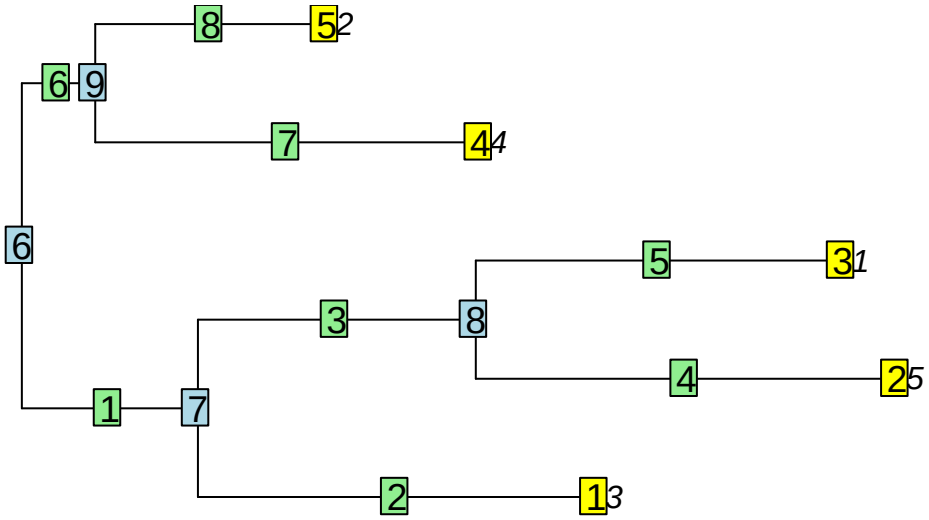
Esto nos dice que el objeto de clase `phylo` en realidad es una lista (como cualquier otra lista en `R`) con 4 elementos:

- edge: una matriz de 2 columnas con el índice de los nodos que son unidos por cada rama (edge).
- tip.label: un vector de caracteres con los nombres de los nodos terminales u "hojas" (generalmente las especies)

- edge.length: un vector numérico con el largo de ramas (opcional)
- Nnode: numero de nodos internos (sin contar las "hojas"). Incluye la raíz del árbol si el árbol es enraizado.

Podemos visualizar la posición de estos elementos:

```
plot(arb)

# añadir etiquetas a las ramas
edgelabels(cex = 1.2)

# añadir etiquetas a los nodos internos
nodelabels(cex = 1.2)

# añadir etiquetas a los nodos externos (hojas)
tiplabels(cex = 1.2)
```



Notemos que las etiquetas de los "tips" corresponden al número de nodo, no a los nombres de las hojas.

–

Podemos ver que los números en la matriz `edge` de nuestro árbol coincide con las etiquetas de los nodos con los que conectan.

```
#juntar "edge" y "edge.length" en una sola matriz10
cbind(arb$edge, arb$edge.length)
```

```
##      [,1] [,2]      [,3]
## [1,]    6    7 0.2723051
## [2,]    7    1 0.6158293
## [3,]    7    8 0.4296715
## [4,]    8    2 0.6516557
## [5,]    8    3 0.5677378
## [6,]    6    9 0.1135090
## [7,]    9    4 0.5959253
## [8,]    9    5 0.3580500
```

Por convención las hojas del árbol estan numeradas de `1:n` para `n` hojas y los nodos internos van de `n+1` a `n+m` para `m` nodos internos.

–

También podemos revisar que tipo de árbol estamos utilizando. Si árbol es ultramétrico

```
is.ultrametric(arb)
```

```
## [1] FALSE
```

Si es totalmente bifurcado (sin politomías)

```
is.binary.tree(arb)
```

```
## [1] TRUE
```

O si es enraizado

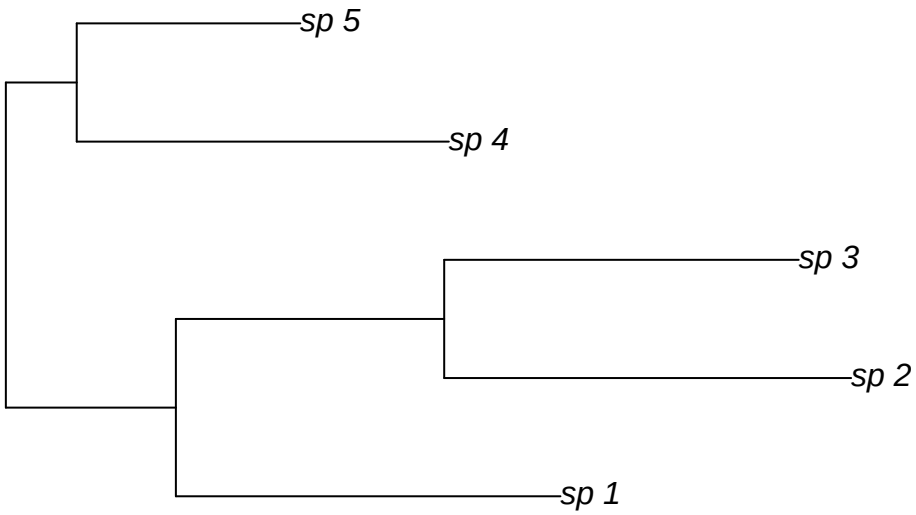
```
is.rooted(arb)
```

```
## [1] TRUE
```

Modificar árboles –

Estos elementos se pueden modificar de igual forma que modificamos cualquier otro objeto de R con estructura similar. Podemos poner el nombre de las especies (o unidades taxonómicas operacionales).

```
spp <- paste("sp", 1:5)

arb$tip.label <- spp

plot(arb)
```



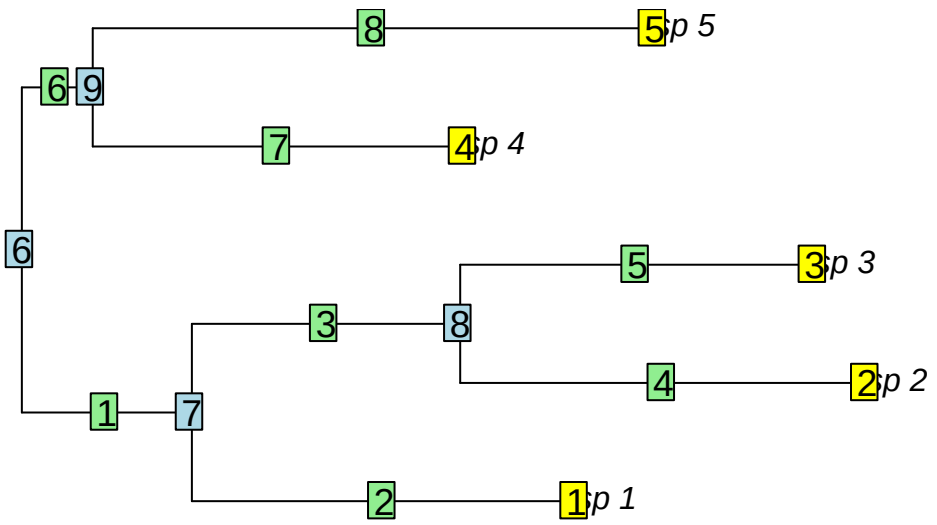
También podríamos cambiar el largo de las ramas. Cambiemos el largo de la rama que lleva a `sp5`. Para esto debemos buscar en la matriz `edge` para ver cual rama une los nodos 9 y 5.

```
cbind(arb$edge, arb$edge.length)
```

```
##      [,1] [,2]      [,3]
## [1,]    6    7 0.2723051
## [2,]    7    1 0.6158293
## [3,]    7    8 0.4296715
## [4,]    8    2 0.6516557
## [5,]    8    3 0.5677378
## [6,]    6    9 0.1135090
## [7,]    9    4 0.5959253
## [8,]    9    5 0.3580500
```

Aquí vemos que el octavo elemento de `edge.length` el que une los nodos 9 y 5. Podemos sobreescribir ese valor para así aumentar el largo de la rama.

```
arb$edge.length[8] <- 0.9

arb$tip.label <- spp

plot(arb)

# añadir etiquetas a las ramas
edgelabels(cex = 1.2)

# añadir etiquetas a los nodos internos
nodelabels(cex = 1.2)

# añadir etiquetas a los nodos externos (hojas)
tiplabels(cex = 1.2)
```

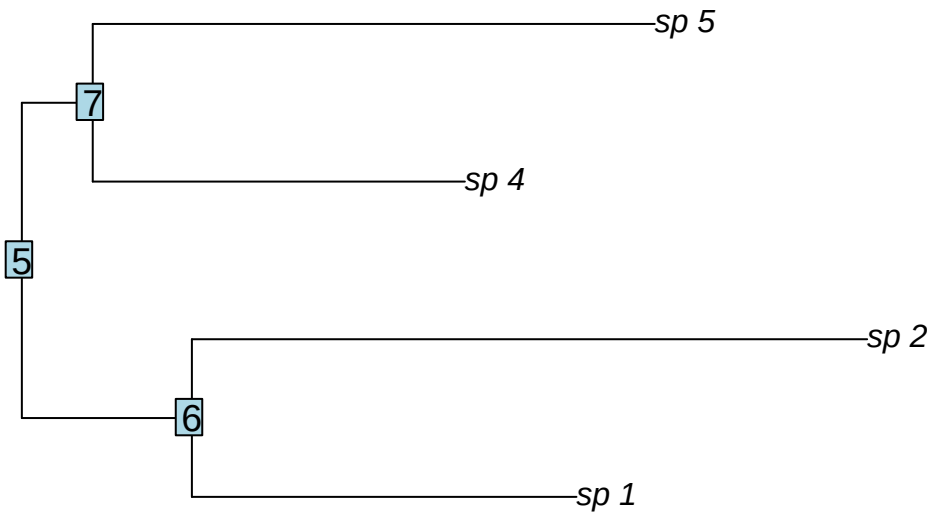


De forma similar podríamos sobreescribir las etiquetas de los nodos y las ramas.

–

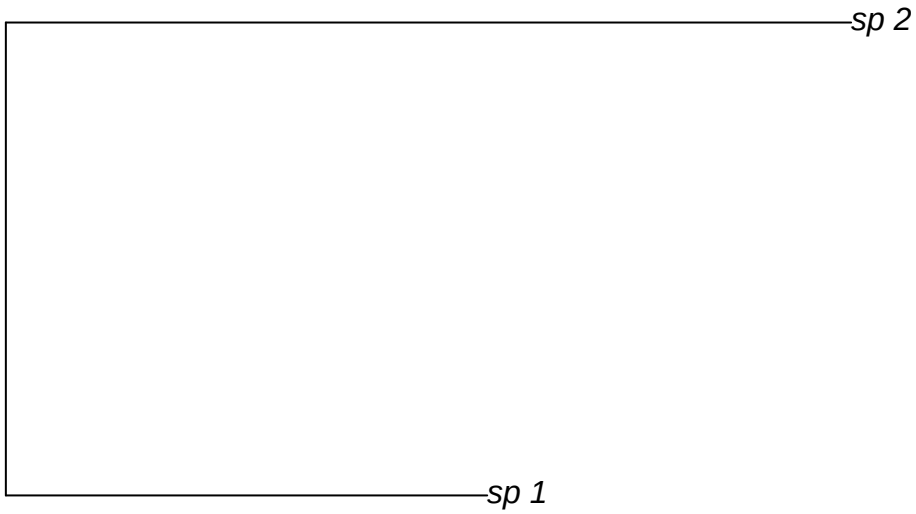
También podemos remover hojas:

```
arb <- drop.tip(arb, 3)

plot(arb)

nodelabels(cex = 1.2)
```



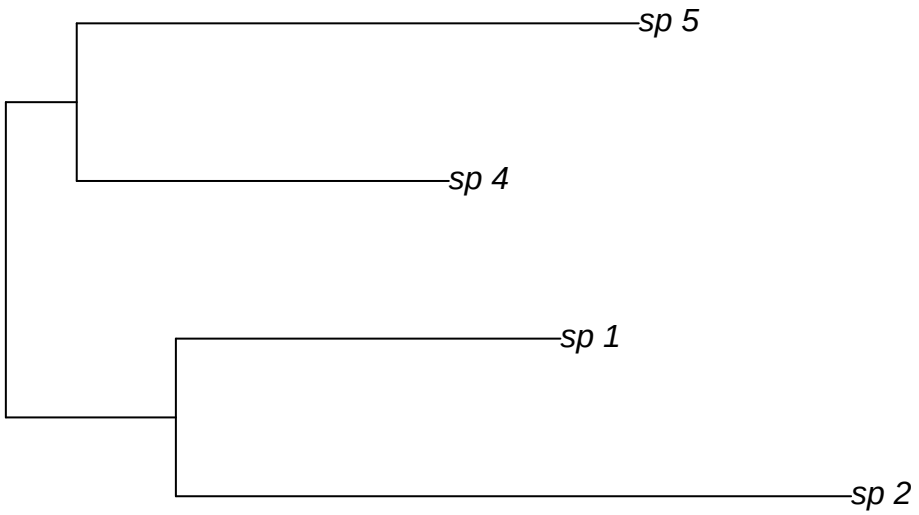
Extraer subclados completos

```
subarb<-extract.clade(arb,node = 6)
plot(subarb)
```

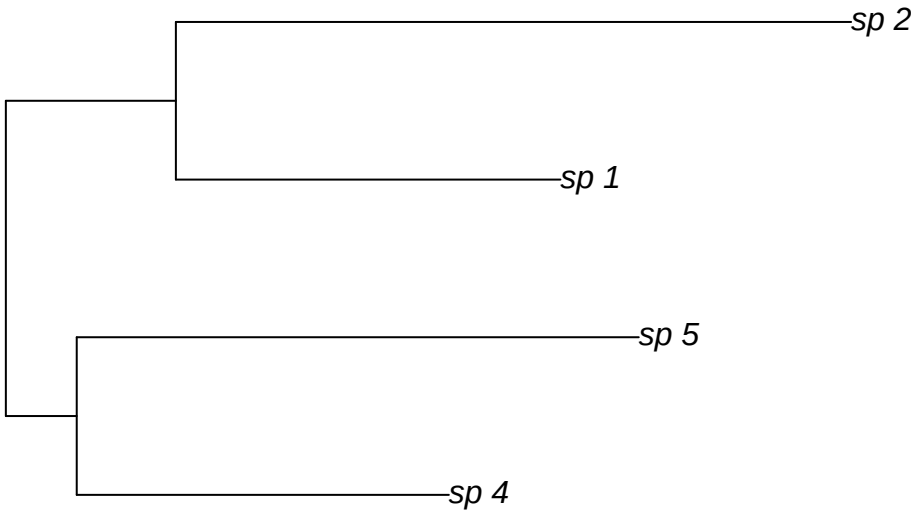


O rotar clados hermanos:

```
arb <- rotate(arb, 6)

plot(arb)
```



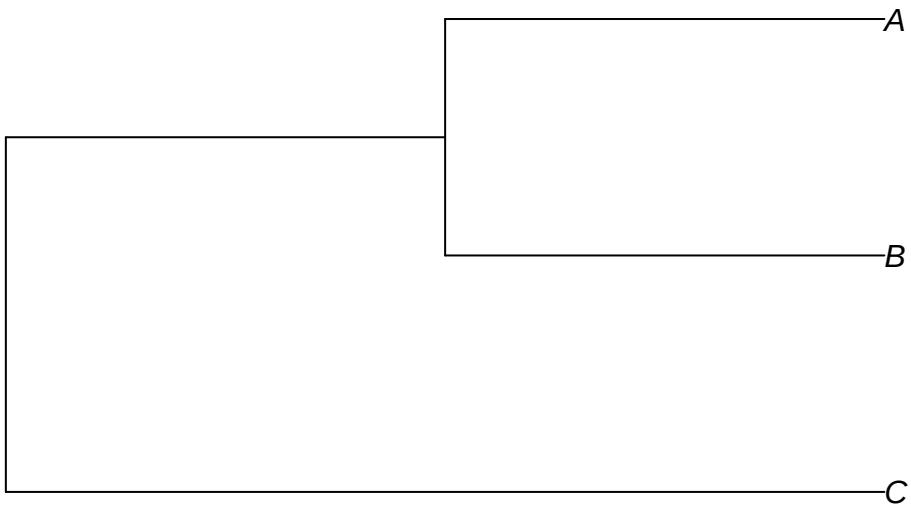

```
arb <- rotate(arb, 5)

plot(arb)
```



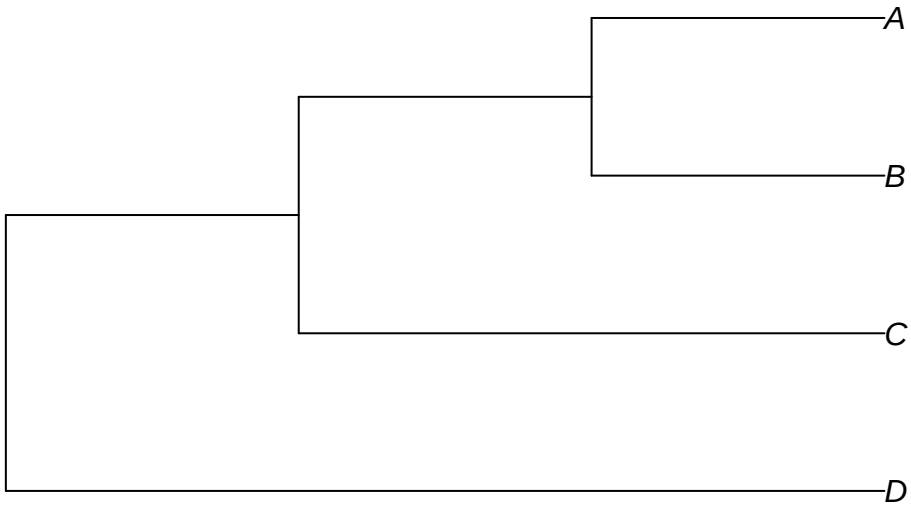
Los árboles pueden ser escritos manualmente siguiendo el formato Newick. Este formato representa la anidación de clados usando paréntesis.

```
plot(read.tree(text = "(B,A);"))
```

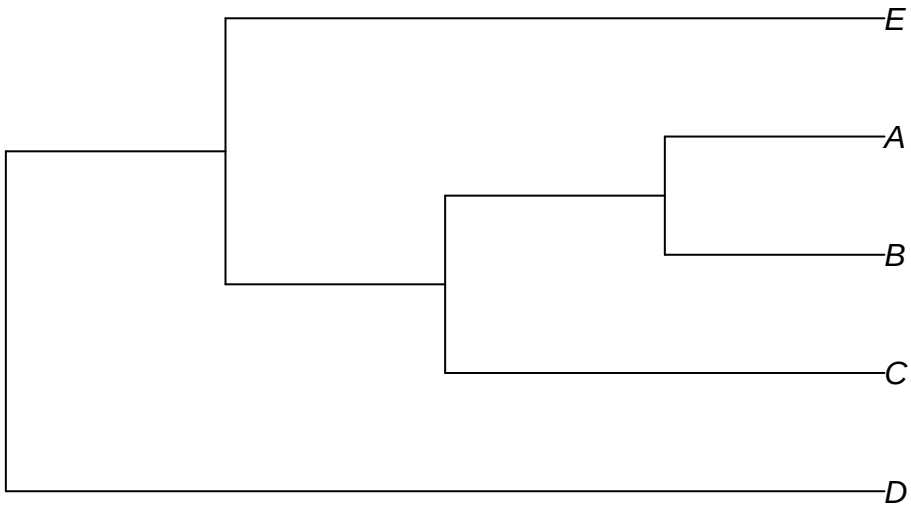

```
plot(read.tree(text = "(C,(B,A));"))
```




```
plot(read.tree(text = "(D,(C,(B,A)));"))
```



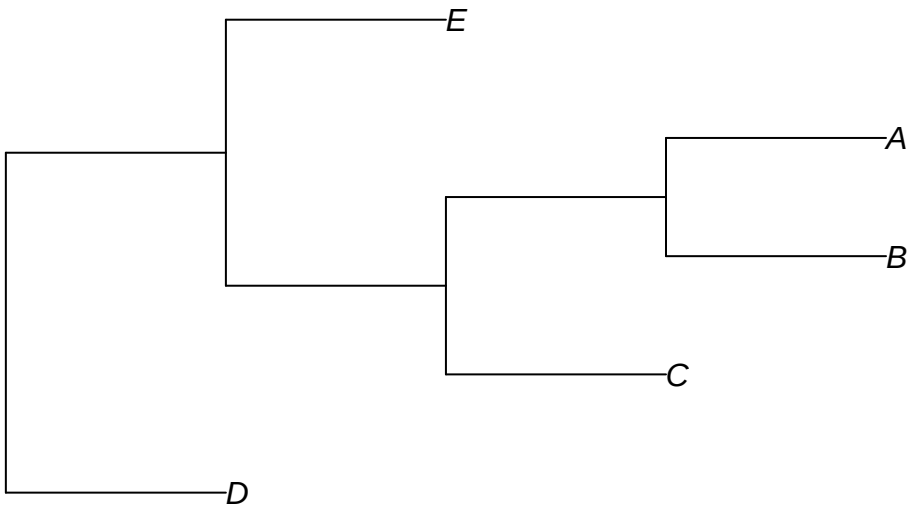
```
plot(read.tree(text = "(D,((C,(B,A)),E));"))
```



Noten q el árbol es graficado como un árbol ultramétrico no datado (sin largo de ramas). También se puede añadir el largo de las ramas usando un `":"` luego de la etiqueta de las hojas.

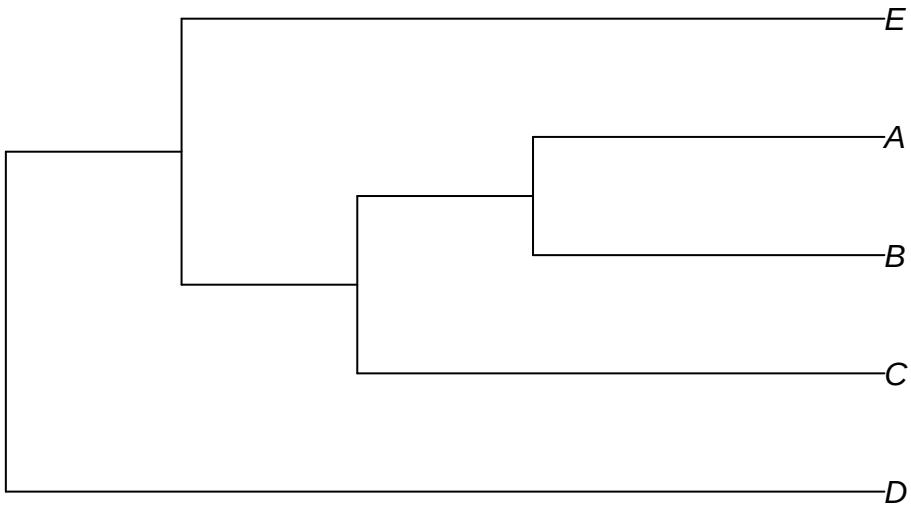
Con largos de rama iguales

```
plot(read.tree(text = "(D:1,((C:1,(B:1,A:1):1):1,E:1):1);"))
```



Con largos de rama variable

```
plot(read.tree(text = "(D:5,((C:3,(B:2,A:2):1):1,E:4):1);"))
```



Generalmente no es necesario escribir los árboles manualmente. Este ejemplo es solo para tener una idea de como son codificados los árboles.

–

Importar árboles de otros programas –

Los árboles filogenéticos son generalmente guardados como archivos con extension `.tre` o `.nex`.. En R podemos guardar los árboles en estos formatos.

```
#primero usemos una carpeta temporal
setwd(tempdir())

#excrbir en formato nexus (.nex)
writeNexus(arb,"árbol.nex")

#excrbir en formato tree (.tre)
write.tree(arb,"árbol.tre")

#en esta carpeta puede econtrar los archivos
getwd()
```

o importarlos de archivos previamente creados.

```
#leer en formato nexus (.nex)
arb <- read.nexus("árbol.nex")

#excrbir en formato tree (.tre)
arb <- read.tree("árbol.tre")
```

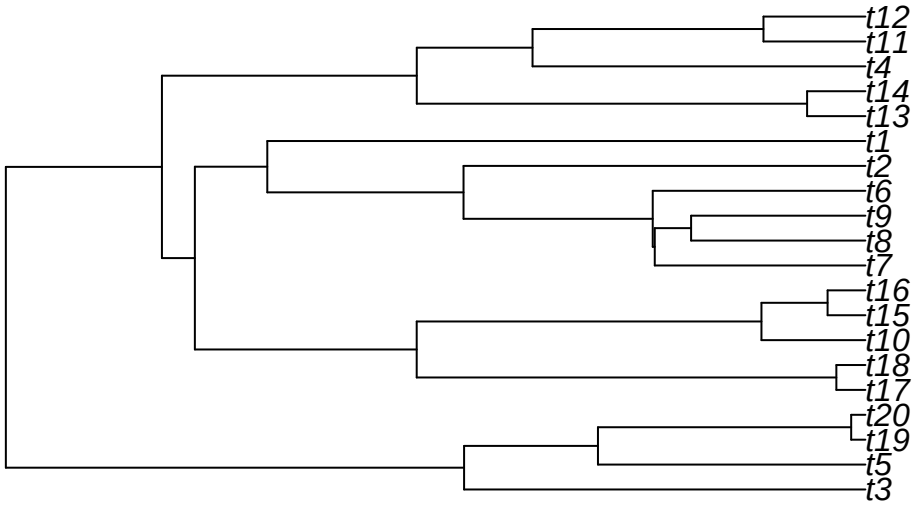
Graficar filogenias –

Existen muchas formas de graficar árboles filogenéticos. Los paquetes mas útiles para esto son `ape` y (sobre todo) `phytools`. Este último sobresale por la cantidad de opciones para representar la evolución de rasgos, los cuales son mapeados sobre las filogenias. Sin embargo por ahora solo trabajaremos con representaciones de los árboles.

–

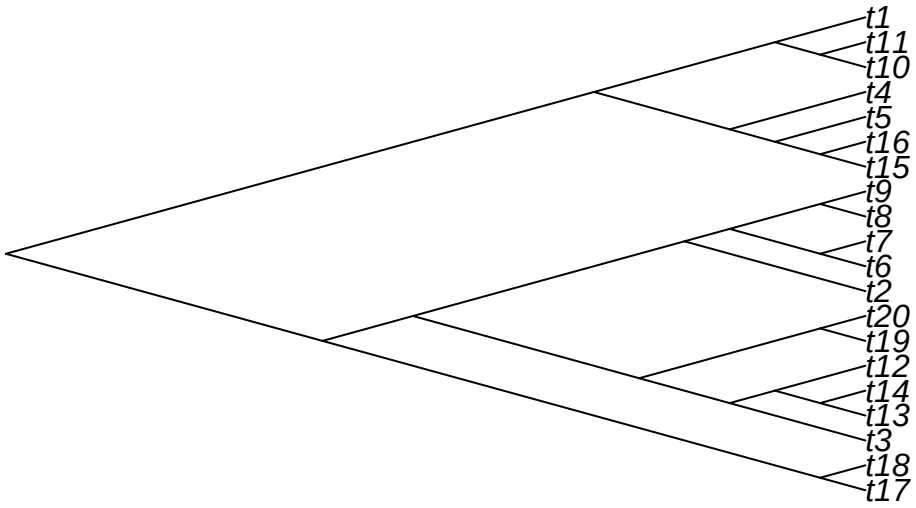
Como ya hemos visto los árboles pueden ser graficados con la raíz a la izquierda y las hojas a la derecha. En este caso usamos la función `pbtree` del paquete `phytools`, la cual genera un árbol ultramétrico.

```
#primero crear un árbol aleatorio
arb <- pbtree(n = 20)

plot(arb, type = "phylogram")
```



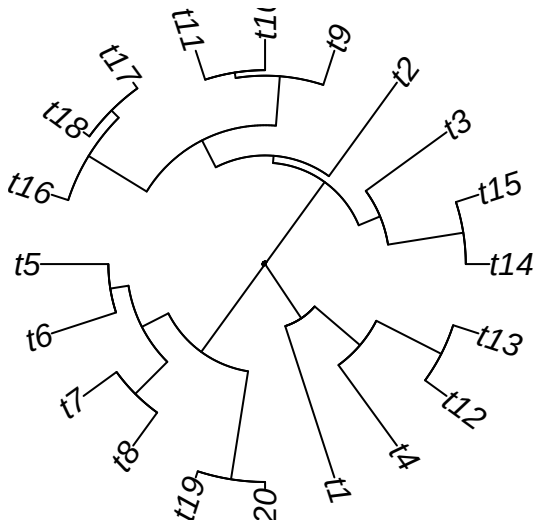
Tambien podemos crear un cladograma

```
#primero crear un árbol aleatorio
arb <- pbtree(n = 20)

plot(arb, type = "cladogram",use.edge.length = FALSE)
```



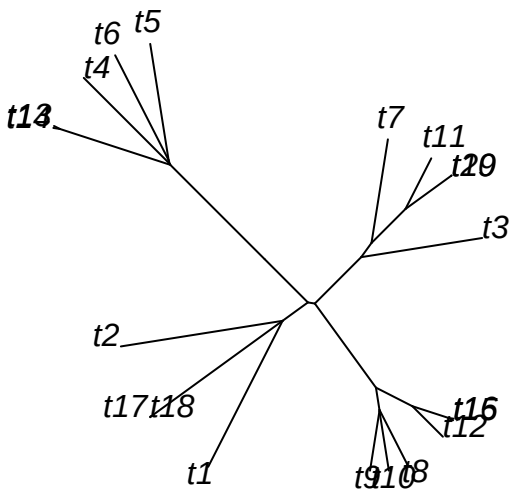
Un árbol tipo abanico

```r
#primero crear un árbol aleatorio
arb <- pbtree(n = 20)

plot(arb, type = "fan",use.edge.length = TRUE)
```



O un árbol no enraizado

```r
#primero crear un árbol aleatorio
arb <- pbtree(n = 20)

plot(arb, type = "unrooted",use.edge.length = TRUE)
```



Ejemplo: Manipulación del árbol de psitácidos

Este es un ejemplo de como se pueden usar las funciones con que hemos trabajado en la preparación de árboles para análisis comparativos. El siguiente código lo usé para preparar la filogenia de psitácidos para los análisis comparativos en un estudio sobre la evolución de las vocalizaciones ((Medina-García et al. 2015))

Primero debemos leer la filogenia. La podemos bajar directamente de internet

```
psitasidos = read.nexus("http://marceloarayasalas.weebly.com/uploads/2/5/5/2/25524573/psitacidos.tre")
```

O la podemos leer de un archivo en nuestra computadora

```
psitasidos = read.nexus("psitacidos.tre")
```

Primero es importante revisar que tipo de árbol tenemos (ultramétrico, enraizado, etc)

```
#es ultrametrico?
is.ultrametric(psitasidos)
```

```
## [1] TRUE
```

```
#es binario (sin politomias)?
is.binary.tree(psitasidos)
```

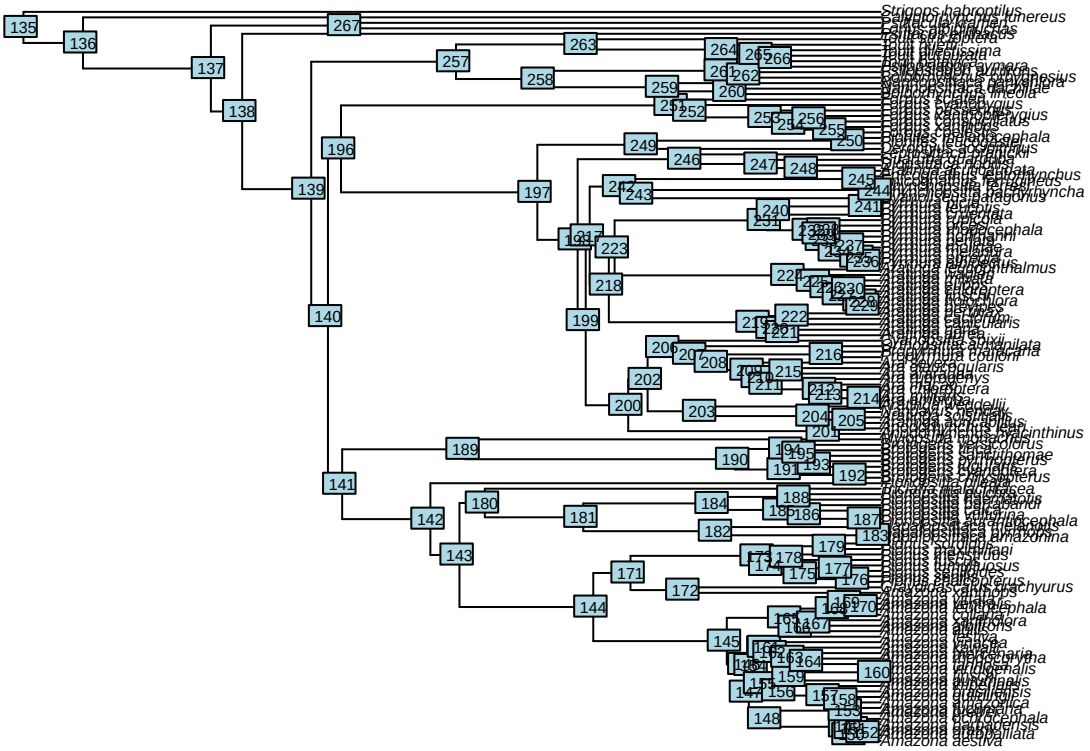
```
## [1] TRUE
```

```
#es enraizado?
is.rooted(psitasidos)
```

```
## [1] TRUE
```

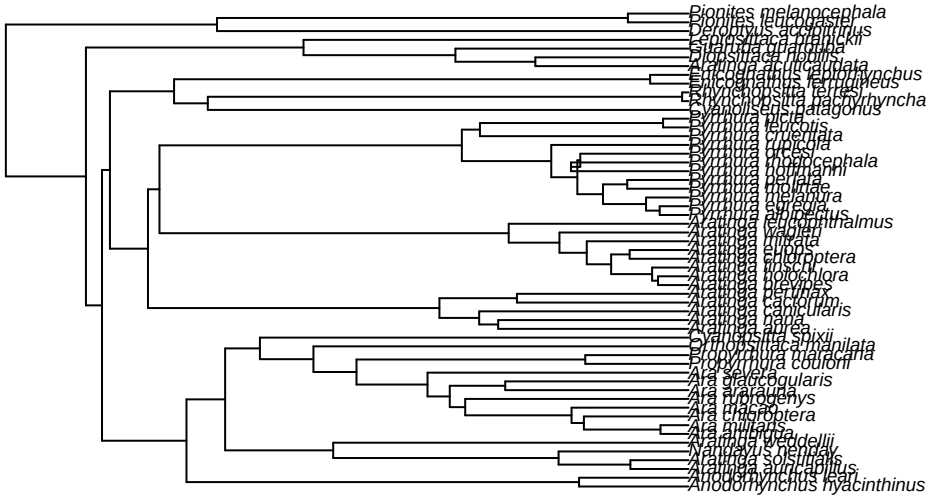
Y luego revisamos visualmente que la filogenia tenga sentido

```
par(mar = rep(1, 4))

plot(psitasidos, cex = 0.5)
nodelabels(cex = 0.5)
```



En este estudio estabamos interesados en la tribu Arini, un subclado de psitácidos neotropicales. Así que debemos extraer ese subclado

```
Arini<-extract.clade(psitasidos,node = 197)
plot(Arini, cex = 0.6)
```



Luego debemos asegurarnos que los nombres de especies sean los correctos (e.i. mas actuales). En este caso se deben renombrar algunas especies

```
print(Arini$tip.label)
```

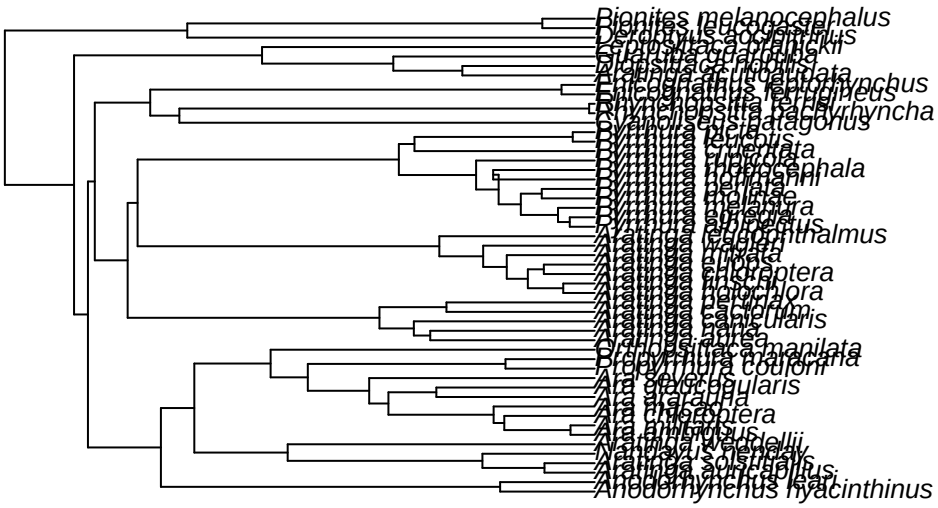
```
##  [1] "Anodorhynchus_hyacinthinus" "Anodorhynchus_leari"
##  [3] "Aratinga_auricapillus"      "Aratinga_solstitialis"
```

```
##  [5] "Nandayus_nenday"            "Aratinga_weddellii"
##  [7] "Ara_ambigua"                "Ara_militaris"
##  [9] "Ara_chloroptera"            "Ara_macao"
## [11] "Ara_rubrogenys"             "Ara_ararauna"
## [13] "Ara_glaucogularis"          "Ara_severa"
## [15] "Propyrrhura_couloni"        "Propyrrhura_maracana"
## [17] "Orthopsittaca_manilata"     "Cyanopsitta_spixii"
## [19] "Aratinga_aurea"             "Aratinga_nana"
## [21] "Aratinga_canicularis"       "Aratinga_cactorum"
## [23] "Aratinga_pertinax"          "Aratinga_brevipes"
## [25] "Aratinga_holochlora"        "Aratinga_finschi"
## [27] "Aratinga_chloroptera"       "Aratinga_euops"
## [29] "Aratinga_mitrata"           "Aratinga_wagleri"
## [31] "Aratinga_leucophthalmus"    "Pyrrhura_albipectus"
## [33] "Pyrrhura_egregia"           "Pyrrhura_melanura"
## [35] "Pyrrhura_molinae"           "Pyrrhura_perlata"
## [37] "Pyrrhura_hoffmanni"         "Pyrrhura_rhodocephala"
## [39] "Pyrrhura_orcesi"            "Pyrrhura_rupicola"
## [41] "Pyrrhura_cruentata"         "Pyrrhura_leucotis"
## [43] "Pyrrhura_picta"             "Cyanoliseus_patagonus"
## [45] "Rhynchopsitta_pachyrhyncha" "Rhynchopsitta_terresi"
## [47] "Enicognathus_ferrugineus"   "Enicognathus_leptorhynchus"
## [49] "Aratinga_acuticaudata"      "Diopsittaca_nobilis"
## [51] "Guaruba_guarouba"           "Leptosittaca_branickii"
## [53] "Deroptyus_accipitrinus"     "Pionites_leucogaster"
## [55] "Pionites_melanocephala"
```

```
#cambiar el nombre de especies
Arini$tip.label<-gsub("Pionites_melanocephala","Pionites_melanocephalus",Arini$tip.label)
Arini$tip.label<-gsub("Rhynchopsitta_terresi","Rhynchopsitta_terrisi",Arini$tip.label)
Arini$tip.label<-gsub("Ara_ambigua","Ara_ambiguus",Arini$tip.label)
Arini$tip.label<-gsub("Ara_severa","Ara_severus",Arini$tip.label)
```

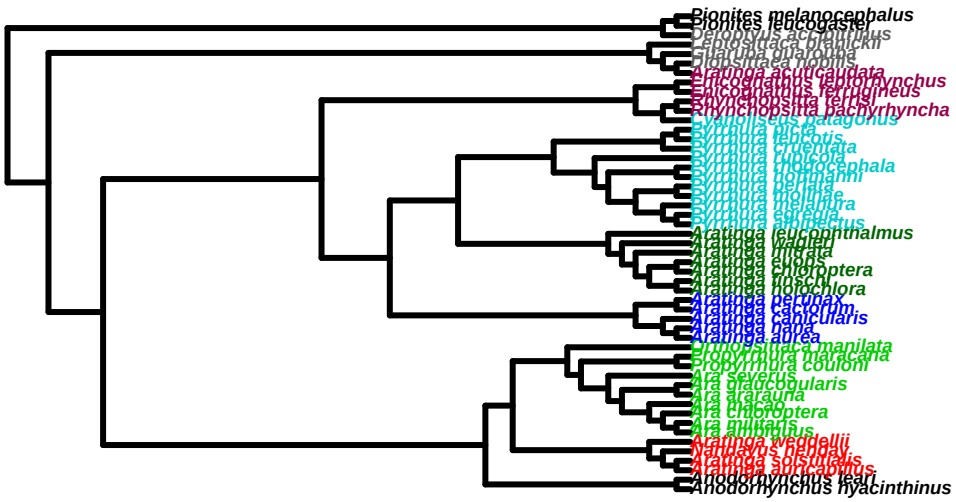
Tambien se removimos las especies para las cuales no teniamos datos

```
Arini51sp<-drop.tip(Arini,c("Ara_rubrogenys","Cyanopsitta_spixii","Aratinga_brevipes",
                            "Pyrrhura_orcesi"))

plot(Arini51sp,cex=.8)
```



En este gráfico usamos colores para resaltar ls diferentes generos

```
plot.phylo(Arini51sp,adj=0,use.edge.length=FALSE,font=4,cex=0.6,
           tip.color=c(rep(1,2),rep(2,4),rep(3,10),rep(4,5),rep("#006600",7),
            rep("#00CCCC",12),rep("#99004C",5), rep("#606060",4),rep(1,2)),edge.width=3)
```



Es una buena idea guardar el árbol una vez que esta listo

```
write.tree(Arini51sp,"Arini51sp.tre")
```

–

Ahora que ya saben manipular filogenias pueden comenzar a buscar sus propios árboles y "jugar" con ellos. Estos son algunos repositorios en linea donde pueden bajar arboles filogenéticos:

- `treebase`: Filogenias de cualquier grupo taxonomico (El paquete de R `treebase` permite bajar las filogenias directamente en `R`)
- `10ktrees`: Filogenias de mamíferos.
- `birdtree`: Filogenias de aves.

Referencias

- Garamszegi, Z. (Ed.). 2014. Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice (1st ed.). Berlin, Heidelberg: Springer.
- Nunn, C.L.C. (2011). The comparative approach in evolutionary anthropology and biology. (1st ed.). University of Chicago Press.
- Paradis, E. (2012). Analysis of Phylogenetics and Evolution with R.
- Revell LJ (2012) Phytools: an R package for phylogenetic comparative biology (and other things). Methods Ecol Evol 3:217-223.