

Dessine-moi un arbre :
représentation graphique
des arbres phylogénétiques
avec le package ape

NATTIER, Romain

*Muséum national d'Histoire naturelle
UMR 7205 OSEB*



28 janvier 2011

**Dessine-moi un arbre:
représentation graphique des arbres phylogénétiques avec le package *ape***

Pourquoi utiliser R pour manipuler des arbres phylogénétiques ?

- logiciel gratuit et a code source ouvert
- multiplateforme (Linux, Windows et Macintosh)
- grande communauté
- répétable (scripts)

Commandes sous R en gras+it+violet

Ex: ***plot(tr)***

Présentation du package APE

BIOINFORMATICS APPLICATIONS NOTE Vol. 20 no. 2 2004, pages 289–290
DOI: 10.1093/bioinformatics/btg412



APE: Analyses of Phylogenetics and Evolution in R language

Emmanuel Paradis^{1,*}, Julien Claude¹ and Korbinian Strimmer²

¹Laboratoire de Paléontologie, Paléobiologie and Phylogénie, Institut des Sciences de l'Évolution, Université Montpellier II, F-34095 Montpellier cédex 05, France and
²Department of Statistics, University of Munich, Ludwigstrasse 33, D-80539 Munich, Germany

Received on April 11, 2003; revised on July 11, 2003; accepted on July 29, 2003

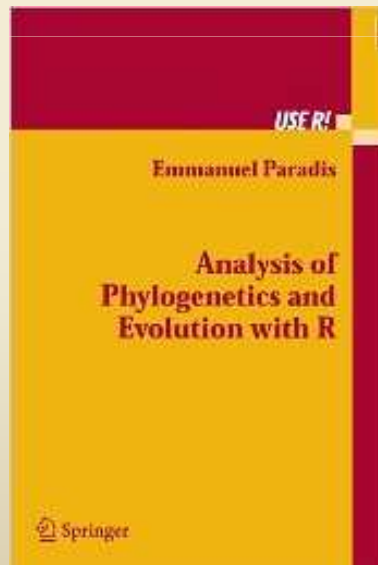


Table 1. Special functions available in APE 1.1

Application	Available commands
Input/output	<code>read.dna</code> , <code>write.dna</code> , <code>read.nexus</code> , <code>write.nexus</code> , <code>read.tree</code> , <code>write.tree</code> , <code>read.GenBank</code>
Graphics	<code>add.scale.bar</code> , <code>plot.mst</code> , <code>plot.phylo</code> , <code>plot.skyline</code> , <code>lines.skyline</code> , <code>ltt.plot</code>
Tree manipulation	<code>bind.tree</code> , <code>drop.tip</code> , <code>is.binary.tree</code> , <code>is.ultrametric</code>
Comparative method	<code>compar.gee</code> , <code>compar.lynch</code> , <code>pic</code> , <code>vcv.phylo</code>
Diversification	<code>birthdeath</code> , <code>cherry</code> , <code>diversi.gof</code> , <code>diversi.time</code> , <code>gamma.stat</code>
Population genetics	<code>branching.times</code> , <code>coalescent.intervals</code> , <code>collapsed.intervals</code> , <code>find.skyline.epsilon</code> , <code>heterozygosity</code> , <code>skylineplot</code> , <code>skyline</code> , <code>theta.h</code> , <code>theta.k</code> , <code>theta.s</code>
Molecular dating	<code>chronogram</code> , <code>ratogram</code> , <code>NPRS.criterion</code>
Miscellaneous	<code>all.equal.phylo</code> , <code>balance</code> , <code>base.freq</code> , <code>dist.dna</code> , <code>dist.gene</code> , <code>dist.phylo</code> , <code>GC.content</code> , <code>klastorin</code> , <code>mantel.test</code> , <code>mst.summary.phylo</code>
Data sets	<code>bird.families</code> , <code>bird.orders</code> , <code>hivtree</code> , <code>landplants</code> , <code>opsin</code> , <code>woodmouse</code> , <code>xenarthra</code>

Plusieurs formats:

- Parenthetic Format (= Newick or New Hampshire format)

```
((((Lépidoptères:1,Diptères:1),Hyménoptères:1):1,Coléoptères:1),((Dictyoptères:1,Orthoptères:1),Hémiptères:1):1,Odonates:1):1,Collemboles:1):1;
```

read.tree("nom_du_fichier")

- Nexus Format (Madison *et al.*, 1997)

- comprised a "blocks" such as the taxa block, data block, sets block, trees block, PAUP block, MrBayes block
- used by software such as MrBayes and PAUP

```
#NEXUS
```

```
begin data;  
dimensions ntax=61 nchar=2702;  
format datatype=dna interleave=no missing=?;
```

```
MATRIX
```

```
Ruspolia
```

```
AATCTACTTATTCGTCGCAATTAGGACAACCCGATTTTTAATTGGCGATGATCAGATTTATAATGTAATTGTTACTGCCACTCTTTATAATAATTTCTTCATAGTAATACCTATTATAATTGGAGGATTCCGTAATTGATTAGTTCCTTTAATGTTAGGAGCTCCT  
GATATAGCATTCCCTCGAATAAATAATATAAGTTTTGATTACTCCACCATCGCTAACCTTTTACTCGCCAGAAGATTAGTGGAAAGTGGCGCGGGTACAGGTTGAACGGTTTATCCTCCGCTTTCTGCAGGAATCGCCATGCAGGAGCCTCAGTTGACTAG  
CTATCTTTTCTTTACATTTAGCAGGGGTTTCTTCAATTTCTGGGGCCGTAACCTTCACTACTACAACAATTAATACGAGCTCCAGGTATATCATTAGATCAAAACACCTTTATTTGTTGAGCTGTCGCTATTACAGCTCTCTTATTACTTCTCTCCCTTCCCGTTCTTG  
CCGGAGCAATTACAATACTTTTAAACAGACCGTAATTTAAATACATCTTTCTTGATCCTGCAGGTGGGGGAGATCCTATTCTTTATCAACACTTATTTTGATTCTTTGGACACCCAGAAGTTTATATTTAATTTTACCAGGATTTGGAATAATTTCCCATATTATTAGTC  
AAGAAAAGTGGTAAAAGGAAGCCTTTGGAACATTAGGAATAATTTATGCTATATT
```

```
Locusta_migratoria
```

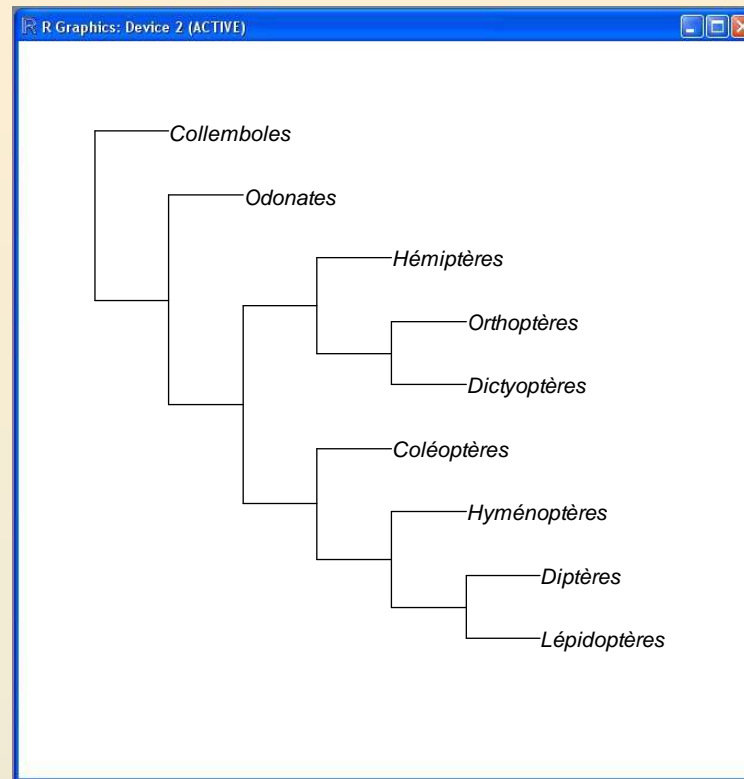
```
AGAATAATTATTTCGAGCTGAATTAGTCAACCAGGAACAATAATTAAACGATGATCAACTATATAATGTAATCATTACAGCACAGCATTGTTATAATTTCTTCATGGTTATGCCAATTATAATTGGAGGATTCCGAAATTGATTAGTACCATTAATAATTGGAGCTCC  
AGATATAGCTTTTCCACGAATAAATAACATAAGATTTTATTATACCACCATCATTAAACACTCCTACTAATGTCTTCTGTAGTAGATAATGGAGCTGGTACAGGATGAACAGTATATCCTCCACTAGCCAGAGTCATTGCTCATAGAGGAGCTTCTGTAGATTTAGCA  
ATTTTTTTCATTACATCTAGCAGGTGTTTCTCAATTTTGGAGCTATTAATTTTCACTACAACAGCAATCAATATACGATCAAATAATATAACCCTTGATCAAACACCATTTATTTGTTGATCAGTAGCAATTACAGCCTTATTACTTTTATTATCATTACCAGTATTAGCTG  
GAGCAATTACTATATTATTAACCTGATCGAAACCTTAATACGTCATTCTTCACCCGGCAGGAGGGTGACCCAATTTTATATCAACACTTATTTTGATTCTTTGGTCAACCAGAAGTTTATATTTAATTTTACCAGGATTTGGTATTATTTTACATATCGTTTGTCAAG  
AAAGAGGAAAAATTGAATCATTTGGAACAATTGGTATAATTTATGCAATATTATC...
```

```
; END;
```

read.nexus("nom_du_fichier")

Lire et introduire un arbre

```
tr<-read.tree("Hexapoda.txt")  
plot(tr)
```

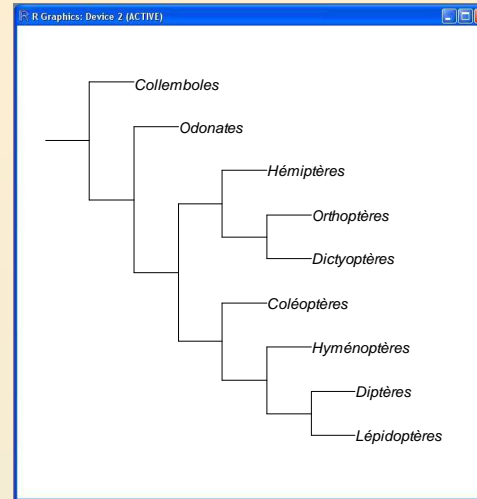


- Read Character Data In NEXUS Format => *read.nexus.data*
- Write Tree File in Nexus Format => *write.nexus*
- Write Tree File in Parenthetic Format => *write.tree*

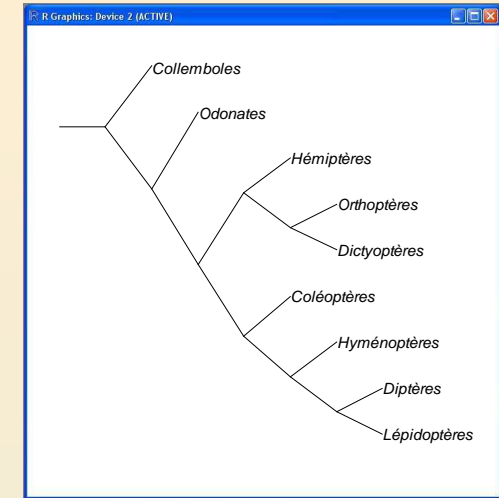
Lire et introduire un arbre

type of tree

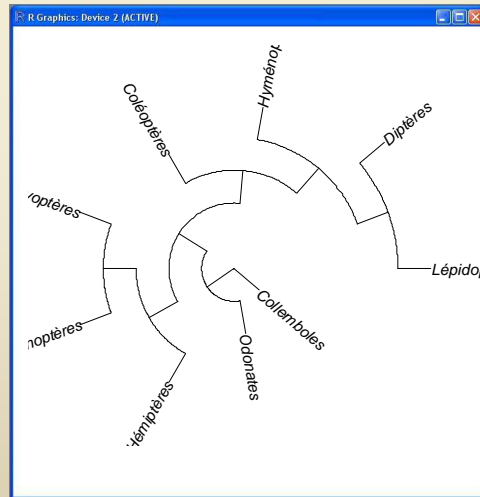
```
plot(tr,type="p",root.edge=TRUE) # default phylogramm  
plot(tr,type="c",root.edge=TRUE) # cladogramm  
plot(tr,type="f",root.edge=TRUE) # fan  
plot(tr,type="u",root.edge=TRUE) # unrooted  
plot(tr,type="r",root.edge=TRUE) # radial
```



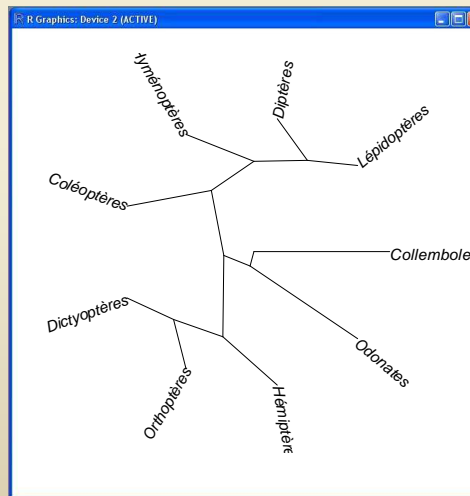
default phylogramm



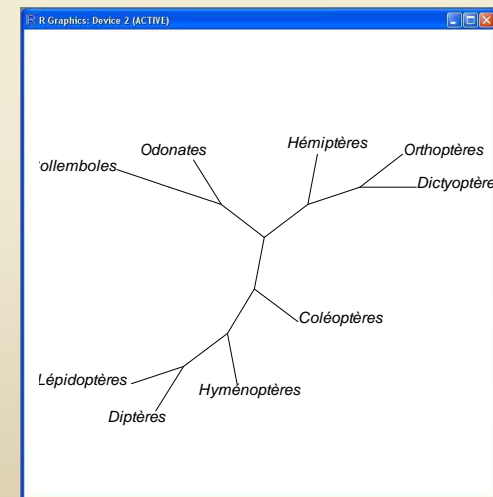
cladogramm



fan



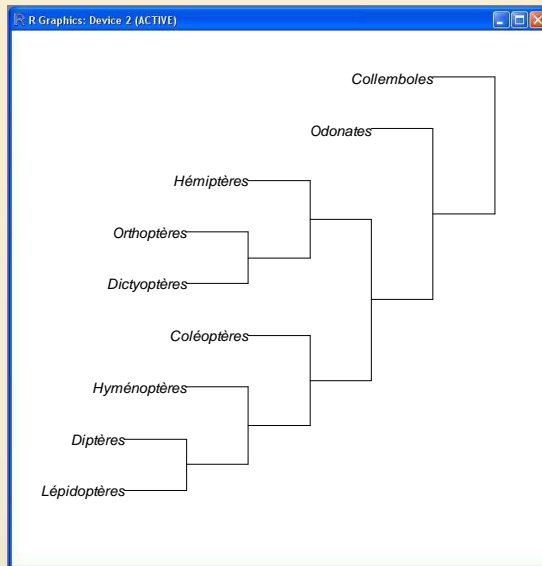
unrooted



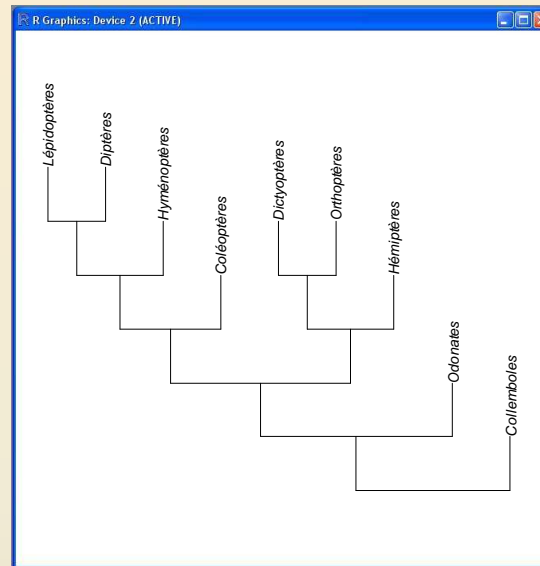
radial

Faire bouger un arbre !

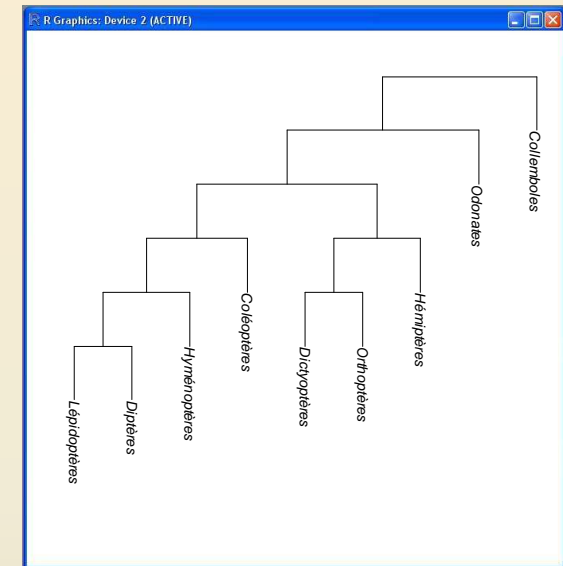
tree direction: plot(tr,direction="x")



plot(tr,direction="l") # left



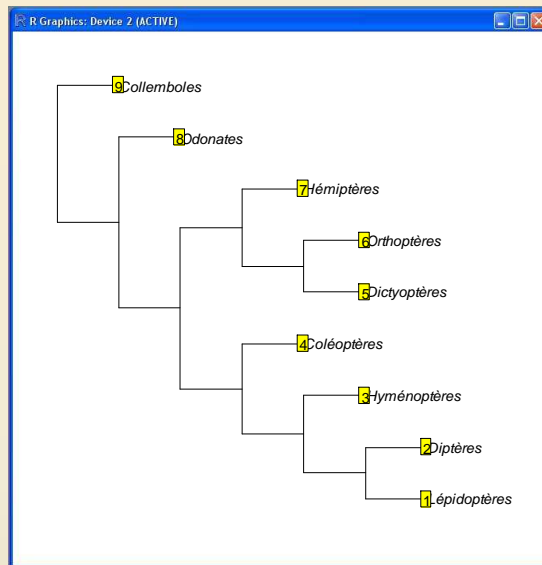
plot(tr,direction="u") # up



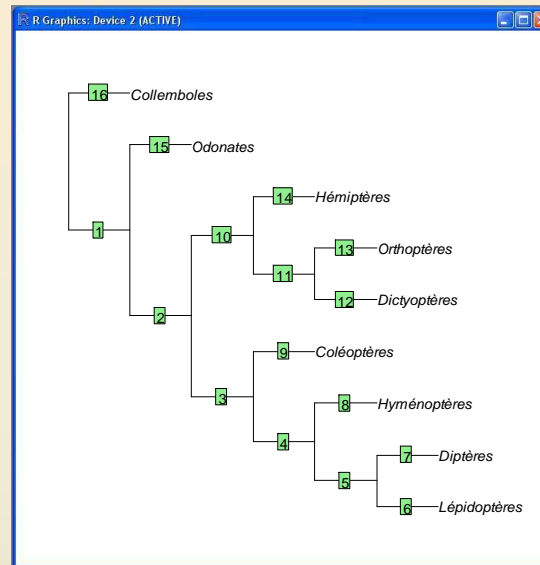
plot(tr,direction="d") # down

Faire bouger un arbre !

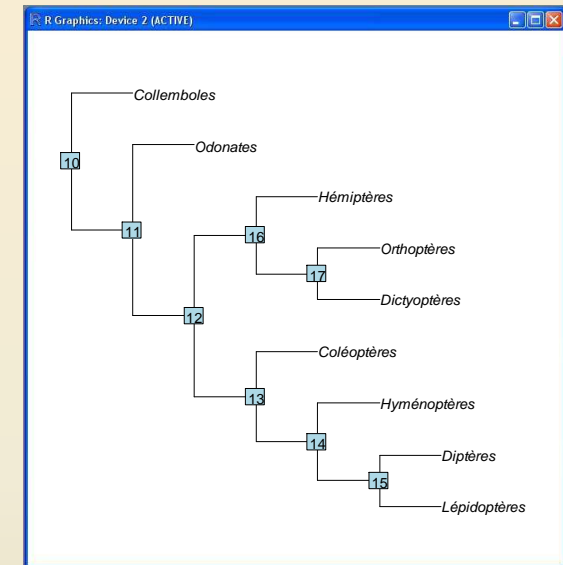
tips, nodes and edges labels: where are they?



tiplabels()



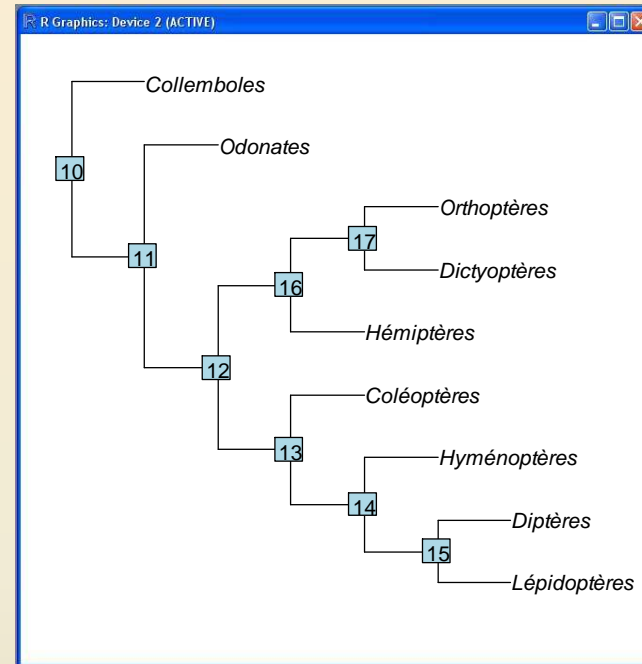
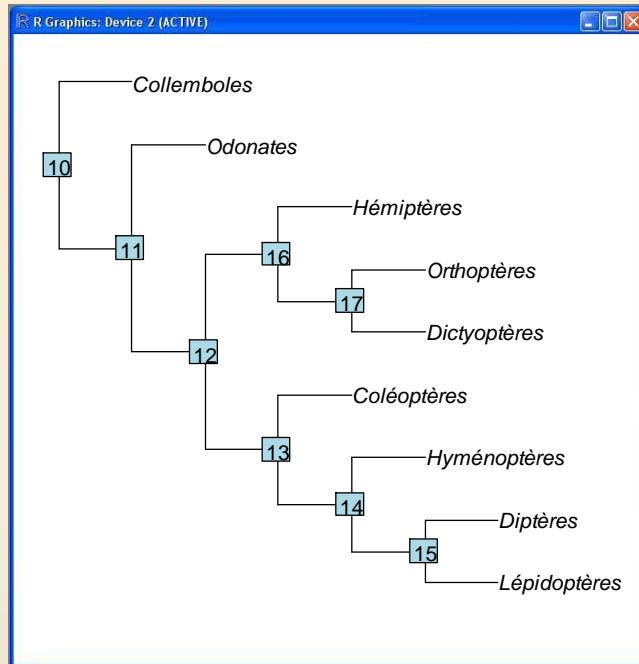
edgelabels()



nodelabels()

Faire bouger un arbre !

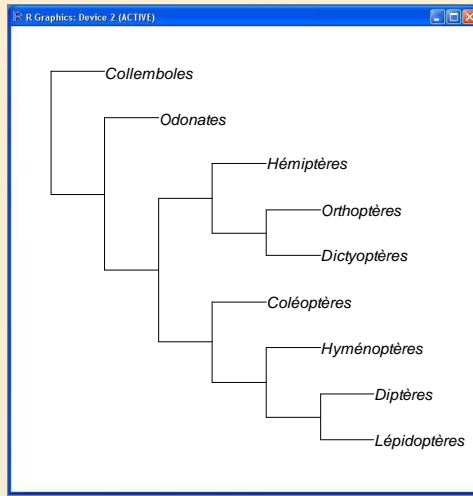
rotation around a node



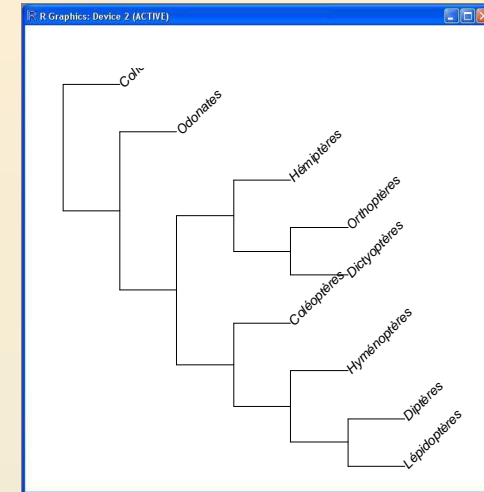
plot(rotate(tr,node=16))

Faire bouger un arbre !

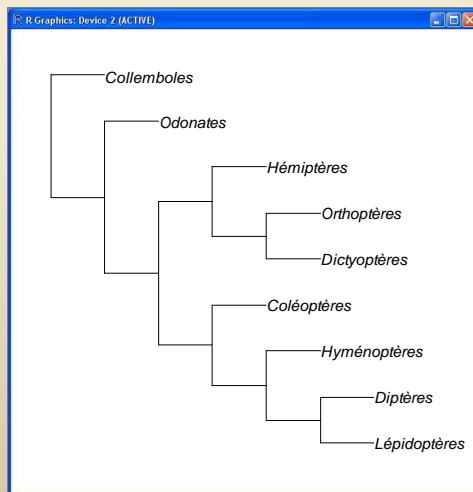
rotate tip labels: `tr.srt=angle`



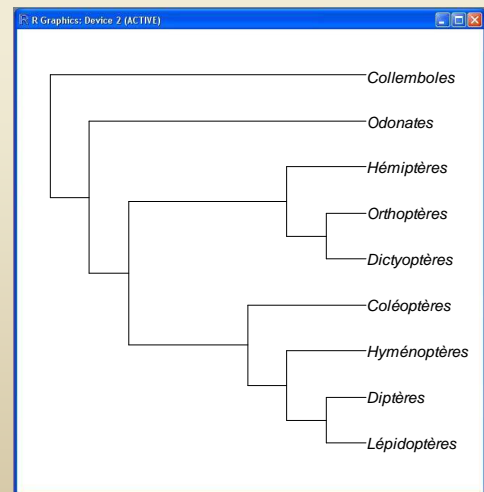
`plot (tr,srt=45)`



aligner les taxons: `use.edge.length=FALSE`

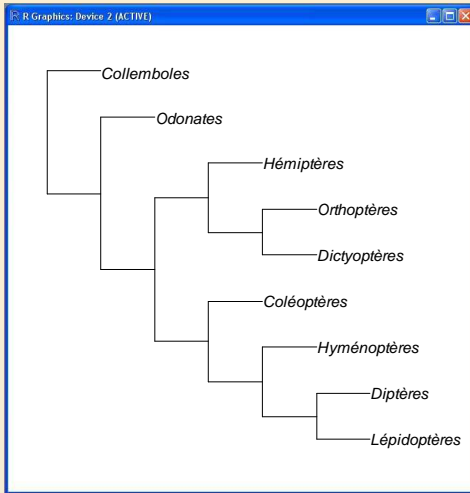


`plot (tr,use.edge.length=FALSE)`

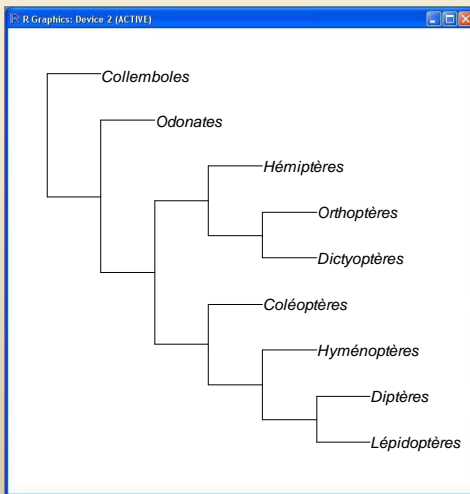
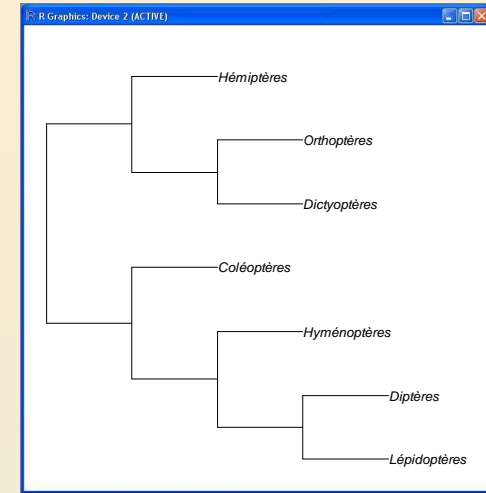


Faire bouger un arbre !

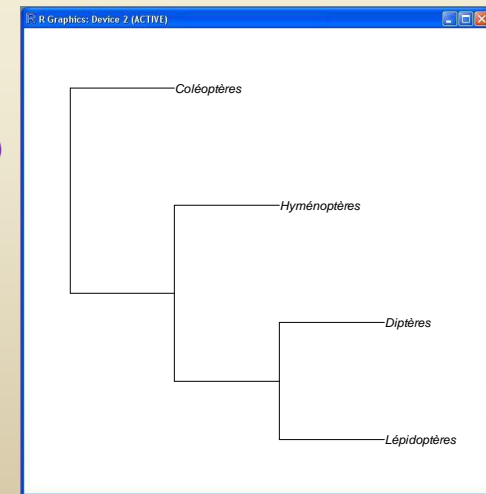
#selectionner un subsample of a tree (drop a tip)



tip <- c("Collemboles", "Odonates")
plot(drop.tip(tr, tip))



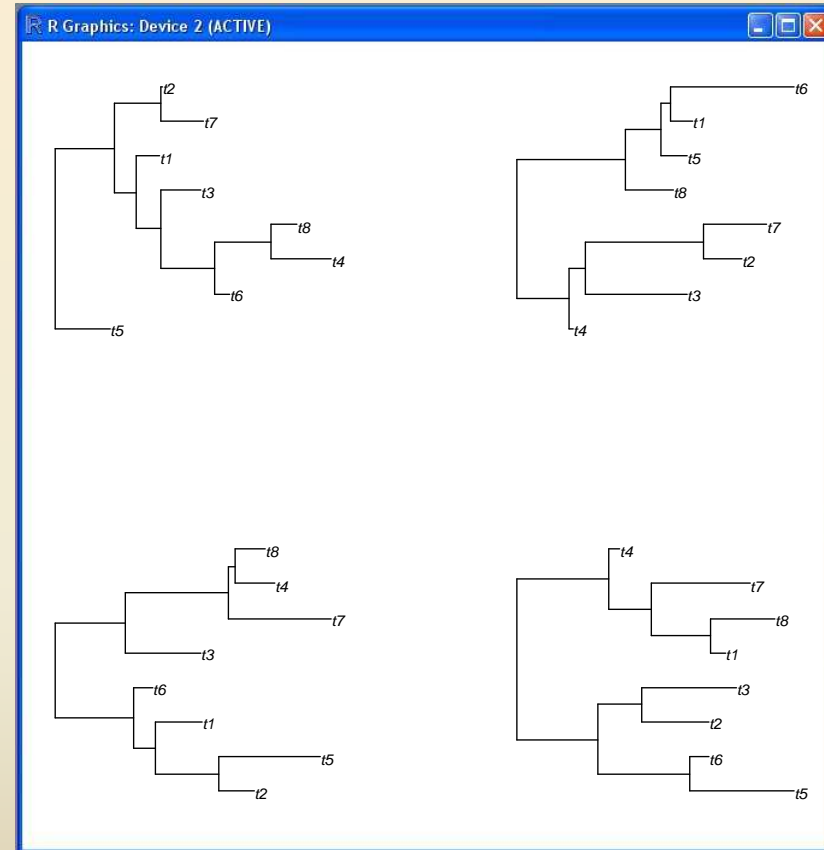
tip <- c("Collemboles",
"Odonates", "Hémiptères", "Orthoptères", "Dictyoptères")
plot(drop.tip(tr, tip))



Il n'y a pas qu'un seul arbre !

plot as many tree as you wish

```
op <- par(mfrow=c(2,2))  
plot(rtree(8))  
plot(rtree(8))  
plot(rtree(8))  
plot(rtree(8))  
par(op)
```

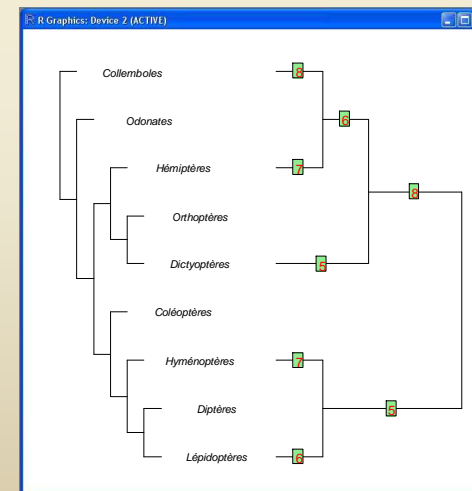
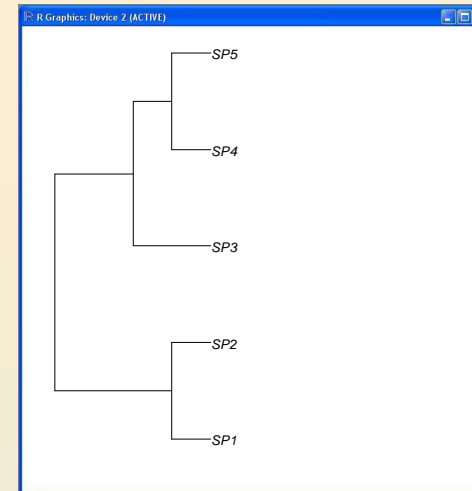


Il n'y a pas qu'un seul arbre !

facing trees: do it directly with cophyloplot()

```
t2<-"((SP1,SP2),(SP3,(SP4,SP5))));"  
t2<-read.tree(text=t2)
```

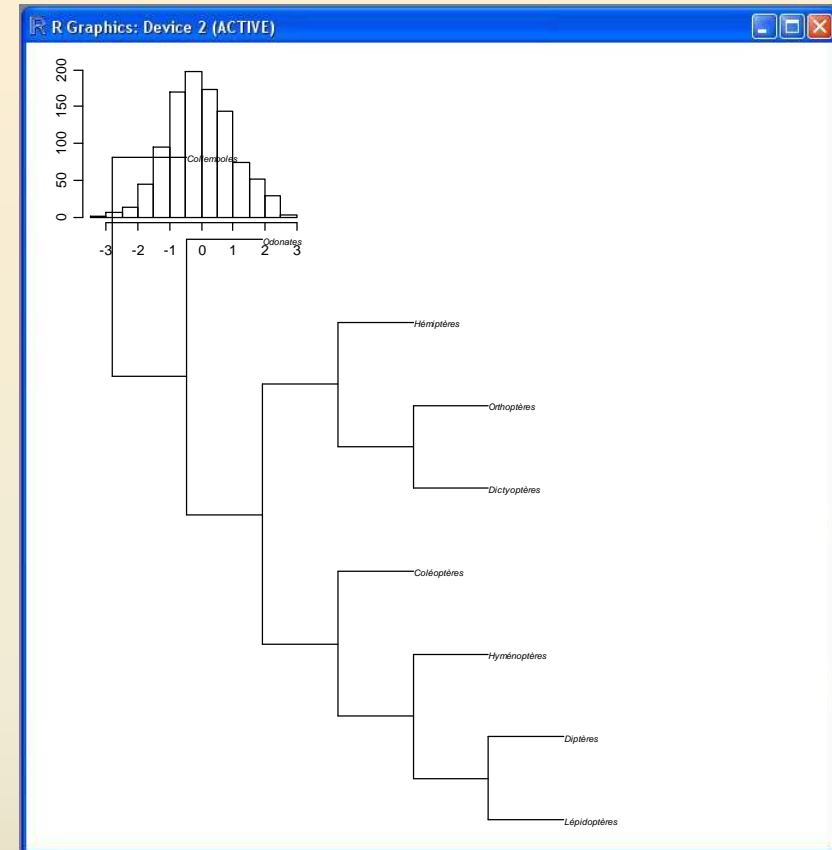
```
layout(matrix(1:2,1,2),width=c(1,1))  
par(mar=c(5,4,4,1))  
plot(tr,adj=0.5,cex=0.8,  
x.lim=c(0,11),label.offset=1.1)  
par(mar=c(5,0,4,5))  
plot(t2,adj=0.5,cex=0.8,x.lim=c(1,5),  
direction="l",show.tip.label=FALSE)  
edgelabels(5:8,col="red")
```



Il n'y a pas qu'un seul arbre !

tree and external graph

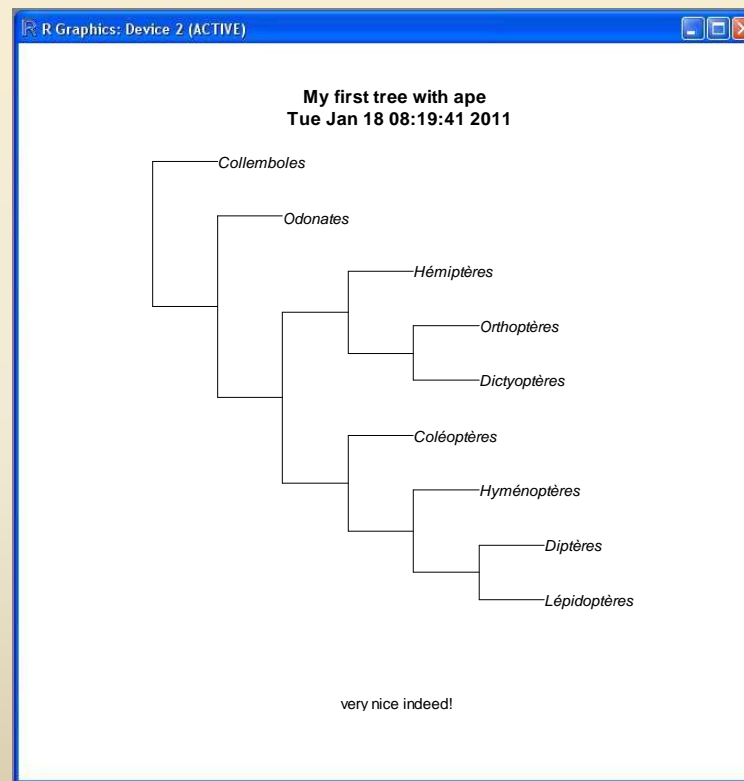
```
m <- matrix(c(2, rep(1, 8)), 3, 3)
layout(m)
par(mar=c(4, 3, 5, 1))
plot(tr)
par(mar=c(7, 3, 1, 1))
hist(rnorm(1000), xlab="", ylab="", main="")
```



Rendre beau un arbre (l'habiller, le maquiller...)

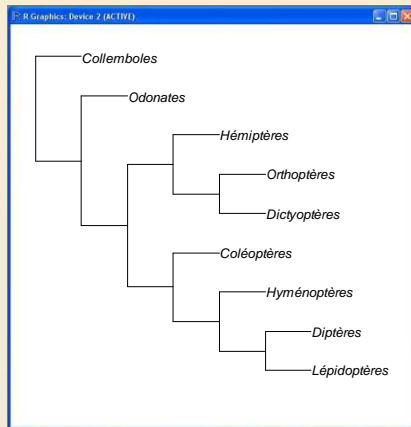
add a title

```
title(main=paste("My first tree with ape \n", date()),sub="very nice indeed!")
```



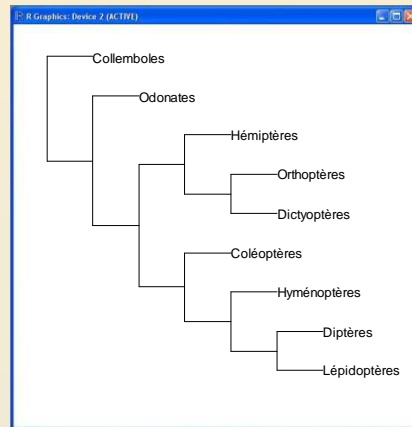
Rendre beau un arbre (l'habiller, le maquiller...)

change font of tip labels



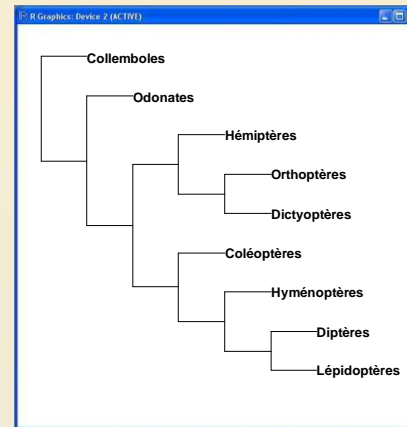
plot(tr,font=3)

It (default)



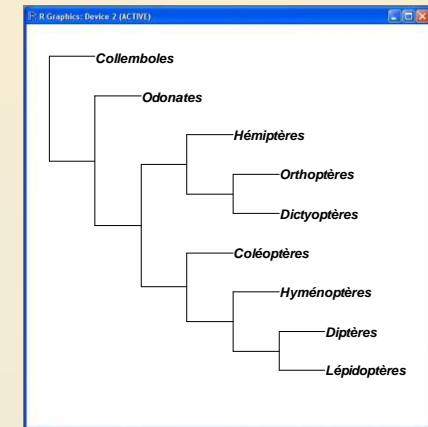
plot(tr,font=1)

normal



plot(tr,font=2)

gras



plot(tr,font=4)

gras & it

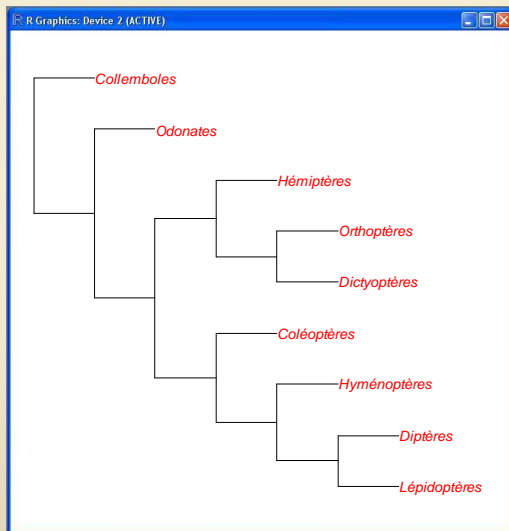
+ nombreuses polices disponibles...

Rendre beau un arbre (l'habiller, le maquiller...)

change color of tip labels

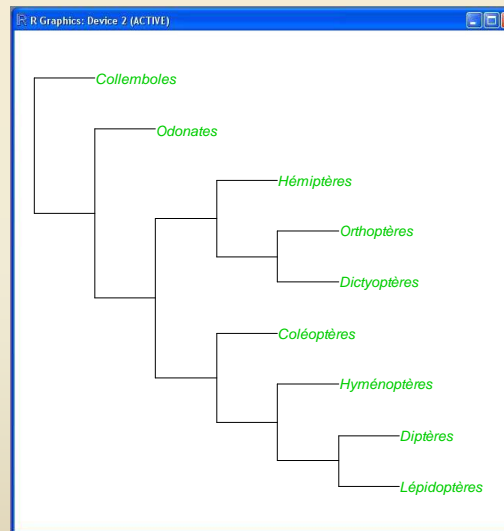
`colors()`

```
RGui - [R Console]
Fichier Edition Voir Misc Packages Fenêtres Aide
> colors()
[1] "white"           "aliceblue"
[3] "antiquewhite"   "antiquewhite1"
[5] "antiquewhite2"  "antiquewhite3"
[7] "antiquewhite4"  "aquamarine"
[9] "aquamarine1"    "aquamarine2"
[11] "aquamarine3"    "aquamarine4"
[13] "azure"          "azure1"
[15] "azure2"         "azure3"
[17] "azure4"         "beige"
[19] "bisque"         "bisque1"
[21] "bisque2"        "bisque3"
[23] "bisque4"        "black"
[25] "blanchedalmond" "blue"
[27] "blue"           "blue1"
[29] "blue2"          "blue3"
[31] "blueviolet"     "brown"
[33] "brown"          "brown1"
[35] "brown2"         "brown3"
[37] "burlywood"      "burlywood1"
[39] "burlywood2"    "burlywood3"
[41] "burlywood4"    "cadetblue"
[43] "cadetblue1"    "cadetblue2"
[45] "cadetblue3"    "cadetblue4"
[47] "chartreuse"     "chartreuse1"
[49] "chartreuse2"   "chartreuse3"
[51] "chartreuse4"   "chocolate"
[53] "chocolate1"   "chocolate2"
[55] "chocolate3"   "chocolate4"
[57] "coral"         "coral1"
[59] "coral2"        "coral3"
[61] "coral4"        "cornflowerblue"
[63] "cornsilk"      "cornsilk1"
[65] "cornsilk2"     "cornsilk3"
[67] "cornsilk4"     "cyan"
[69] "cyan"          "cyan1"
```



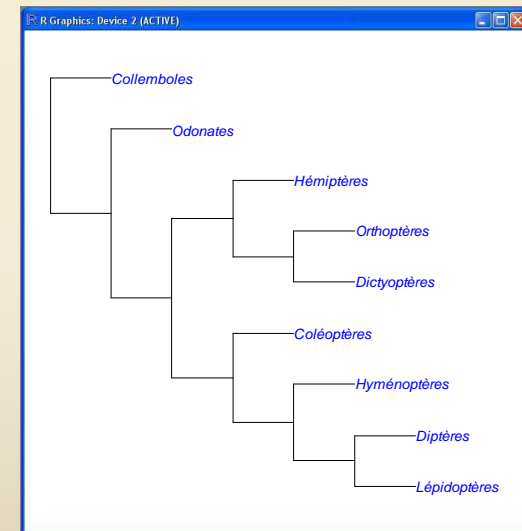
`plot(tr,tip.color=2)`

rouge



`plot(tr,tip.color=3)`

vert

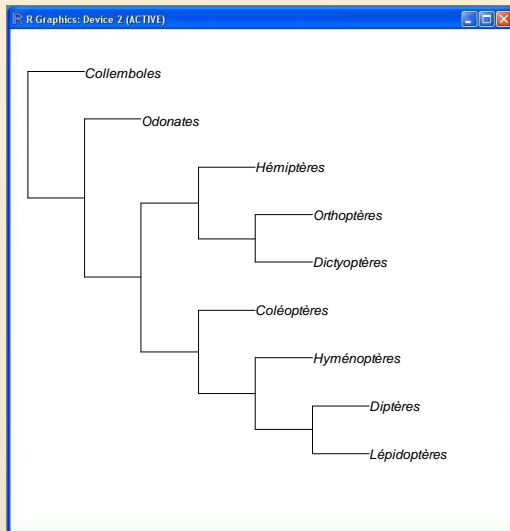


`plot(tr,tip.color=4)`

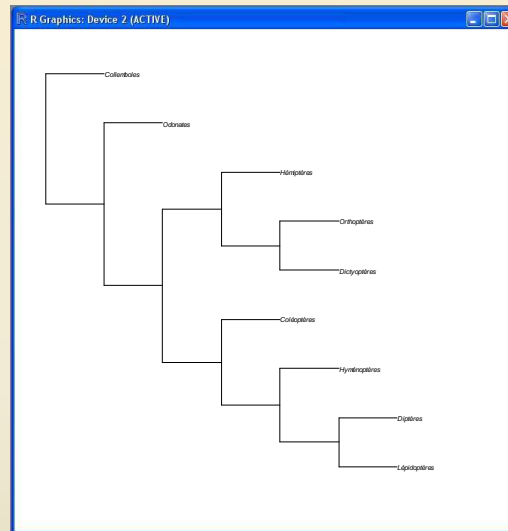
bleu

Rendre beau un arbre (l'habiller, le maquiller...)

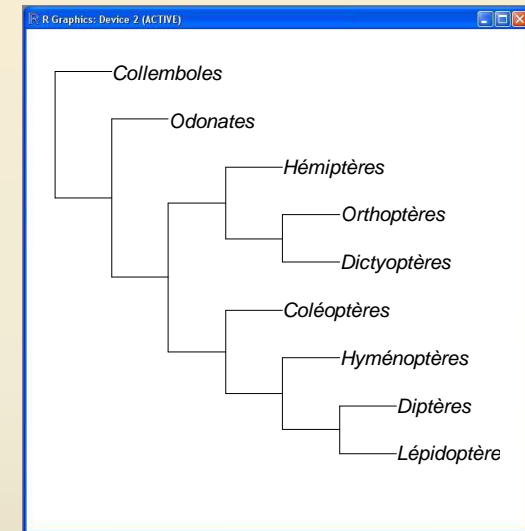
change size of tip labels



plot(tr,cex=1) (default)



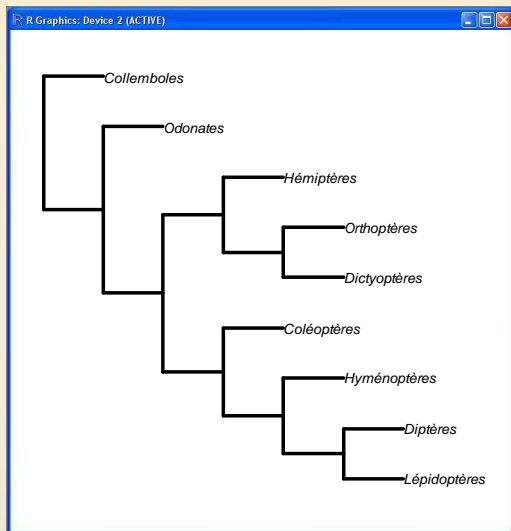
plot(tr,cex=0.5)



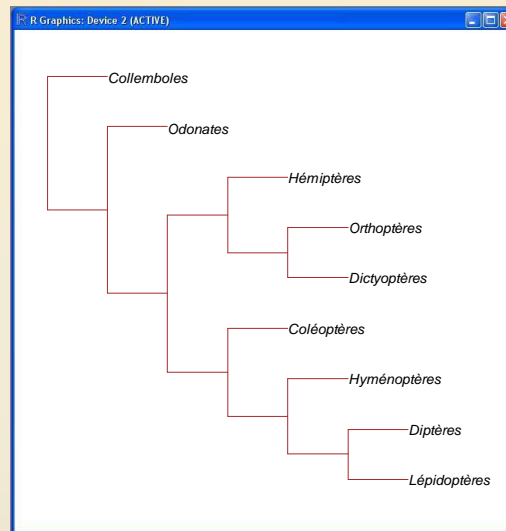
plot(tr,cex=1.5)

Rendre beau un arbre (l'habiller, le maquiller...)

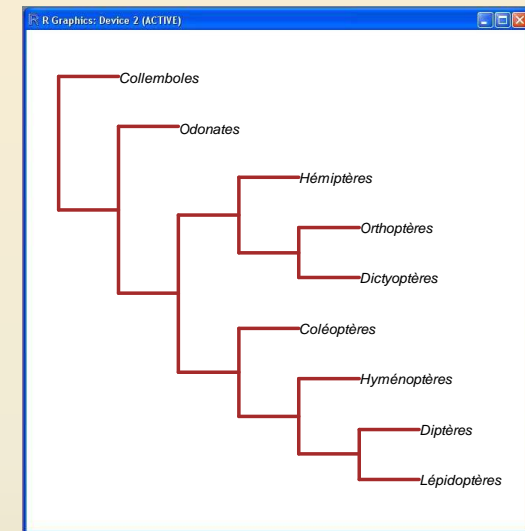
change width and color of edges



`plot(tr,edge.width=4)`



`plot(tr,edge.color="brown")`



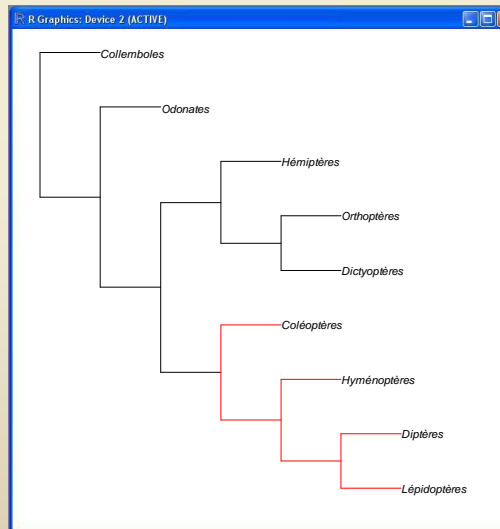
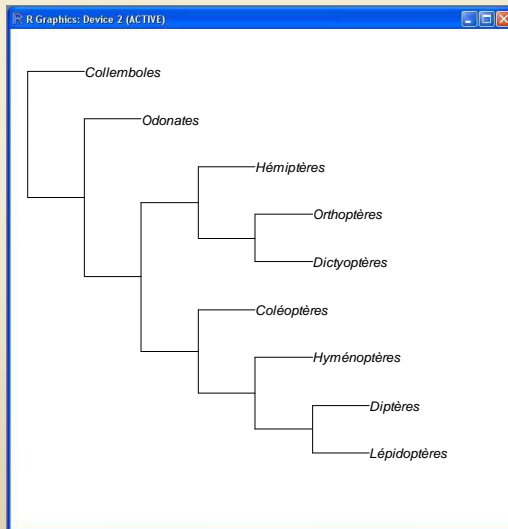
`plot(tr,edge.width=4,edge.color="brown")`

Rendre beau un arbre (l'habiller, le maquiller...)

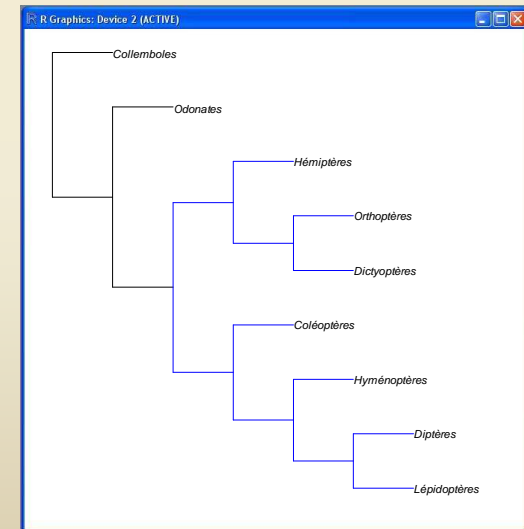
Quand davantage de taxons....

```
Holométaboles=node.leaves(tr, mrca(tr)["Coléoptères", "Lépidoptères"])  
Néoptères=node.leaves(tr, mrca(tr)["Hémiptères", "Lépidoptères"])  
color=rep("black", dim(tr$edge)[1])  
c=which.edge(tr, Holométaboles)  
color[c]="red"  
c=which.edge(tr, Néoptères)  
color[c]="blue"  
plot(tr, edge.col=color, no.margin=T, lwd=2)
```

N.B: La fonction `node.leaves()` appartient au package `geiger`.



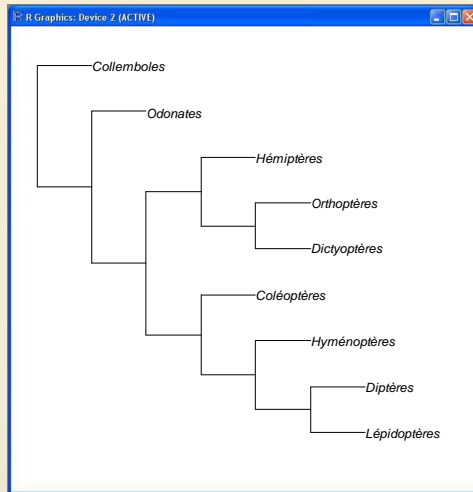
Holométaboles



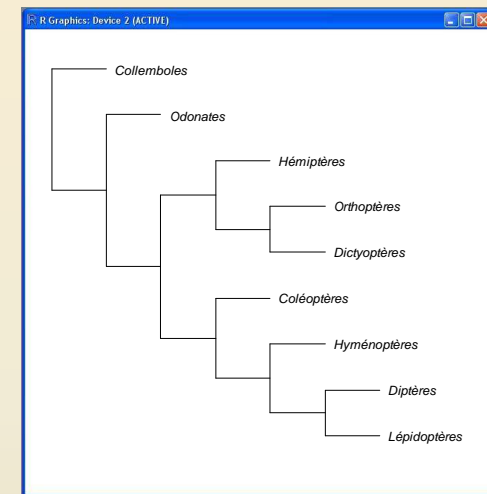
Néoptères

Rendre beau un arbre (l'habiller, le maquiller...)

add a blank between tip labels and terminal edges



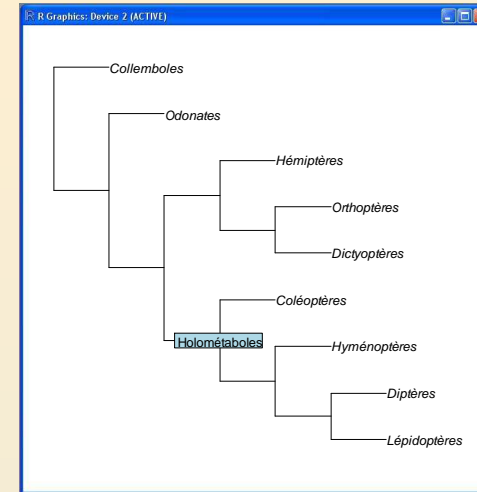
plot(tr,label.offset=0.15)



Rendre beau un arbre (l'habiller, le maquiller...)

add node labels

```
nodelabels("Holométaboles", node=13)
```



bootstrap labels

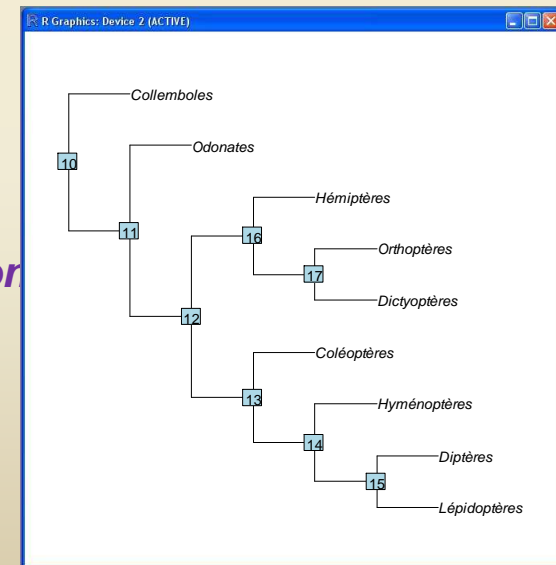
```
plot(tr)
```

```
nodelabels()
```

```
bs.pars <- scan()
```

```
plot(tr)
```

```
nodelabels(bs.pars, adj = c(-0.2, -0.1), frame = "n", cex = 0.8, for
```



Exemple

Un exemple concret

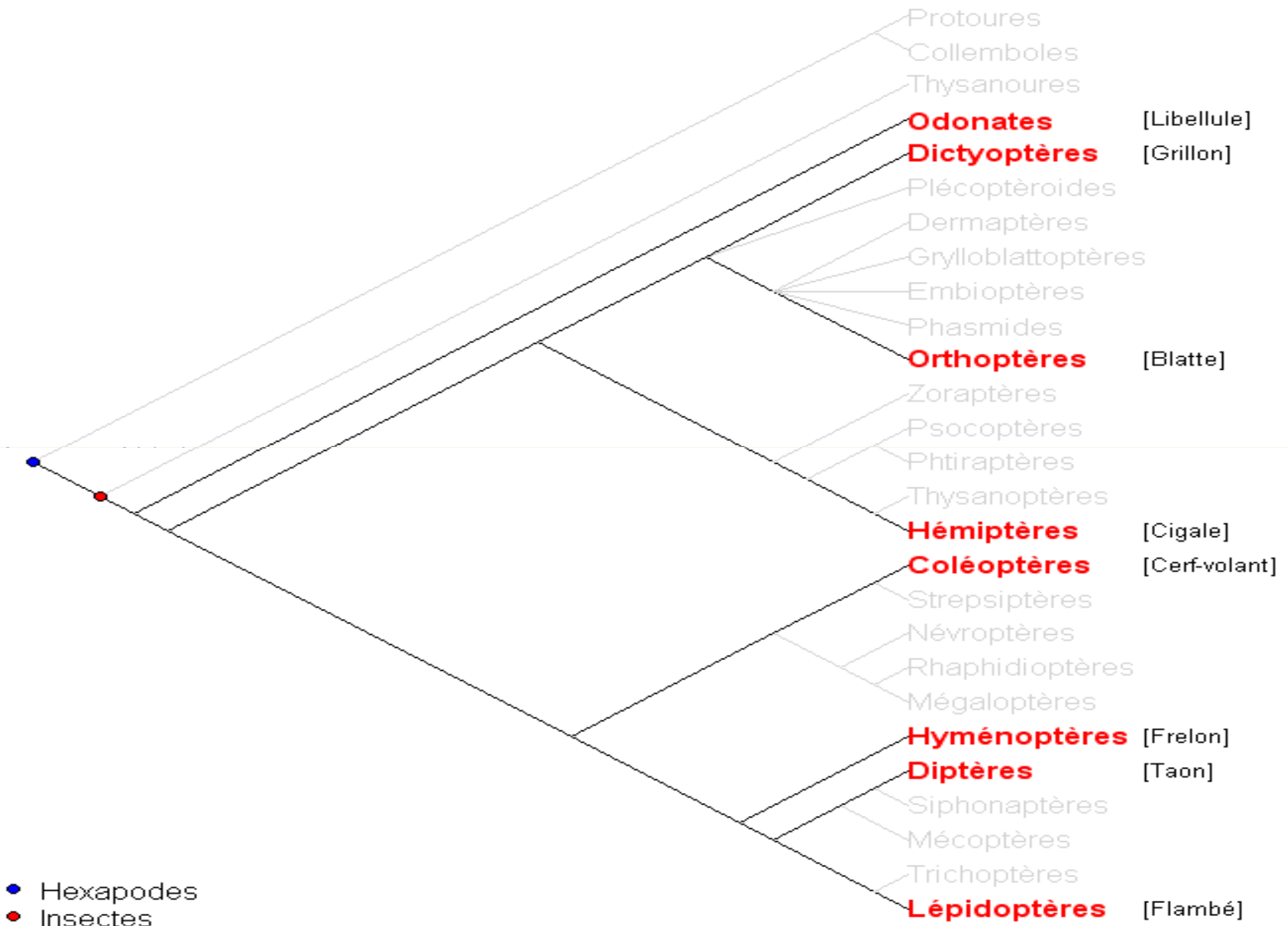
```
# a nice example

# HEXAPODES de Lecointre & Leguyader (2001)
# entrée de l'arbre sous format Newick
# division en plusieurs clades pour plus de lisibilité
clade1 <- "((((Lépidoptères, Trichoptères),(Mécoptères,(Siphonaptères,Diptères))), Hyménoptères),(((Mégaloptères, Raphidioptères), Névroptères),(Strepsiptères, Coléoptères)))"
clade2 <- "((((Hémiptères, Thysanoptères),(Phtiraptères, Psocoptères)), Zoraptères), ((Orthoptères, Phasmides, Embioptères, Grylloblattoptères, Dermaptères), Plécoptéroïdes, Dictyoptères))"
clade3 <- paste("(", clade1, ", ", clade2, ")")
hexapods <- paste("(", clade3, ", Odonates), Thysanoures),(Collemboles, Protoures));")
hexapods <- read.tree(text=hexapods)

# gestion des couleurs
# couleurs des branches
edge.color <- rep(1, 49)
edge.color[c(9,11,13,17,18,19,20,21,23,30,31,32,33,34,38,39,40,41,42,45,46,47,48)]<-"lightgrey"
# font et couleur des étiquettes
font <- tip.color <- rep(2,27)
sel tip <- c(2,3,4,7,8,9,10,13,14:16,18:22,25:27)
font[sel tip]<-1
tip.color[sel tip]<-"lightgrey"
# récupérer les limites de x
x.lim <- plot(hexapods, type="c")$x.lim
dev.off()
# plot l'arbre
plot(hexapods,edge.color=edge.color,tip.color=tip.color, type="c", no.margin=TRUE,font=font, x.lim=c(x.lim[1],x.lim[2]+2))
# texte supplémentaire
text(c("[Flambé]", "[Taon]", "[Frelon]", "[Cerf-volant]", "[Cigale]", "[Blatte]", "[Grillon]", "[Libellule]"),
     x=rep(33,8), y=c(1,5,6,11,12,17,23,24), adj=0, cex=0.85)
# noeuds
nodelabels(n=28, frame="n", pch=21, bg="blue")
nodelabels(n=29, frame="n", pch=21, bg="red")
# légendes
legend("bottomleft", c("Hexapodes", "Insectes"), pt.bg=c("blue","red"), pch=21, bty="n")
```


Un exemple concret

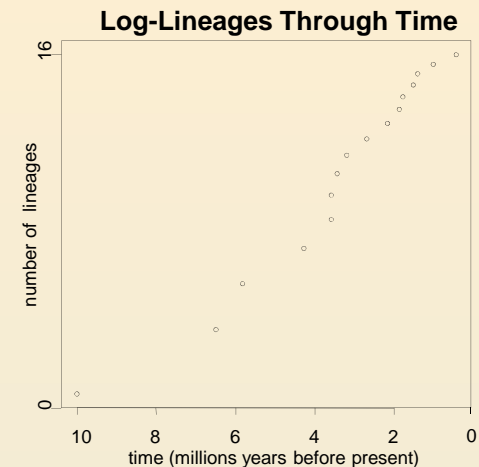
```
# a nic
# HEX
# entre
# divis
clade1
Névro
clade2
Gryllo
clade3
hexap
hexap
# gest
# coul
edge.c
edge.c
# font
font <
sel tip
font[s
tip.col
# recu
x.lim <
dev.of
# plot
plot(h
# texte
text(c(
x=r
# noe
node1
node1
# lége
legenc
```



Pour aller plus loin: faire des analyses phylogénétiques sous R ?

Table 1. Special functions available in APE 1.1

Application	Available commands
Input/output	<code>read.dna</code> , <code>write.dna</code> , <code>read.nexus</code> , <code>write.nexus</code> , <code>read.tree</code> , <code>write.tree</code> , <code>read.GenBank</code>
Graphics	<code>add.scale.bar</code> , <code>plot.mst</code> , <code>plot.phylo</code> , <code>plot.skyline</code> , <code>lines.skyline</code> , <code>lft.plot</code>
Tree manipulation	<code>bind.tree</code> , <code>drop.tip</code> , <code>is.binary.tree</code> , <code>is.ultrametric</code>
Comparative method	<code>compar.gee</code> , <code>compar.lynch.pic</code> , <code>vcv.phylo</code>
Diversification	<code>birthdeath</code> , <code>cherry</code> , <code>diversi.gof</code> , <code>diversi.time</code> , <code>gamma.stat</code>
Population genetics	<code>branching.times</code> , <code>coalescent.intervals</code> , <code>collapsed.intervals</code> , <code>find.skyline.epsilon</code> , <code>heterozygosity</code> , <code>skylineplot</code> , <code>skyline</code> , <code>theta.h</code> , <code>theta.k</code> , <code>theta.s</code>
Molecular dating	<code>chronogram</code> , <code>ratogram</code> , <code>NPRS.criterion</code>
Miscellaneous	<code>all.equal.phylo</code> , <code>balance</code> , <code>base.freq</code> , <code>dist.dna</code> , <code>dist.gene</code> , <code>dist.phylo</code> , <code>GC.content</code> , <code>klatorin</code> , <code>mantel.test</code> , <code>mst</code> , <code>summary.phylo</code>
Data sets	<code>bird.families</code> , <code>bird.orders</code> , <code>hivtree</code> , <code>landplants</code> , <code>opsin</code> , <code>woodmouse</code> , <code>xenarthra</code>



LTT plot

Autres packages: LASER, GEIGER...



Remerciements:

- MNHN, Semin-R
- Jérôme Sueur (UMR 7205)
- Loïc Ponger (USM 503)
- Votre attention.... et vos questions !