Branch length evaluation for Phylogenetic Diversity: a worked example

Daniel R. Miranda-Esquivel

2018 - 07 - 02

Four taxa and two areas

Preparing the data space

First, we load the required libraries:

```
## cleaning
rm(list = ls())

## libraries

## installing and loading the package

##install.packages("../../blepd_0.1.1.tar.gz", repos = NULL, type="source")

library(blepd)
```

```
## Loading required package: ape

## Loading required package: picante

## Loading required package: vegan

## Loading required package: permute

##
## Attaching package: 'permute'

## The following object is masked from 'package:devtools':
##
##     check

## Loading required package: lattice

## This is vegan 2.5-2

## Loading required package: nlme

## Loading required package: methods
```

```
packageVersion("blepd")
```

```
## [1] '0.1.4.2018.7.5.1625'
```

```
## To plot trees you can use ggtree, ape or phytools. The example is based on
## ggtree as a matter of choice.

library(ggtree)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: treeio

##
## Attaching package: 'treeio'

## The following objects are masked from 'package:ape':
##
##     drop.tip, Nnode, Ntip

## ggtree v1.10.2  For help: https://guangchuangyu.github.io/ggtree
##
## If you use ggtree in published research, please cite:
## Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggtree: an R package for visua

##
## Attaching package: 'ggtree'

## The following object is masked from 'package:nlme':
##
##     collapse

## The following object is masked from 'package:ape':
##
##     rotate
```

```
library(gridExtra)

library(RColorBrewer)
```

Now, we load the data included in the package: tree and distribution.

```
#data(package = "blepd")

## trees

data(tree)

str(tree)
```

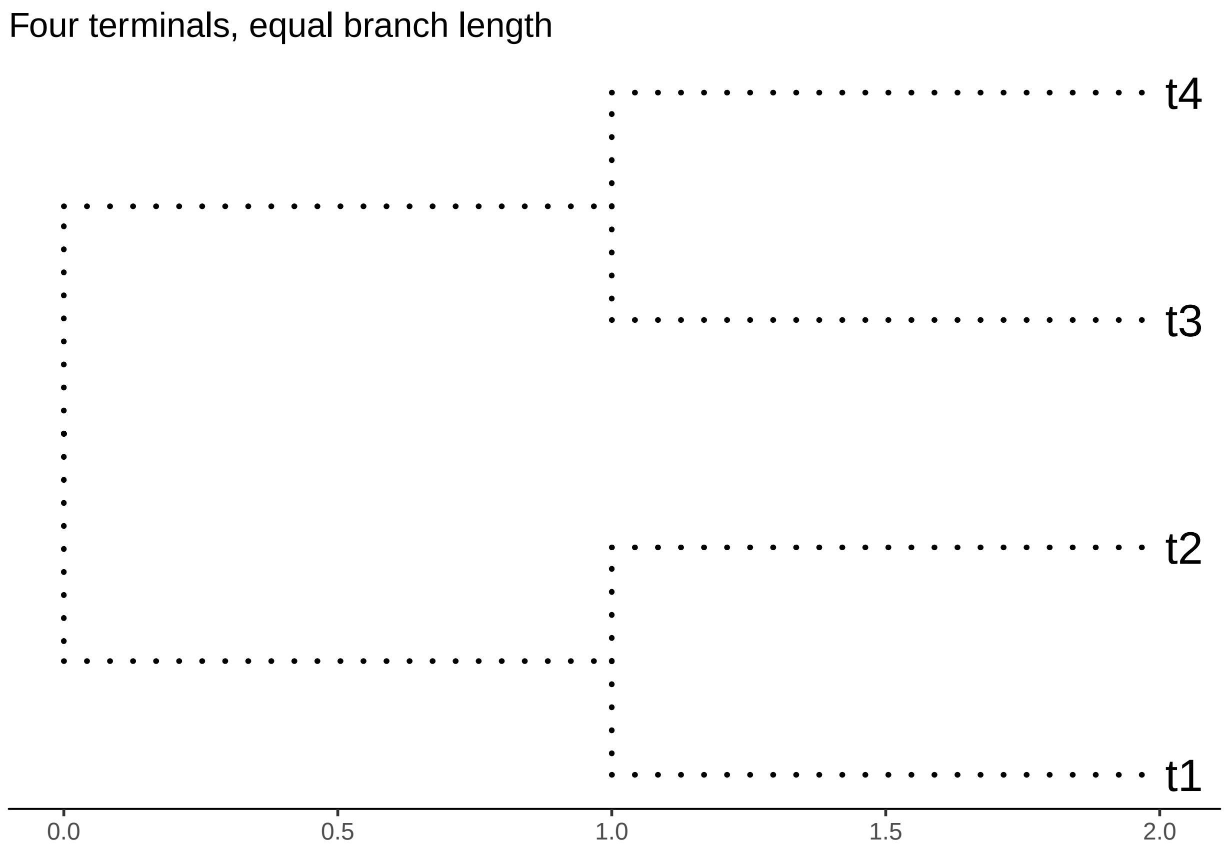
```
## List of 5
##  $ edge       : int [1:6, 1:2] 5 6 6 5 7 7 6 1 2 7 ...
##  $ edge.length: num [1:6] 1 1 1 1 1 1
##  $ Nnode      : int 3
##  $ tip.label  : chr [1:4] "t1" "t2" "t3" "t4"
##  $ root.edge  : num 1
##  - attr(*, "class")= chr "phylo"
##  - attr(*, "order")= chr "cladewise"
```

```
initialTree <- tree


## distributions

data(distribution)

str(distribution)
```

```
##  int [1:2, 1:4] 1 0 0 1 1 0 0 1
##  - attr(*, "dimnames")=List of 2
```

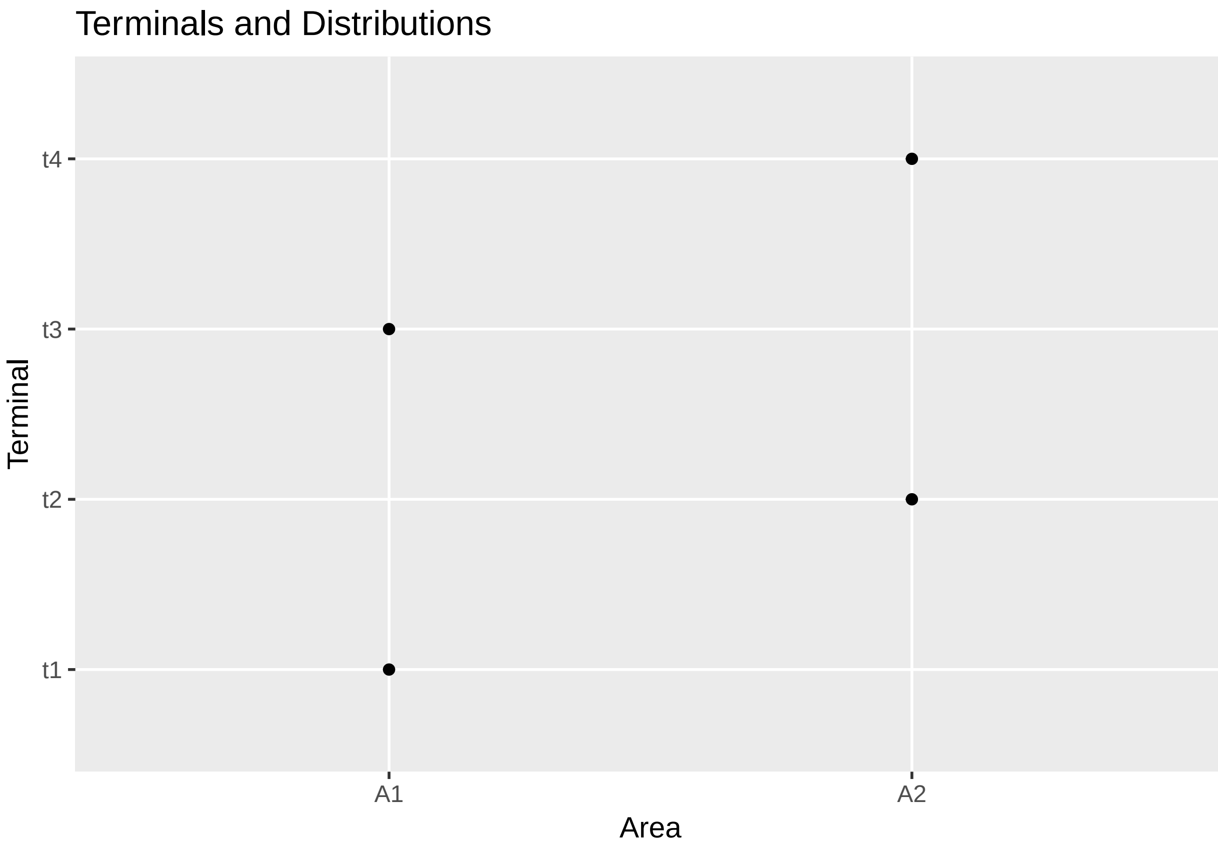
```
##   ..$ : chr [1:2] "A1" "A2"
##   ..$ : chr [1:4] "t1" "t2" "t3" "t4"
```

```
dist4taxa <- distribution

## distribution to XY

distXY <- matrix2XY(dist4taxa)


## plotting

## the tree

plotTree <-  ggtree(initialTree, ladderize=TRUE,
                    color="black", size=1, linetype="dotted") +
             geom_tiplab(size=6, color="black") +
             theme_tree2() +
             labs(title = "Four terminals, equal branch length")


print(plotTree)
```




```
## the distribution

plotDistrib <- ggplot(data=distXY,
                      aes(x= Area, y= Terminal),
                      size =11) +
```

```
                geom_point() +
                labs(title = "Terminals and Distributions",
                     y = "Terminal",
                     x = "Area")

print(plotDistrib)
```



We check whether names in both objects: initialTree and dist4taxa are the same.

```
all(colnames(dist4taxa) == initialTree$tip.label)
```

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

We report the branch length, and calculate the PD values.

```
initialTree$edge.length
```

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

```
initialPD <- PDindex(tree=initialTree, distribution = dist4taxa)

initialPD
```

```
## [1] 4 4
```

Function to evaluate a single terminal

To test the effect of changing the branch length in a single terminal ("t1"), we will use the function *evalTerminal*. This function uses four parameters: tree, distribution, tipToEval (label of the tip), approach (two options: "lower"/"upper", to evaluate from 0 to the actual length or from the actual length to the sum of all branch lengths).

```
evalTerminal(tree = initialTree,
             distribution = dist4taxa,
             tipToEval = "t1",
             approach = "lower" )
```

```
## branchLengthChange    bestInitialArea   bestModifiedArea
##           "0.9999"             "A1A2"               "A2"
##      initialLength
##                "1"
```

The lower limit reported when we change the branch length for terminal t1 is 0.99, therefore any change in this branch length will modify the area selected from A1A2 to A2, as the tie between the path between terminals t1/t3 (area A1) vs t2/t4 (area A2) will be solved in favour of t2/t4 when A1 is shorter.

Tree evaluation function

branch length

The function to test all terminals at the same time is *evalTree*, with two parameters: the tree and the distribution. The function returns a data.frame object with 14 fields: labelTerminal, lowerBranchLength, InitialArea, lowerFinalArea, initialLength, upperBranchLength, upperFinalArea, changeLower, changeUpper, deltaUpper, deltaLower, deltaPD, areaDelta, and abDelta.

```
finalResults <- evalTree(tree = initialTree, distribution = dist4taxa)

finalResults
```

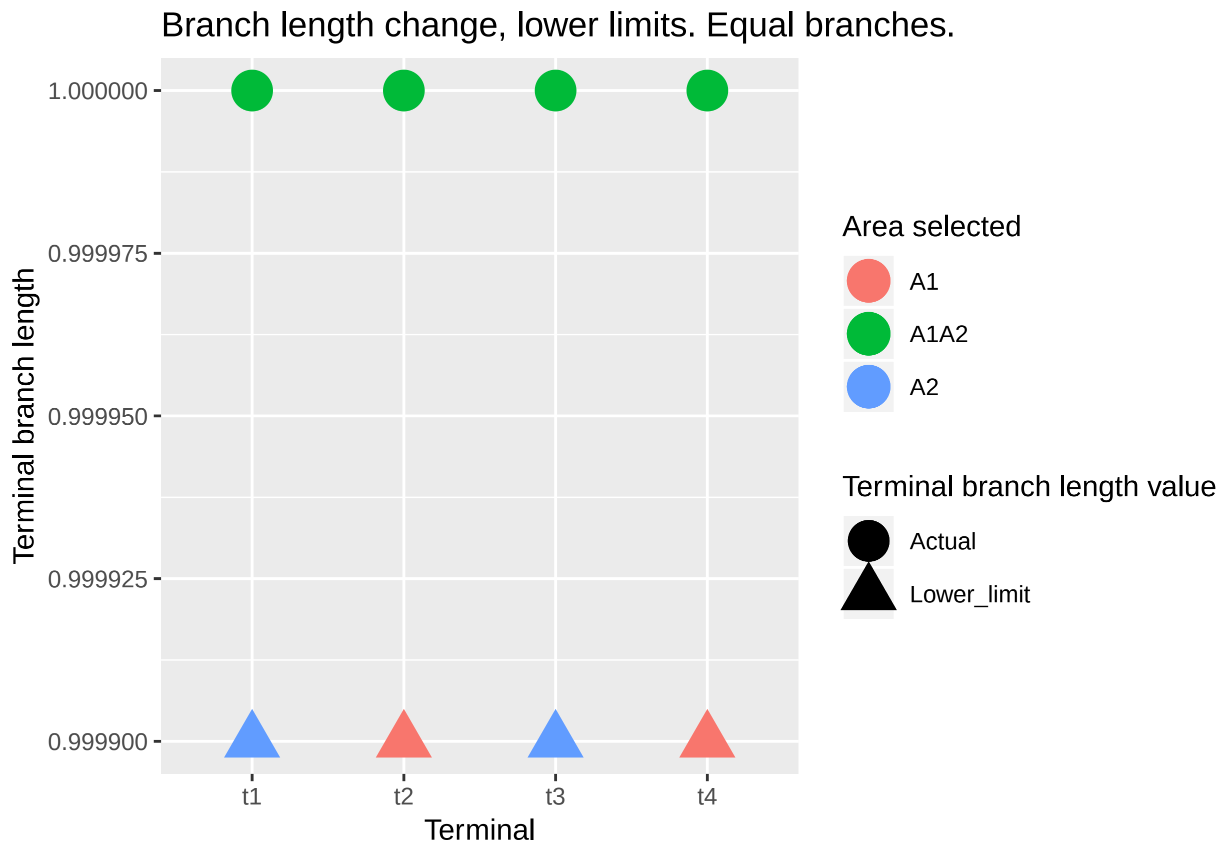
```
##   labelTerminal InitialArea initialLength lowerFinalArea lowerBranchLength
## 1            t1        A1A2             1             A2            0.9999
## 2            t2        A1A2             1             A1            0.9999
## 3            t3        A1A2             1             A2            0.9999
## 4            t4        A1A2             1             A1            0.9999
##   changeLower deltaLower upperFinalArea upperBranchLength changeUpper
## 1          A2      1e-04             A1            1.0001          A1
## 2          A1      1e-04             A2            1.0001          A2
## 3          A2      1e-04             A1            1.0001          A1
## 4          A1      1e-04             A2            1.0001          A2
##   deltaUpper deltaPD    areaDelta abDelta
## 1      1e-04       0 L:_A2_/U:_A1       0
## 2      1e-04       0 L:_A1_/U:_A2       0
## 3      1e-04       0 L:_A2_/U:_A1       0
## 4      1e-04       0 L:_A1_/U:_A2       0
```

The extreme sensitivity of the PD results to the terminal branch length is seen in the column absolute length difference (=abDelta), as any length change -larger than 0-, will modify the area selected.

We plot the results to see the effect in each terminal, as a table.

```
plotResults <- ggplot(data=finalResults, aes(x= labelTerminal, y= initialLength,
                      shape="Actual",
                      colour=InitialArea)) +
               geom_point(size= 7) +
               geom_point(aes(x= labelTerminal, y= lowerBranchLength,
                              colour=lowerFinalArea,
```

```
                          shape="Lower_limit"), size=7) +
             labs(title = "Branch length change, lower limits. Equal branches.",
                  colour = "Area selected",
                  shape = "Terminal branch length value",
                  y = "Terminal branch length",
                  x = "Terminal")

print(plotResults)
```



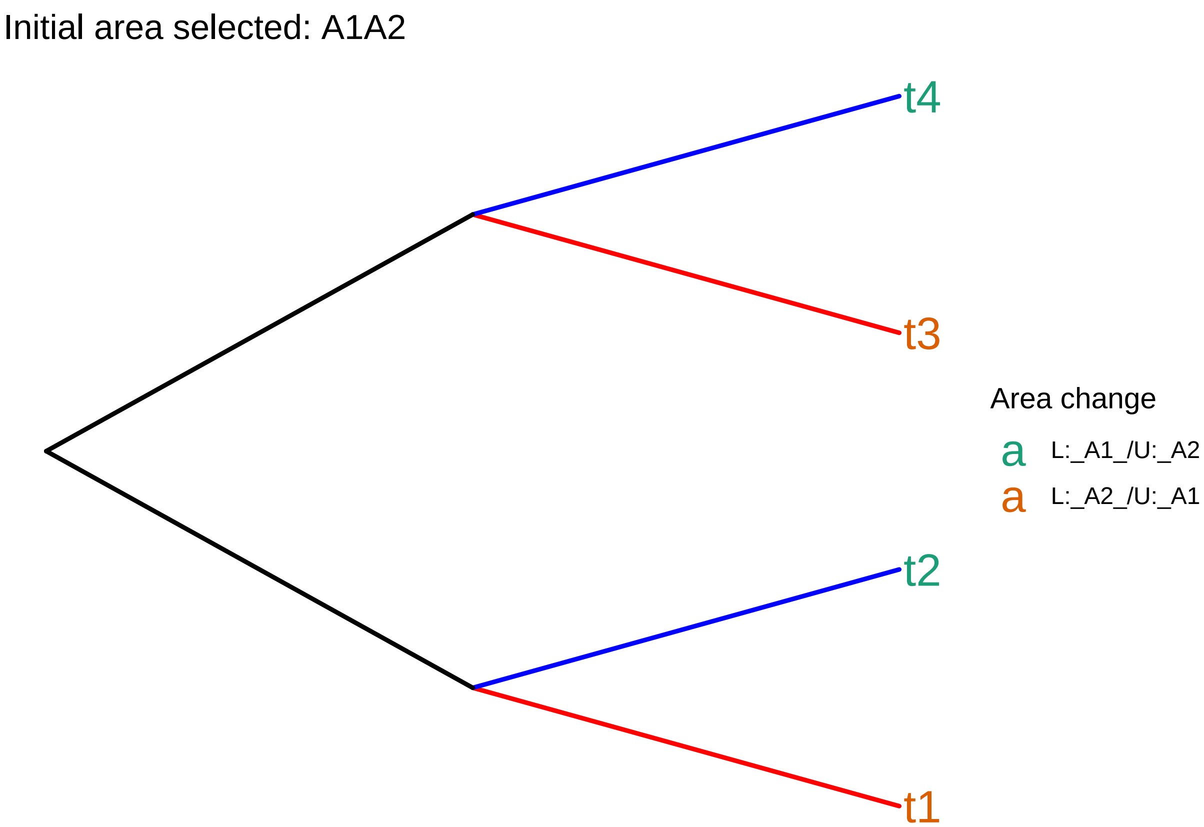
or plotted as a simple table.

```
countFreqChanges <- table(finalResults$areaDelta)


countFreqChanges <- as.data.frame(countFreqChanges, ncol=1)


colnames(countFreqChanges) <- c("Area change","Freq")


row.names(countFreqChanges) <- NULL


countFreqChanges
```

```
##   Area change Freq
## 1 L:_A1_/U:_A2    2
## 2 L:_A2_/U:_A1    2
```

or plotted into the tree:

```
theTitle <- paste("Initial area selected:",finalResults$InitialArea[1])

p0 <-    ggtree(initialTree, layout="slanted", ladderize=TRUE,
                color=c("red","blue","red","blue","black","black","black"),
                 size=0.8 ) +
         theme(legend.position="right") +
         labs(title = theTitle)


p <- p0 %<+% finalResults + geom_tiplab(aes(color=areaDelta), size =6) +
          scale_colour_brewer("Area change", palette="Dark2")


print(p)
```



For terminals t1/t3, a change from 1 to 0.99 in branch length -the lower limit (=L)- will change the initial area selected (A1A2) to A2; or a change from 1 to 1.01 in branch length -the upper limit(=U)-, will change the area to A1.

branch swap

A second, and different approach, is to evaluate the effect in PD when terminal branch lengths are swapped. In this case it is not the sensitivity to the branch length as a parameter, but the stability to the actual branch lengths.

The function to perform the analysis is *swapBL*, that has four parameters: the tree, the distribution, the model to evaluate (valid models are “simpleswap”, “allswap” -default value- and “uniform”), and the number of times to swap (default value = 100).

Uning the default parameters we get.

```
swapBL(tree = initialTree,
       distribution = dist4taxa)
```

```
## model to test allswap reps 100
```

```
##   AreaSelected Freq
## 1       A1A2 *  100
```

As this is a tree where all branches are equal, there is no impact when the branch lengths are swapped.

Or we could use the random uniform branch length model.

```
swapBL(tree = initialTree,
       distribution = dist4taxa,
       model = "uniform")
```

```
## model to test uniform reps 100
```

```
##   AreaSelected Freq
## 1       A1A2 *  100
```

This is a tree where all branches are equal, therefore min and max are equal. There is no impact when the branch lengths are swapped, and areas A1A2 are selected.

An empirical example: *Rhynoclemmys* data.

We read the data sets: distribution and trees.

```
## read distributional data

setwd("../testData/")

distribution <- as.matrix(read.table("Rhinoclemmys_Distribution",
                                  stringsAsFactors=FALSE,
                                  header=TRUE,
                                  row.names=1,
                                  sep=",")
                          )


## trees

treeFiles <- dir(pattern=".tre")

treeFiles
```

```
## [1] "Rhinoclemmys_igrClock_exp_genus_data.nexus.con.tre"
## [2] "Rhinoclemmys_igrClock_exp_uniform_data.nexus.con.tre"
## [3] "Rhinoclemmys_NonClock_genus_data.nexus.con.tre"
```

```
RhinoclemmysData <- list()

RhinoclemmysData$distribution <- distribution
```

```
RhinoclemmysData$trees  <- lapply(treeFiles,FUN=read.nexus)

RhinoclemmysData$trees  <-lapply(RhinoclemmysData$trees,
                                 FUN=root,
                                 outgroup=c("Go_polyphemus","Go_agassizii"),
                                 resolve.root = TRUE)
```

These three trees correspond to different clock models, and we want to test whether the clock used will have any effect in the areas chosen.

Fist we calculate the PD value for each tree, we save the values as a table and later we convert the table to a matrix:

```
RhinoclemmysData$tablePD <- lapply(RhinoclemmysData$trees,
                                   FUN=PDindex,
                                   distribution=RhinoclemmysData$dist,
                                   root=TRUE)

RhinoclemmysData$matrixPD <- as.data.frame(
                                  matrix(
                                      unlist(RhinoclemmysData$tablePD),
                                      nrow= length(treeFiles),
                                      byrow=TRUE))


RhinoclemmysData$tablePDPercentage <- lapply(RhinoclemmysData$trees,
                                             FUN=PDindex,
                                             distribution=RhinoclemmysData$dist,
                                             percentual=TRUE,
                                             root=TRUE)

RhinoclemmysData$matrixPDPercentage <- as.data.frame(
                                            matrix(
                                                unlist(RhinoclemmysData$tablePDPercentage),
                                                nrow= length(treeFiles),
                                                byrow=TRUE))
```

now, it is time to name trees, terminals and areas:

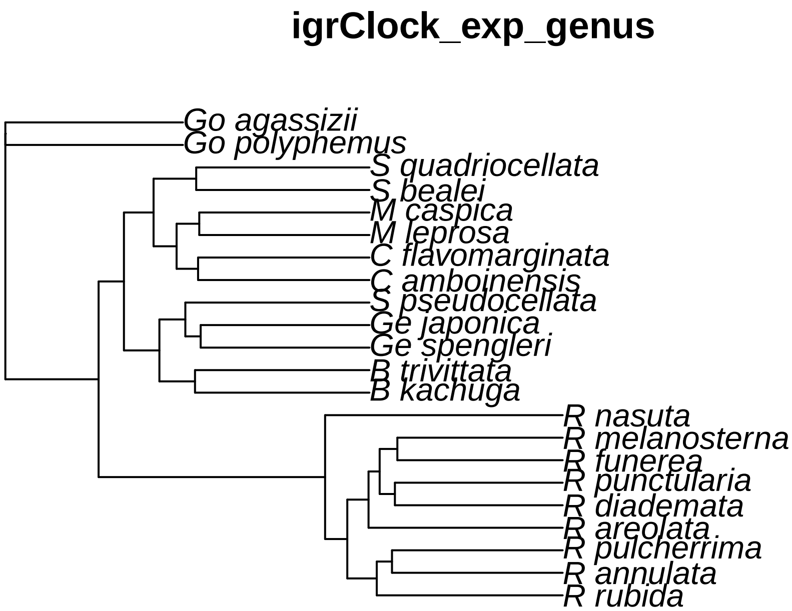
```
RhinoclemmysData$nameTrees <- rownames(RhinoclemmysData$matrixPD) <-
                              rownames(RhinoclemmysData$matrixPDPercentage) <-
                                       gsub("Rhinoclemmys_","",
                                           gsub("_data.nexus.con.tre","",
                                                      treeFiles))

RhinoclemmysData$nameAreas <- colnames(RhinoclemmysData$matrixPD) <-
                              colnames(RhinoclemmysData$matrixPDPercentage) <-
                              rownames(RhinoclemmysData$dist)

RhinoclemmysData$nameTerminals <- colnames(RhinoclemmysData$dist)
```

now we can plot the trees

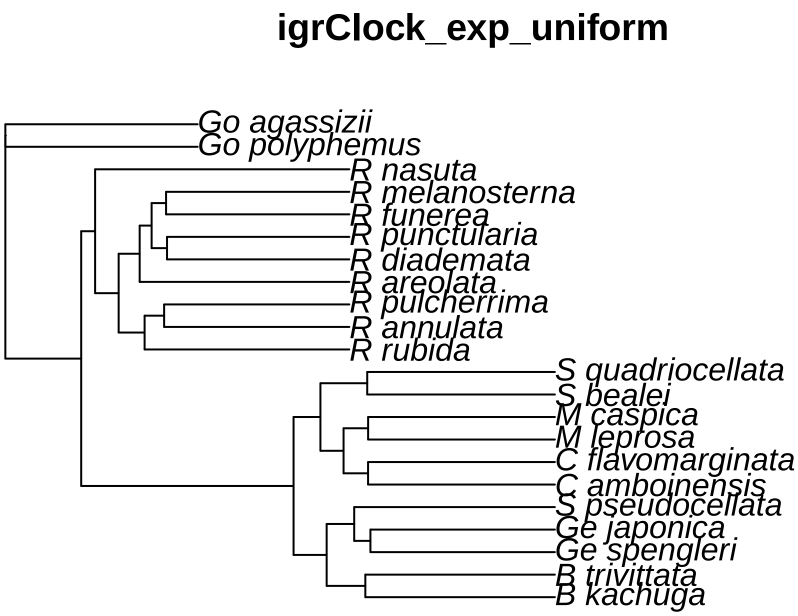
```
for(treeNumber in 1:length(treeFiles)){

cat(RhinoclemmysData$nameTrees[[treeNumber]],"\n")
```

```
    plot.phylo(RhinoclemmysData$trees[[treeNumber]],
               main=RhinoclemmysData$nameTrees[[treeNumber]])



}
```

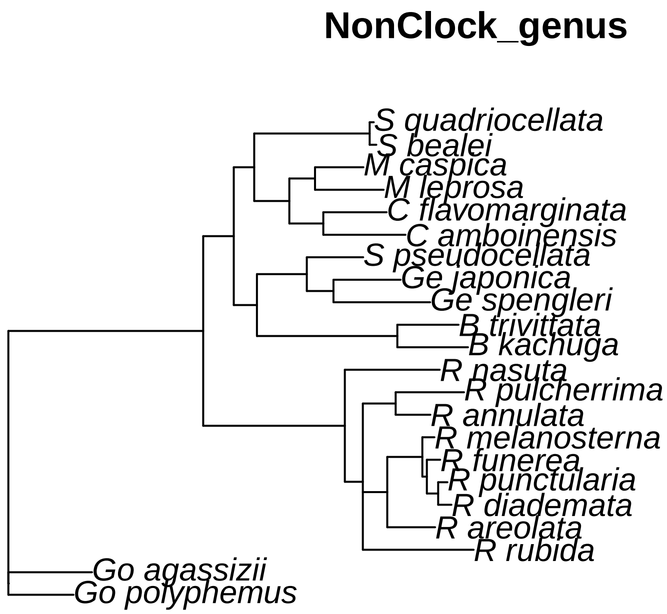
```
## igrClock_exp_genus
```




```
## igrClock_exp_uniform
```




```
## NonClock_genus
```



We perform some basic calculations:

```
# eval tree function

RhinoclemmysData$evalPD <- lapply(RhinoclemmysData$trees,
                                  FUN=evalTree,
                                  distribution=RhinoclemmysData$dist,
                                  root=TRUE)



# sum of deltas
RhinoclemmysData$sumDeltas <-   sapply(RhinoclemmysData$evalPD,
                                    function (x) {sum(x$abDelta/x$deltaPD)})



# max PD value

RhinoclemmysData$maxPD <- colnames(RhinoclemmysData$matrixPD)[apply(
                                      RhinoclemmysData$matrixPD,1,which.max)]




RhinoclemmysData$matrixPDPercentage
```

```
##                          A     B    C    D    E     F    G    H    I    J
## igrClock_exp_genus   16.40 18.92 2.20 2.20 4.51 17.26 4.51 4.51 4.51 4.51
## igrClock_exp_uniform 12.56 14.89 2.06 2.06 5.88 17.77 5.88 5.88 5.88 5.88
## NonClock_genus       11.98 10.56 1.08 1.38 7.58 15.82 6.47 6.55 6.55 6.55
##                         K    L    M    N
## igrClock_exp_genus   4.51 4.51 4.51 6.91
## igrClock_exp_uniform 5.88 5.88 5.88 3.68
## NonClock_genus       6.19 6.19 5.86 7.24
```

```r
RhinoclemmysData$maxPD
```

```
## [1] "B" "F" "F"
```

Dependending on the tree/clock, PD prefers whether area B or F, but we do not know if the values in the analyses are close enough to consider the difference in PD values an artifact or a real difference given the tree / clock used.

```r
# swap branch lengths, allswap


RhinoclemmysData$swapBLalls <- lapply(RhinoclemmysData$trees,
                                    FUN=swapBL,
                                    distribution=RhinoclemmysData$dist,
                                    root=TRUE)
```

```
## model to test allswap reps 100
## model to test allswap reps 100
## model to test allswap reps 100
```

```r
RhinoclemmysData$swapBLalls
```

```
## [[1]]
##   AreaSelected Freq
## 1          B *   98
## 2            F    2
##
## [[2]]
##   AreaSelected Freq
## 1          F *  100
##
## [[3]]
##   AreaSelected Freq
## 1            B    8
## 2          F *   92
```

```r
RhinoclemmysData$swapBLunif <- lapply(RhinoclemmysData$trees,
                                    FUN=swapBL,
                                    distribution=RhinoclemmysData$dist,
                                    model = "uniform",
                                    root=TRUE)
```

```
## model to test uniform reps 100
## model to test uniform reps 100
## model to test uniform reps 100
```

```r
RhinoclemmysData$swapBLunif
```

```
## [[1]]
##   AreaSelected Freq
## 1          B *   94
## 2            F    6
##
## [[2]]
##   AreaSelected Freq
## 1          F *  100
##
```

```
## [[3]]
##   AreaSelected Freq
## 1            B    2
## 2          F *   98
```

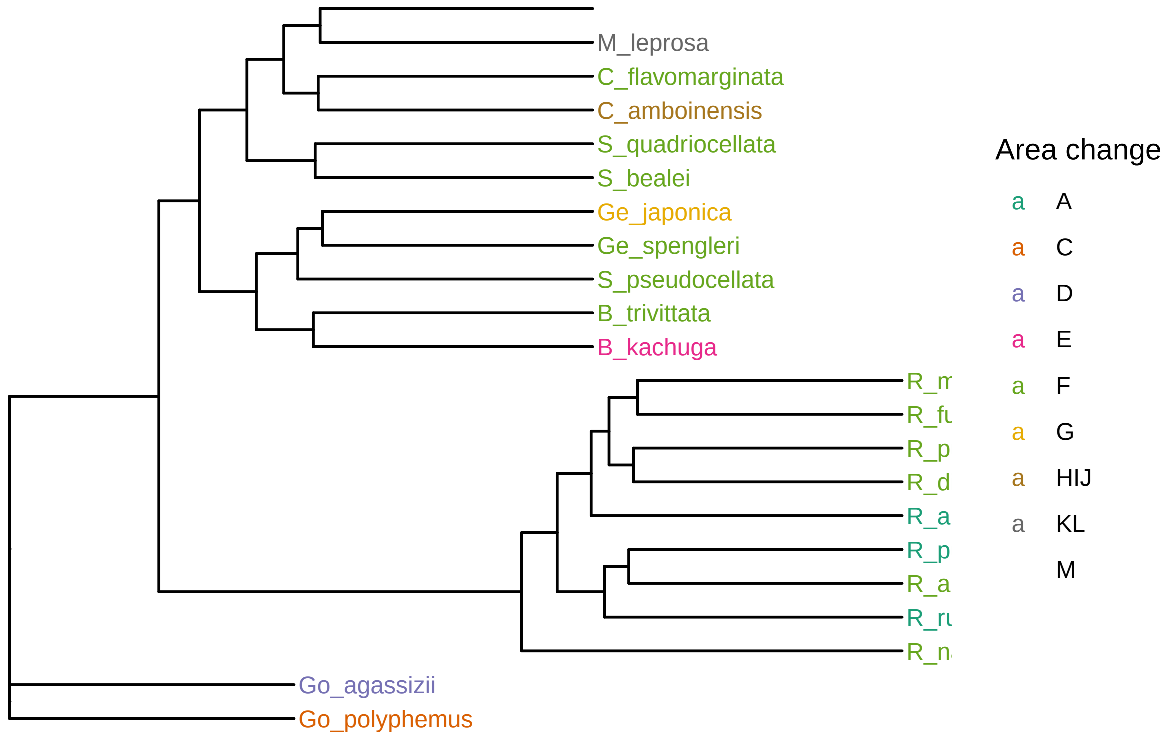
In this case, the tree selected changes the area selected, and we are confident in the results as in the three cases the branch swap does not affect the results.

And plotted into the trees:

```r
options(warn=-1)

for(treeNumber in 1:length(treeFiles)){

cat(RhinoclemmysData$nameTrees[[treeNumber]],"\n")


theTitle <- paste("Initial area selected:",RhinoclemmysData$evalPD[[treeNumber]]$InitialArea[1])



p0 <-    ggtree(RhinoclemmysData$trees[[treeNumber]], layout="rectangular", ladderize=TRUE,
                ##color=c("red","blue","red","blue","black","black","black"),
                color=c("black"),
                 size=0.5 ) +
         theme(legend.position="right") +
         labs(title = theTitle)



p <- p0 %<+% RhinoclemmysData$evalPD[[treeNumber]] +
             geom_tiplab(aes(color=areaDelta), size =3) +
             scale_colour_brewer("Area change", palette="Dark2")


print(p)


}
```

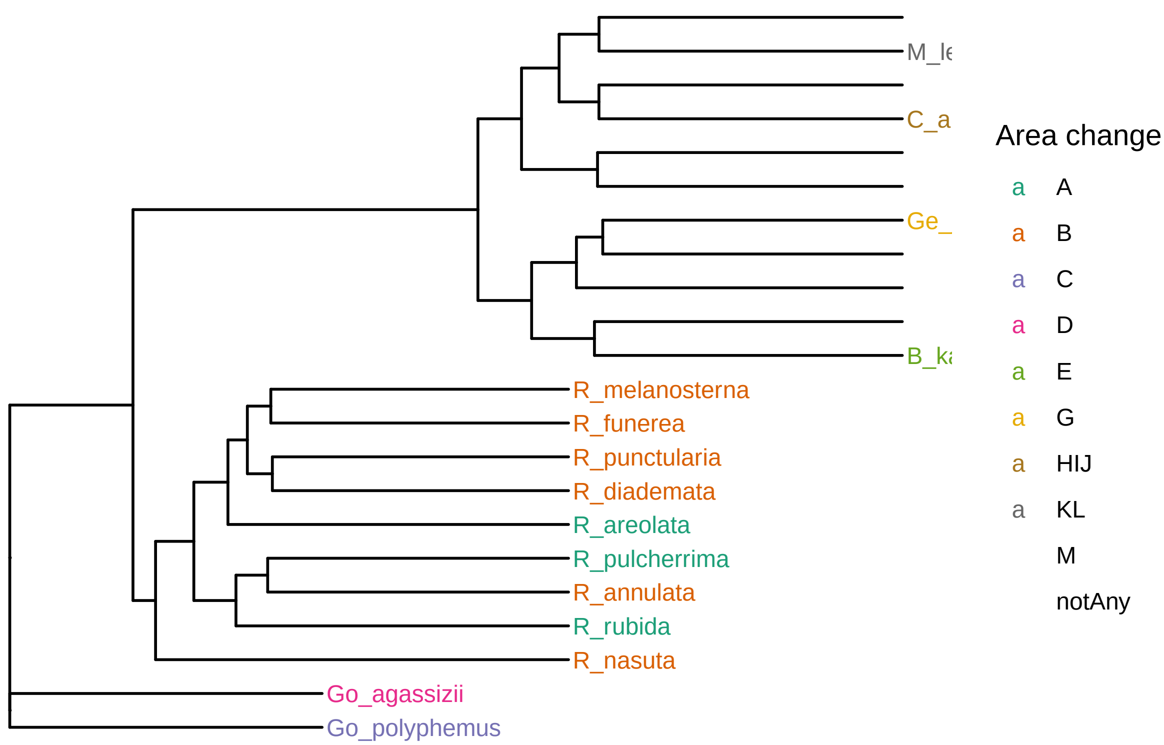
```
## igrClock_exp_genus
```

Initial area selected: B



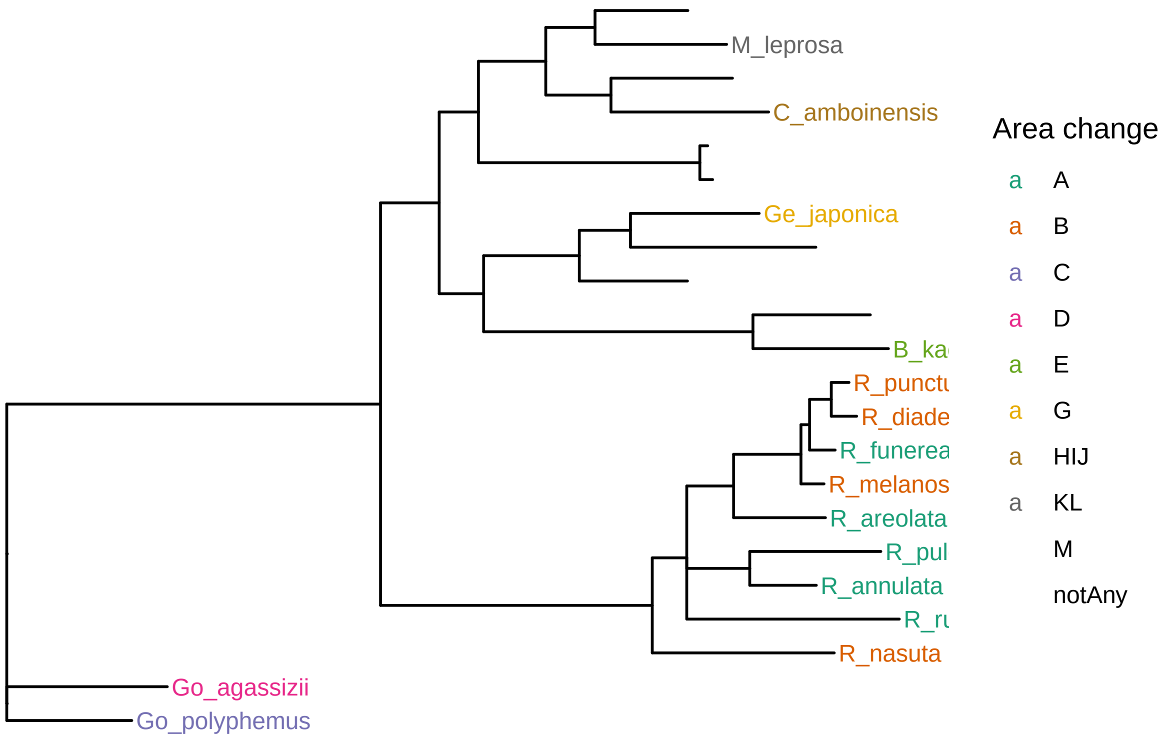

```
## igrClock_exp_uniform
```

Initial area selected: F



```
## NonClock_genus
```

Initial area selected: F



As this is too messy, we will focus our attention in areas B and F

```
options(warn=-1)

for(treeNumber in 1:length(treeFiles)){

if(treeNumber == 1){tmpCol <- cbind(RhinoclemmysData$evalPD[[1]][,1],
                                 RhinoclemmysData$evalPD[[treeNumber]]$areaDelta)
                   }else{
                      tmpCol <- cbind(tmpCol,RhinoclemmysData$evalPD[[treeNumber]]$areaDelta)
                    }

}


rowNames <- tmpCol[,1]

tmpCol <- tmpCol[,-1]

colnames(tmpCol) <- RhinoclemmysData$nameTrees

row.names(tmpCol) <- rowNames

tmpCol
```

```
##                 igrClock_exp_genus igrClock_exp_uniform NonClock_genus
## Go_polyphemus   "C"                "C"                  "C"
```

```
## Go_agassizii     "D"                 "D"                  "D"
## R_rubida         "A"                 "A"                  "A"
## R_nasuta         "F"                 "B"                  "A"
## R_areolata       "A"                 "A"                  "B"
## R_diademata      "F"                 "B"                  "B"
## R_punctularia    "F"                 "B"                  "A"
## R_funerea        "F"                 "B"                  "B"
## R_melanosterna   "F"                 "B"                  "A"
## R_annulata       "F"                 "B"                  "A"
## R_pulcherrima    "A"                 "A"                  "B"
## B_kachuga        "E"                 "E"                  "E"
## B_trivittata     "F"                 "notAny"             "notAny"
## C_amboinensis    "HIJ"               "HIJ"                "notAny"
## C_flavomarginata "F"                 "notAny"             "G"
## M_leprosa        "KL"                "KL"                 "notAny"
## M_caspica        "M"                 "M"                  "HIJ"
## S_bealei         "F"                 "notAny"             "notAny"
## S_quadriocellata "F"                 "notAny"             "KL"
## Ge_spengleri     "F"                 "notAny"             "M"
## S_pseudocellata  "F"                 "notAny"             "notAny"
## Ge_japonica      "G"                 "G"                  "notAny"
```

In all three cases *R. diademata* changes the initial selection to the second "optional" area, what might suggests that for this species, its distribution (only in area B) or its branch length, could be leading the results.

To test our hypothesis, we can delete the species or we could remove the species or change its branch length to zero, or change its distribution to null, and rerun the analysis.

```
## already done using evalTree; here only _R_diademata_ will be tested.

RhinoclemmysData$evalRdiademata <- lapply(RhinoclemmysData$trees,
                                  FUN=evalTerminal,
                                  distribution=RhinoclemmysData$distribution,
                                  tipToEval = "R_diademata",
                                  approach="lower",
                                  root=TRUE)

RhinoclemmysData$evalRdiademata
```

```
## [[1]]
## branchLengthChange    bestInitialArea   bestModifiedArea
##           "2.7137"                "B"                "F"
##      initialLength
##          "13.2663"
##
## [[2]]
##           maxPD bestInitialArea  unModifiedArea   initialLength
##             "0"             "F"             "*"       "14.4657"
##
## [[3]]
##           maxPD bestInitialArea  unModifiedArea   initialLength
##             "0"             "F"             "*"        "0.0062"
```

```
## removing the distribution of the species
```

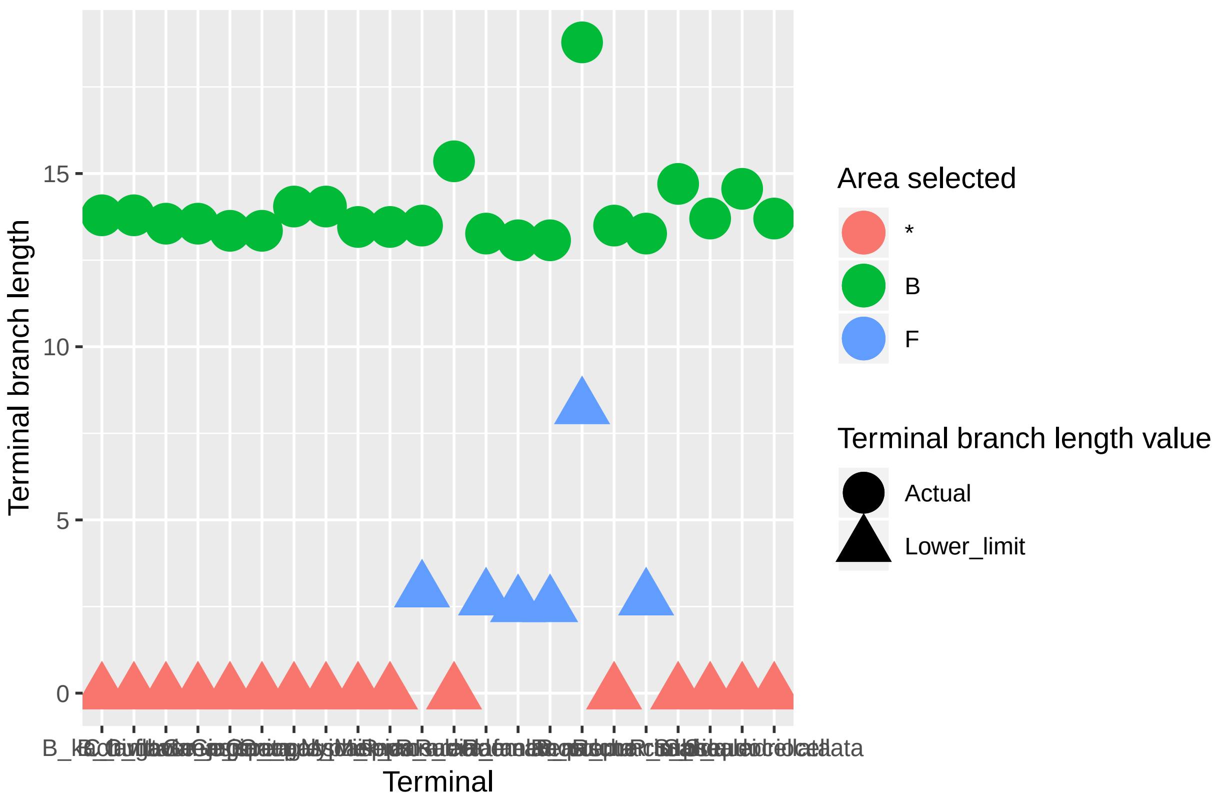
```r
RhinoclemmysData$distribution[,13] <- rep(0,14)

RhinoclemmysData$newPD <- lapply(RhinoclemmysData$trees,
                                  FUN=PDindex,
                                  distribution=RhinoclemmysData$dist,
                                  root=TRUE)


RhinoclemmysData$newMatrixPD <- as.data.frame(
                                  matrix(
                                      unlist(RhinoclemmysData$newPD),
                                      nrow= length(treeFiles),
                                      byrow=TRUE))

colnames(RhinoclemmysData$newMatrixPD) <- rownames(RhinoclemmysData$dist)

RhinoclemmysData$newMaxPD <- colnames(RhinoclemmysData$newMatrixPD)[apply(
                                      RhinoclemmysData$newMatrixPD,1,which.max)]


RhinoclemmysData$newMaxPD
```

```
## [1] "F" "F" "F"
```

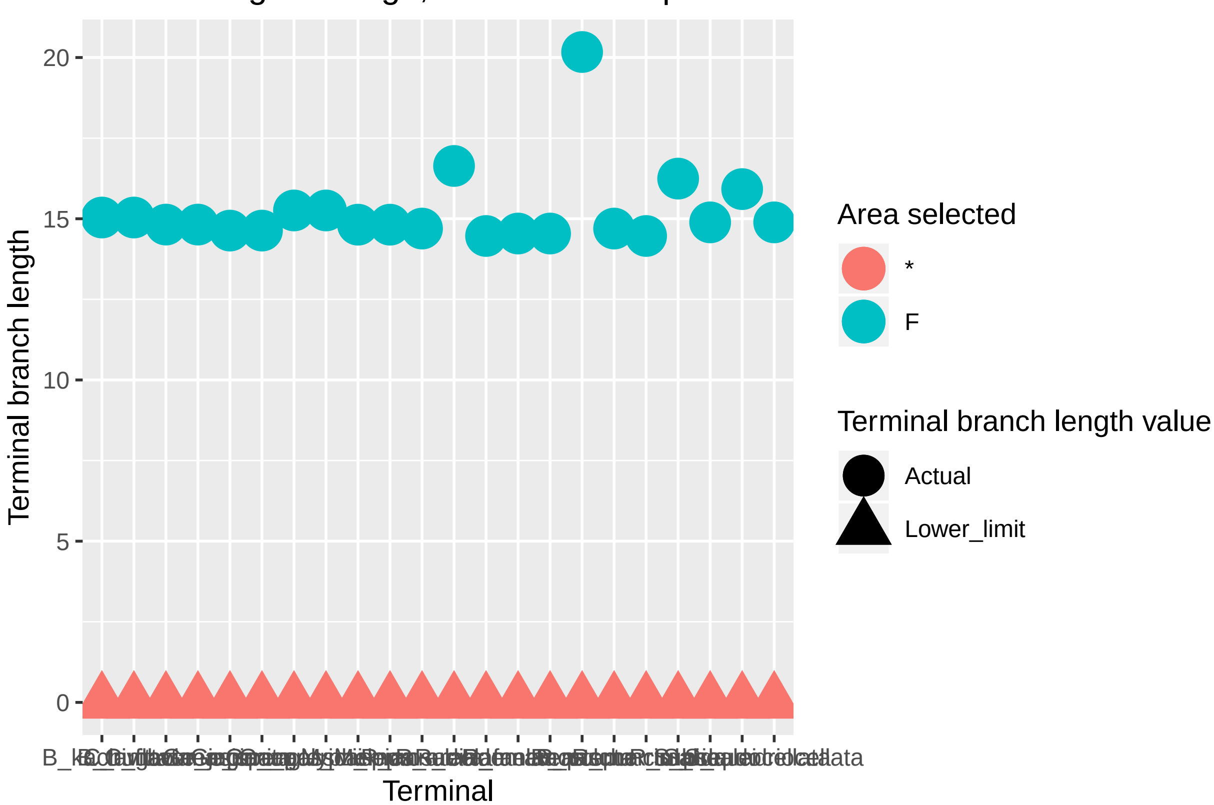
As expected, now all trees give the same result, they select the area F. While this effect could be assigned to the distribution alone -the species is a singleton-. most of the species (17 out of 22) are singletons, therefore, the most plausible explanation must include the branch length.

```r
for(treeNumber in 1:length(treeFiles)){

##print(RhinoclemmysData$evalPD[[treeNumber]])

finalResults <- RhinoclemmysData$evalPD[[treeNumber]]



plotResults <- ggplot(data=finalResults, aes(x= labelTerminal, y= initialLength,
                          shape="Actual",
                          colour=InitialArea)) +
              geom_point(size= 7) +
              geom_point(aes(x= labelTerminal, y= lowerBranchLength,
                              colour=lowerFinalArea,
                              shape="Lower_limit"), size=7) +
              labs(title = "Branch length change, lower limits. Equal branches.",
                   colour = "Area selected",
                   shape = "Terminal branch length value",
                   y = "Terminal branch length",
                   x = "Terminal")

print(plotResults)


}
```

Branch length change, lower limits. Equal branches.



Branch length change, lower limits. Equal branches.




Branch length change, lower limits. Equal branches.
Terminal branch length
0.05
0.04
0.03
0.02
0.01
0.00
Terminal
Area selected
*
F
Terminal branch length value
Actual
Lower_limit