Branch length evaluation for Phylogenetic Diversity: a worked example

Daniel R. Miranda-Esquivel

2018 - 04 - 01

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)

packageVersion("blepd")
```

```
## [1] '0.1.4.2018.4.1.2110'
```

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

library(ggtree)

library(gridExtra)

library(RColorBrewer)
```

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

```
## trees

data(package = "blepd")

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
q
```

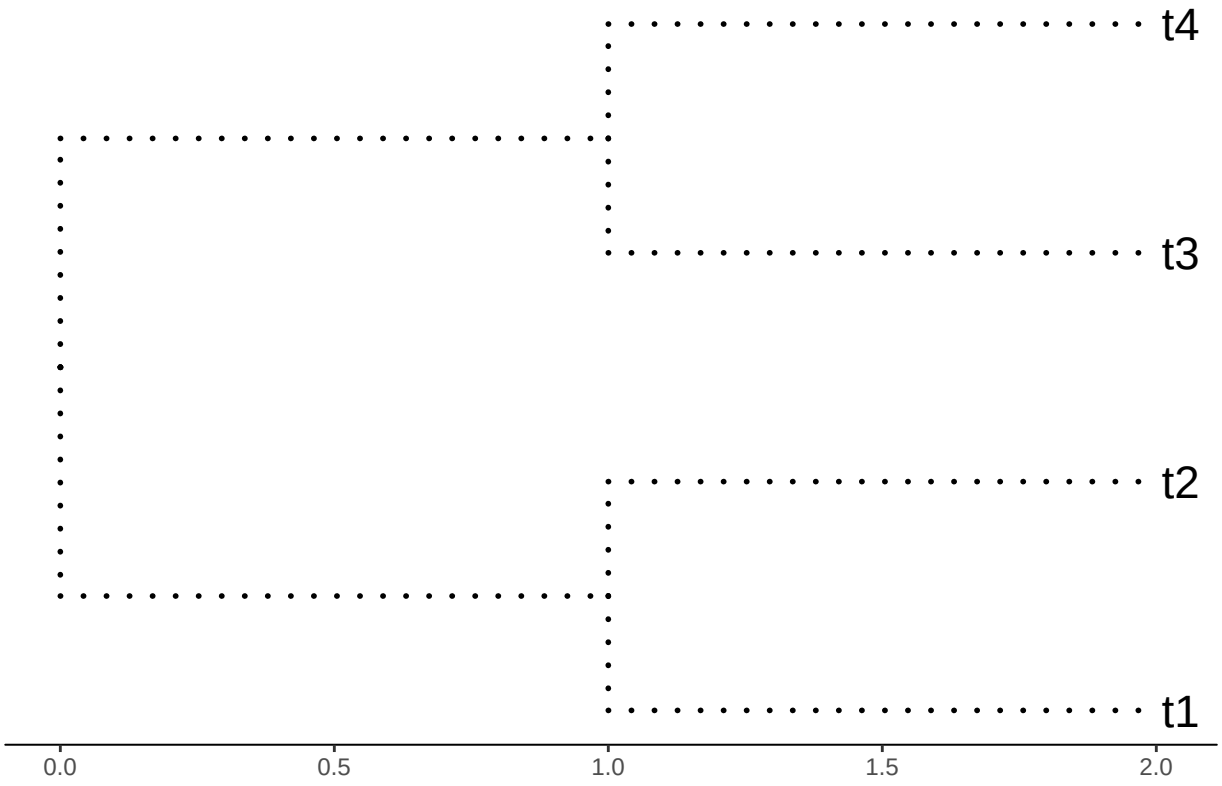
```
## function (save = "default", status = 0, runLast = TRUE)
## .Internal(quit(save, status, runLast))
## <bytecode: 0xa371608>
## <environment: namespace:base>
```

```
data(distribution)

str(distribution)
```

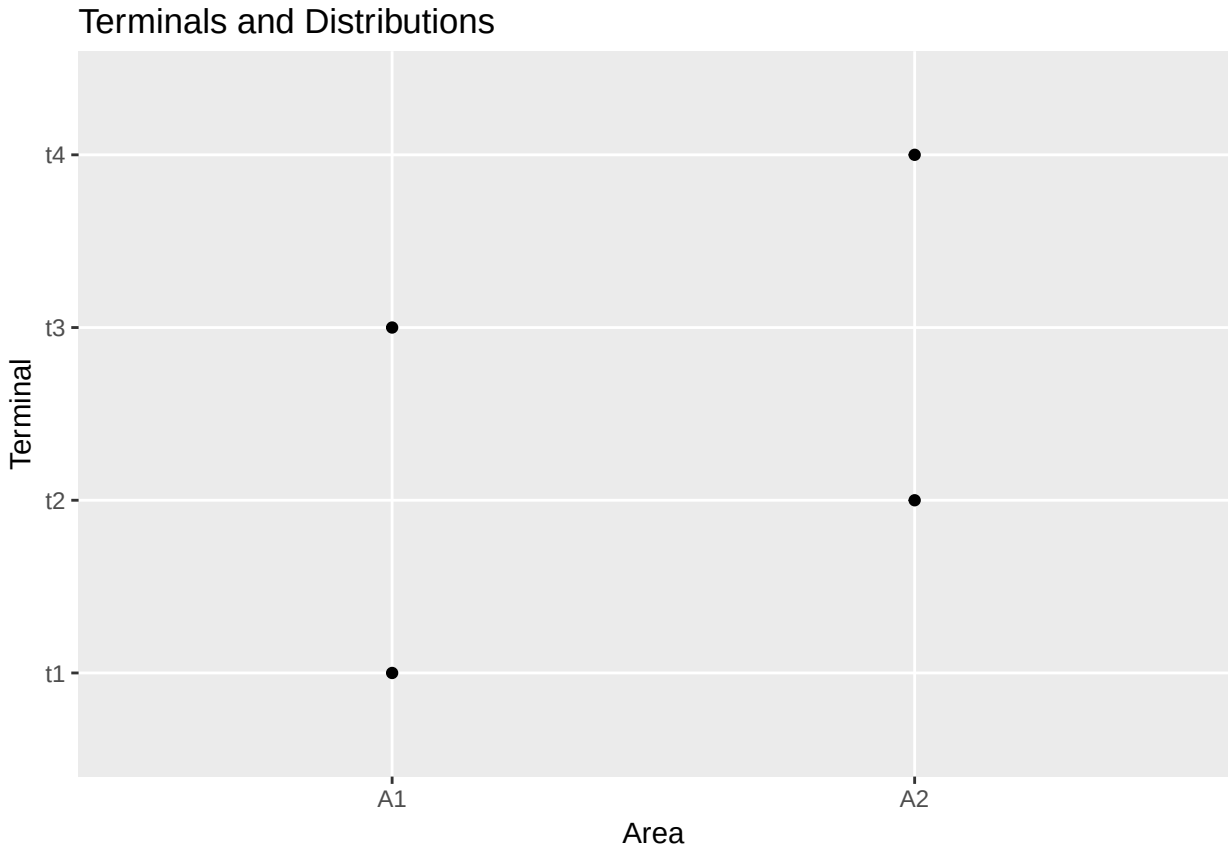
```
##  int [1:2, 1:4] 1 0 0 1 1 0 0 1
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2] "A1" "A2"
##   ..$ : chr [1:4] "t1" "t2" "t3" "t4"
```

```
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)
```

Four terminals, equal branch length




```
## 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 <- myPD(tree=initialTree, distribution = dist4taxa)

initialPD
```

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

Function to evalute a single terminal

To test the effect of changing the branch length for 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
```

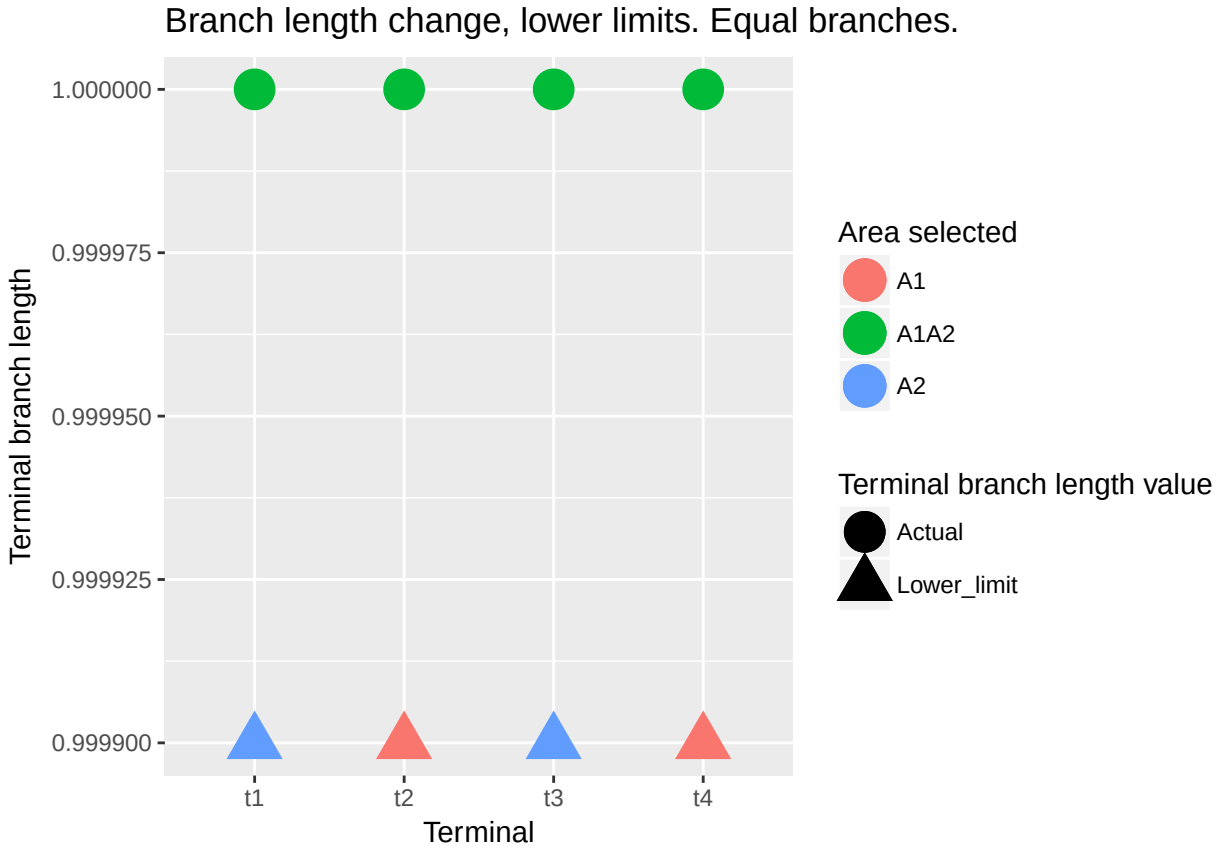
```
##   labelTerminal InitialArea initialLength lowerFinalArea lowerBranchLength
## 1            t1        A1A2             1             A2            0.9999
## 2            t2        A1A2             1             A1            0.9999
## 3            t3        A1A2             1             A2            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)
```



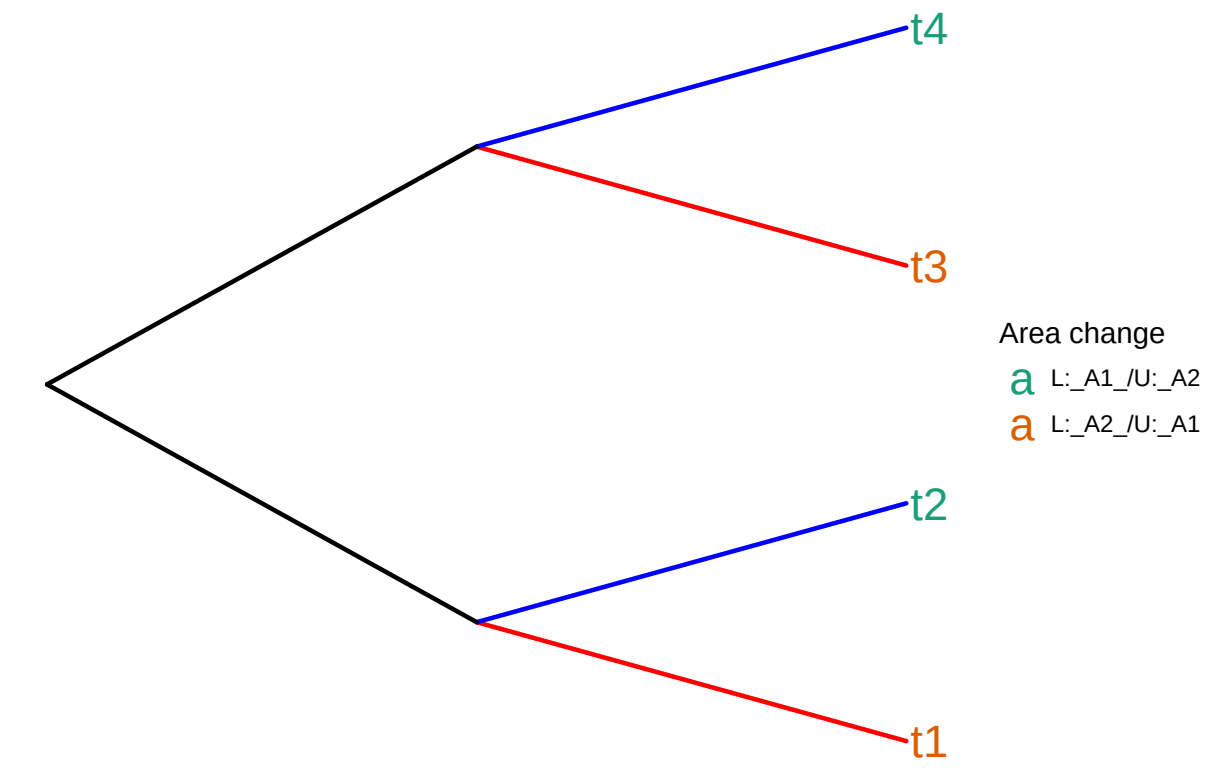
or plotted as a simple table:

```
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 two 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.

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

```
## model to test simpleswap 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 a random uniform brach length distribution

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

```
## model to test uniform 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.