

Branch length evaluation for Phylogenetic Diversity: a worked example

Daniel R. Miranda-Esquivel

2018 - 02 - 07

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] '0.1.4.2018.1.22.2218'
```

```
## 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

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

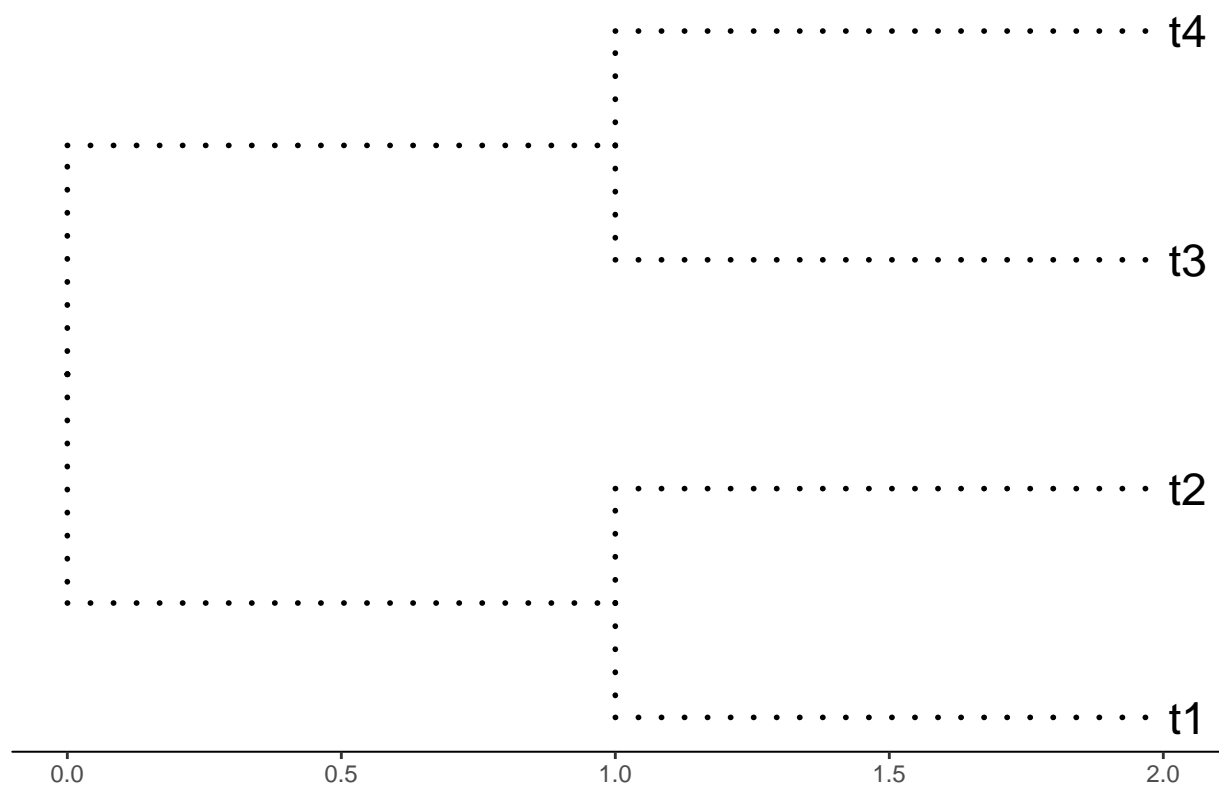
## the tree

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

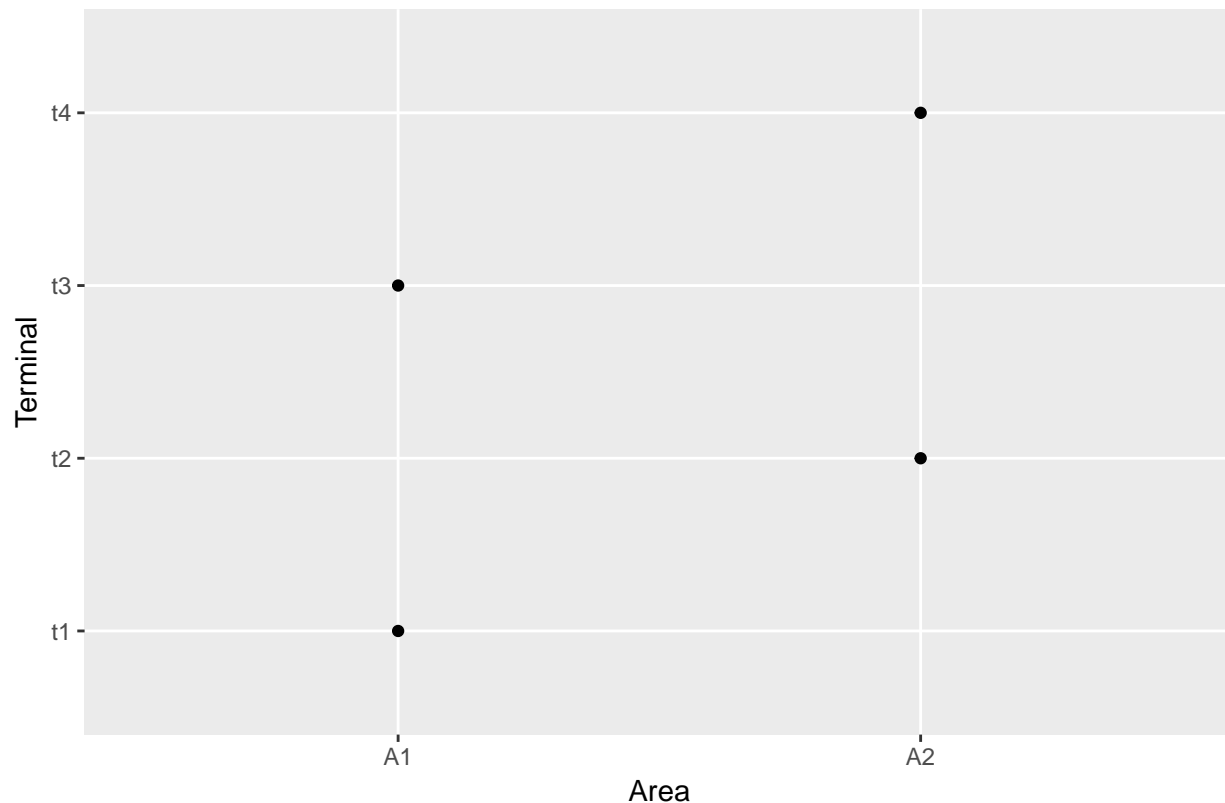


```
## the distribution

plotDistrib <- ggplot(data=distXY,
  aes(x= Area, y= Terminal),
  size =11) +
  geom_point() +
  labs(title = "Terminals and Distributions",
    y = "Terminal",
    x = "Area")

print(plotDistrib)
```

Terminals and Distributions



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 evaluate 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 brach 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 branch length will modify the area selected from A1A2 to A2, as the tie between the paths 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

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 lowerBranchLength InitialArea lowerFinalArea initialLength
## 1            t1           0.9999      A1A2           A2             1
## 2            t2           0.9999      A1A2           A1             1
## 3            t3           0.9999      A1A2           A2             1
## 4            t4           0.9999      A1A2           A1             1
##   upperBranchLength upperFinalArea changeLower changeUpper deltaUpper
## 1           1.0001           A1           A2           A1       1e-04
## 2           1.0001           A2           A1           A2       1e-04
## 3           1.0001           A1           A2           A1       1e-04
## 4           1.0001           A2           A1           A2       1e-04
##   deltaLower deltaPD areaDelta abDelta
## 1       1e-04      0 LU_A2_A1      0
## 2       1e-04      0 LU_A1_A2      0
## 3       1e-04      0 LU_A2_A1      0
## 4       1e-04      0 LU_A1_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 change the area selected.

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

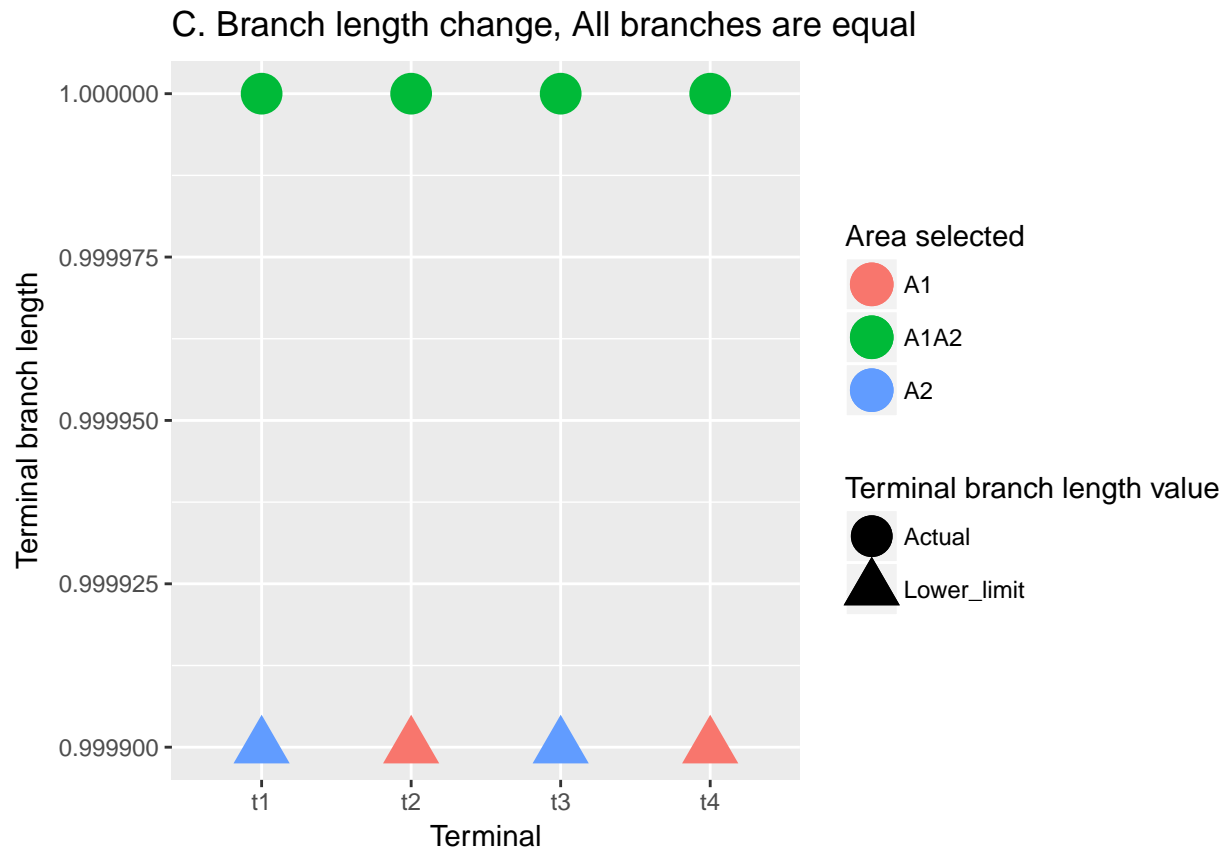
```
plotResults <- ggplot(data=finalResults, aes(x= labelTerminal, y= initialLength,
      shape="Actual",
      colour=InitialArea)) +
  geom_point(size= 7) +
  geom_point(aes(x= labelTerminal, y= lowerBranchLength,
      colour=lowerFinalArea,
      shape="Lower_limit"), size=7) +
  labs(title = "C. Branch length change, All branches are equal",
```

```

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  LU_A1_A2     2
## 2  LU_A2_A1     2

```

or plotted into the tree:

```
p0 <- ggtree(initialTree, layout="slanted", ladderize=TRUE,  
             color="black", size=0.8 ) +  
      theme(legend.position="right") +  
      labs(title = paste("Initial area selected: ",finalResults$maxPD, sep = ""))  
  
p <- p0 %<+% finalResults + geom_tiplab(aes(color=areaDelta), size =6) +  
      scale_colour_brewer("Area change", palette="Dark2")  
  
print(p)
```

Initial area selected:

