

## R scripts and examples

The R function “DP” calculates tree-based dissimilarity between pairs of plots taking into account the functional or phylogenetic dissimilarities between species. With this function, tree-based dissimilarity is calculated according to Equation 2 in the main text with the Bray-Curtis index. This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License <http://www.gnu.org/licenses/>. It will be integrated in version 1.4 of the adiv package of R. <https://cran.r-project.org/web/packages/adiv/index.html>.

**Disclaimer:** users of this code are cautioned that, while due care has been taken and it is believed accurate, it has not been rigorously tested and its use and results are solely the responsibilities of the user.

**Description:** given a matrix of  $N$  plots  $\times$   $S$  species’ relative or absolute abundance value, together with an  $S \times S$  dissimilarity matrix, the function “DP” calculates a semimatrix with the values of the dissimilarity index for each pair of plots, as proposed in the main text.

**Dependencies:** package adiv.

**Usage:** `DP(mtree, comm, height = NULL, diag = FALSE, upper = FALSE, tol = 0.001)`

### Arguments:

`mtree`: an object inheriting the class `phylo` (see package `ape`), `phylo4` (see package `phylobase`), or `hclust`. The tree must be ultrametric: equal distance from any tip to the root.

`comm`: a matrix of  $N$  plots  $\times$   $S$  species containing the relative or absolute abundance of all species. Columns are species and plots are rows. Column labels (species names) should be assigned as in `mtree`.

`height`: either `NULL` or a numeric. See details.

`diag`: a logical value indicating whether the diagonal of the distance matrix should be printed by function `print.dist`.

`upper`: a logical value indicating whether the upper triangle of the distance matrix should be printed by function `print.dist`.

`tol`: a tolerance threshold. A value between `-tol` and `tol` is considered as zero. See details.

### Value:

The function returns a (semi-)matrix of class `dist` with the values of the proposed dissimilarities for each pair of plots.

### Details:

Object `mtree` defines a tree with species as tips. If argument `height` is `NULL`, then the root of the tree will be placed at the most recent common ancestor of all species occurring in the set of plots (given in object `comm`). An alternative position for the root can be given by specifying the height of the tree (argument `height`). In that case, `height` must be higher than the distance between tips and the most recent common ancestor of all species.

The tolerance threshold `tol` is particularly important if your tree is not exactly ultrametric due to approximation problems. In that case, the distance from tip to root varies according to the tip considered, although it should not (variations are due to approximation problems). A difference smaller than `tol` in the distance to root for two species will thus be considered as null.

## Function Syntax:

```
DP <-  
function(mtree, comm, height = NULL, diag = FALSE, upper = FALSE, tol =  
0.001){  
  
  if(! (inherits(comm, "matrix") | inherits(comm, "data.frame")))  
    stop("comm must be a matrix or a data frame")  
  ncom <- nrow(comm)  
  if (is.null(colnames(comm)))  
    stop("comm must have names for column")  
  if (ncom < 2)  
    stop("At least two rows for comm are required")  
  if (is.null(colnames(comm)))  
    stop("comm must have names for column")  
  TA <- tecAptree(mtree, tol = tol)  
  if (any(!colnames(comm) %in% names(TA$list[[1]])))  
    stop("comm contains tip names that are not available in mtree")  
  if (any(comm < 0))  
    stop("comm should contain nonnegative values")  
  if (any(rowSums(comm) == 0))  
    stop("empty communities should be discarded in comm")  
  sp_names <- names(TA$list[[1]])  
  comm <- comm[, sp_names]  
  FUN_COM <- function(groups){  
    COM <- apply(comm, 1, function(x) tapply(x, groups, sum))  
    return(t(COM))  
  }  
  FUN_BC <- function(tab){  
    d <- matrix(0, ncom, ncom)  
    funBC <- function(x) {  
      p <- tab[x[1], ]  
      q <- tab[x[2], ]  
      ps <- p[(p + q) > 0]  
      qs <- q[(p + q) > 0]  
      w <- sum(abs(ps - qs))/sum(ps + qs)  
      return(w)  
    }  
    index <- cbind(col(d)[col(d) < row(d)],  
      row(d)[col(d) < row(d)])  
    d <- unlist(apply(index, 1, funBC))  
    return(d)  
  }  
  LISTCOM <- lapply(TA$list, FUN_COM)  
  LISTd <- lapply(LISTCOM, FUN_BC)  
  d <- LISTd[[1]] * TA$plength[1]  
  for(i in 2:length(LISTd)) {  
    d <- d + LISTd[[i]] * TA$plength[i]  
  }  
  if(!is.null(height) && is.numeric(height) && height >  
sum(TA$plength))  
    d <- d / height  
  else  
    d <- d / sum(TA$plength)  
  attr(d, "Size") <- ncom  
  attr(d, "Labels") <- rownames(comm)  
  attr(d, "Diag") <- diag
```

```

attr(d, "Upper") <- upper
attr(d, "method") <- "DP"
attr(d, "call") <- match.call()
class(d) <- "dist"
return(d)
}

```

### Example:

Load in R the table contained in the Appendix S2 of Ricotta et al. (2016) with species names as row names and name the table `FullTab`. Take care to put an underscore between Genus and species names. The first six rows of this table will thus be:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
Achillea_moschata	0	1	0	1	0	0	1	0	0	2	0	0	0
Adenostyles_leucophylla	0	0	0	0	2	2	2	1	2	0	0	0	0
Agrostis_rupestris	2	1	0	2	2	0	2	2	0	2	0	1	0
Agrostis_schraderiana	0	0	0	0	0	0	0	0	0	0	0	0	0
Antennaria_dioica	0	0	0	0	0	0	0	1	0	0	0	0	0
Anthoxanthum_odoratum	0	0	2	1	2	0	0	0	0	2	0	0	1
	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24		
Achillea_moschata	0	0	0	0	2	1	2	0	1	2	2		
Adenostyles_leucophylla	0	0	0	1	0	0	0	0	0	0	0		
Agrostis_rupestris	0	0	0	2	0	0	1	0	3	2	0		
Agrostis_schraderiana	0	0	0	0	0	0	0	0	2	0	0		
Antennaria_dioica	0	0	0	0	1	1	0	0	0	0	0		
Anthoxanthum_odoratum	0	0	0	0	3	4	1	1	1	1	0		
	X25	X26	X27	X28	X29	X30	X31	X32	X33	X34	X35		
Achillea_moschata	0	4	3	0	3	2	2	3	2	2	2		
Adenostyles_leucophylla	0	0	0	0	0	0	0	0	0	0	0		
Agrostis_rupestris	2	3	1	2	4	1	1	2	2	1	3		
Agrostis_schraderiana	0	0	0	0	0	1	0	0	0	0	0		
Antennaria_dioica	0	0	0	0	2	1	1	0	1	1	0		
Anthoxanthum_odoratum	2	1	3	1	2	1	2	1	2	1	2		
	X36	X37	X38	X39	X40	X41	X42	X43	X44	X45	X46		
Achillea_moschata	2	2	2	1	2	2	1	3	1	2	1		
Adenostyles_leucophylla	0	0	0	0	0	0	0	0	0	0	0		
Agrostis_rupestris	2	2	2	2	2	1	0	0	0	0	0		
Agrostis_schraderiana	0	0	0	0	0	0	2	0	0	0	0		
Antennaria_dioica	1	0	0	1	0	1	0	0	0	0	2		
Anthoxanthum_odoratum	1	1	0	3	2	3	1	2	1	0	1		
	X47	X48	X49	X50	X51	X52	X53	X54	X55	X56	X57		
Achillea_moschata	0	2	3	0	0	0	0	0	0	0	0		
Adenostyles_leucophylla	0	0	0	0	0	0	0	0	0	0	0		
Agrostis_rupestris	1	0	1	0	0	0	1	2	0	0	2		
Agrostis_schraderiana	0	0	0	0	0	0	0	0	0	0	0		
Antennaria_dioica	1	0	0	0	0	1	2	1	0	1	0		
Anthoxanthum_odoratum	1	0	4	2	2	2	2	2	2	1	2		
	X58	X59		C		S		R					
Achillea_moschata	0	0		38.57		23.81		37.62					
Adenostyles_leucophylla	0	0		49.59		0.00		50.41					
Agrostis_rupestris	1	1		17.45		65.26		17.29					
Agrostis_schraderiana	0	0		41.97		39.47		18.56					
Antennaria_dioica	0	0		28.92		56.75		14.34					
Anthoxanthum_odoratum	0	0		17.65		60.53		21.82					

The species traits are in the columns 60, 61 and 62 of this table (columns named "C", "S", and "R"):

```
Traits <- FullTab[60:62]
```

The abundances of species in plots are in the remaining columns:

```
ab <- FullTab[-(60:62)]
ab <- t(ab)
```

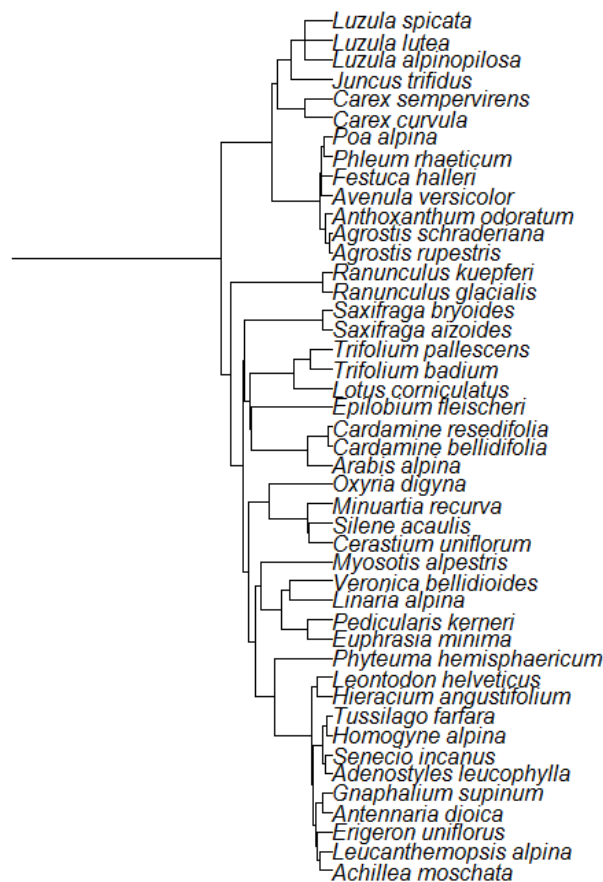
In `ab`, plots are in rows and species in columns.

Then we need to perform the functional clustering with the unweighted pair group method with arithmetic mean (UPGMA) on the Bray-Curtis distance applied to species' traits.

```
library(vegan)
library(ape)
H <- hclust(vegdist(Traits), "average") # By default, function vegdist in
package vegan calculates the Bray Curtis coefficient.
```

With function `read.tree` of package `ape`, load in R the phylogenetic tree contained in the Appendix A of Ricotta et al. (2015) and name it `phy`.

The tree is as follows:



Load function `DP` in R.

The species-based dissimilarities between plots are:

```
Ds <- vegdist(ab)
```

The functional dissimilarities are:

```
library(adiv)
Df <- DP(H, ab, height=0.5, tol=0.00001)
```

and the phylogenetic dissimilarities are:

```
Dp <- DP(phy, ab, tol=0.00001)
```

In function DP, the parameter "height" allows to control the scaling procedure to unit height from the tips to the root. The default was used for the phylogenetic tree which means that the root of the tree was placed at the most recent common ancestor of all species occurring in the set of plots, and the distance from tips to root was set to unity. In the UPGMA tree obtained above, the functional dissimilarity between two species is estimated by twice the way between each of these species and their first shared interior node, which is equal to the sum of branch lengths in the shortest path that connects the two species on the tree. If  $d_{\max}$  is the maximum possible functional dissimilarity between two species, then the root of the tree must thus be placed at a distance of  $d_{\max}/2$  from the tips. To normalize the tree height, the branch lengths on the tree must then be divided by  $d_{\max}/2$  so that the height of the tree is equal to 1. We used the Bray-Curtis index to calculate functional dissimilarities between species. This index is bounded between 0 and 1, so that  $d_{\max}=1$  and  $d_{\max}/2=0.5$  (height=0.5). The dissimilarity matrices Df and Dp can then be included in further analyses, such as the analysis of variance using distance matrices (function `adonis` in package `vegan`). In that case, a factor defining in which successional stage each plot belongs to can be defined as follows:

```
FAC <- rep(c("early", "mid", "late"), c(17, 32, 10))
```

In FAC, "early" means "Early-successional stage", "mid" means "Mid-successional stage" and "late" means "Late-successional stage".

The functional and phylogenetic uniqueness values can be computed as follows:

```
Uf <- Df/Ds
Up <- Dp/Ds
```

Also note that the square root of these matrices (Ds, Df and Dp) are Euclidean and can be analyzed via a principal coordinate analysis (e.g. function `dudi.pco` in package `ade4`).

All data will be included in version 1.4 of the `adiv` package. We will provide the following name to the data set in `adiv`: `RutorGlacier`.

## References

- Ricotta, C., Bacaro, G., Caccianiga, M., Cerabolini, B.E.L., & Moretti, M. (2015) A classical measure of phylogenetic dissimilarity and its relationship with beta diversity. *Basic and Applied Ecology*, 16, 10–18.
- Ricotta, C., de Bello, F., Moretti, M., Caccianiga, M., Cerabolini, B.E.L., & Pavoine, S. (2016) Measuring the functional redundancy of biological communities: a quantitative guide. *Methods in Ecology and Evolution*, 7, 1386–1395.