

Appendix 3. R scripts

The R function "betaUniqueness" calculates plot-to-plot uniqueness and redundancy taking account of functional dissimilarities between species using equation 5 and 6 in the main text. The R function "dislptransport" calculates the D_{KG} measure of dissimilarity (Kosman 1996; Gregorius et al. 2003) applied here to functional differences between plots (eq. 3 in the main text). 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 2.1 of the adiv package of R: <https://cran.r-project.org/web/packages/adiv/index.html>. The functions were checked and applied with R-4.0.2.

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.

Dependencies: lpSolve (Berkelaar et al. 2020).

Function Syntax:

```
betaUniqueness <- function (comm, dis, Nind = 10000)
{
  if (!(inherits(comm, "data.frame") | inherits(comm, "matrix")))
    stop("comm is not a data.frame or a matrix")
  if(nrow(comm) < 2) stop("comm must have at least two rows")
  D <- as.matrix(dis)
  if (any(D > 1)) {
    D <- D/max(D)
    warnings("All values in dis have been divided by their maximum
(highest observed value)")
  }
  U <- dislptransport(comm, D, diag = FALSE, upper = FALSE, Nind = Nind)
  dis0 <- matrix(1, ncol(comm), ncol(comm))-diag(rep(1, ncol(comm)))
  U0 <- dislptransport(comm, dis0, diag = FALSE, upper = FALSE, Nind =
Nind)
  Ustar <- U0
  Ustar[Ustar<=0] <- 1
  res <- list()
  res$betaUniqueness <- as.matrix(U/Ustar)
  res$betaRedundancy <- 1 - as.matrix(U/Ustar)
  res$dissimilarityGap <- as.matrix(U0-U)
  res$DR <- as.matrix(U0)
  res$DKG <- as.matrix(U)
  return(res)
}

dislptransport <-
function (comm, dis, diag = FALSE, upper = FALSE, Nind = 10000)
{

  if (!(inherits(comm, "data.frame") | inherits(comm, "matrix")))
    stop("comm is not a data.frame or a matrix")
  if(nrow(comm) < 2) stop("comm must have at least two rows")
  D <- as.matrix(dis)
  nlig <- nrow(comm)
  ncol <- ncol(comm)
  d <- matrix(0, nlig, nlig)
  d.names <- row.names(comm)
```

```

df <- as.data.frame(t(comm))
dfp <- t(t(df)/colSums(df))
funlp <- function(x) {
  row_rhs <- round(dfp[,x[1]] * Nind)
  if(sum(row_rhs) < Nind) {
    nmiss <- Nind - sum(row_rhs)
    e <- sample((1:nrow(dfp))[row_rhs < dfp[,x[1]] * Nind], nmiss)
    for(i in e) row_rhs[i] <- row_rhs[i] + 1
  }
  if(sum(row_rhs) > Nind) {
    ntoo <- abs(Nind - sum(row_rhs))
    e <- sample((1:nrow(dfp))[row_rhs > dfp[,x[1]] * Nind], ntoo)
    for(i in e) row_rhs[i] <- row_rhs[i] - 1
  }
  col_rhs <- round(dfp[,x[2]] * Nind)
  if(sum(col_rhs) < Nind) {
    nmiss <- Nind - sum(col_rhs)
    e <- sample((1:nrow(dfp))[col_rhs < dfp[,x[2]] * Nind], nmiss)
    for(i in e) col_rhs[i] <- col_rhs[i] + 1
  }
  if(sum(col_rhs) > Nind) {
    ntoo <- abs(Nind - sum(col_rhs))
    e <- sample((1:nrow(dfp))[col_rhs > dfp[,x[2]] * Nind], ntoo)
    for(i in e) col_rhs[i] <- col_rhs[i] - 1
  }
  row_signs <- rep("<=", ncol)
  col_signs <- rep(">=", ncol)
  res <- sum(lp.transport(D, "min", row_signs, row_rhs, col_signs,
  col_rhs)$solution*D/Nind)
  return(res)
}
index <- cbind(col(d)[col(d) < row(d)], row(d)[col(d) < row(d)])
d <- unlist(apply(index, 1, funlp))
attr(d, "Size") <- nlig
attr(d, "Labels") <- d.names
attr(d, "Diag") <- diag
attr(d, "Upper") <- upper
attr(d, "method") <- "GregoriusGilletZiehe"
attr(d, "call") <- match.call()
class(d) <- "dist"
return(d)
}

```

Usage: betaUniqueness <- function (comm, dis, Nind = 10000)
 dislptransport(comm, dis, diag = FALSE, upper = FALSE, Nind = 10000)

Arguments:

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.

dis: a matrix or an object of class dist providing the functional dissimilarities between species (dissimilarities are nonnegative, symmetric, and the dissimilarity between a species and itself is zero). **Species here must be in the same order as in the columns of comm.**

Nind: an integer. The algorithmic index will be applied by assuming that each plot contains Nind individuals. The highest Nind, the most precise the index value will be (see Gregorius et al. 2003, for more details).

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

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

Value:

The function `dislptransport` returns a matrix with the values of the dissimilarity index D_{KG} for each pair of plots. The function `betaUniqueness` returns a list with the following objects:

`betaUniqueness`: a matrix with the values of the proposed beta uniqueness ($U_\beta = D_{KG}/D_R$) for each pair of plots (main text, eq. 6).

`betaRedundancy`: a matrix with the values of the proposed beta redundancy ($R_\beta = 1 - D_{KG}/D_R$) for each pair of plots (main text, eq. 5).

`dissimilarityGap`: a matrix with the values of the dissimilarity gap index ($D_R - D_{KG}$) for each pair of plots.

`DR`: a matrix with the values of the species-based (Rogers) dissimilarity index (D_R) for each pair of plots (main text, eq. 4).

`DKG`: a matrix with the values of the algorithmic functional dissimilarity index (D_{KG}) for each pair of plots (main text, eq. 3).

Example:

Install and load packages `lpSolve`, `vegan` and `adiv`.

```
install.packages("lpSolve")
install.packages("adiv") # Pavoine (2020a,b)
install.packages("vegan") # Oksanen et al. (2019)

library(lpSolve)
library(adiv)
library(vegan)
```

Load the dataset `RutorGlacier` in the `adiv` package of R.

```
data(RutorGlacier)
```

`RutorGlacier` is a list of several tables and a vector. The table named '`Abund`' contains the abundance of plant species in plots. The species traits are in another table named '`Traits`'.

In `RutorGlacier$Abund`, plots are in rows and species in columns.

Here the functional distances between species are calculated as in the main text as the Euclidean distance applied to species' traits, each standardized to zero mean and unit standard deviation :

```
fundis <- dist(scale(RutorGlacier$Traits2[1:6]))
```

Then the resulting functional distances among species were scaled to the unit range by dividing each distance by the maximum value in the distance matrix :

```
fundis <- fundis/max(fundis)
```

Load functions `betaUniqueness` and `dislptransport` in R. To apply function `betaUniqueness` to the `RutorGlacier` dataset, use the following command:

```
frameDKG <- betaUniqueness(RutorGlacier$Abund, fundis)
```

The plot-to-plot values of functional beta uniqueness are:

```
betaU_A <- frameDKG$betaUniqueness
```

and the plot-to-plot values of functional beta redundancy are:

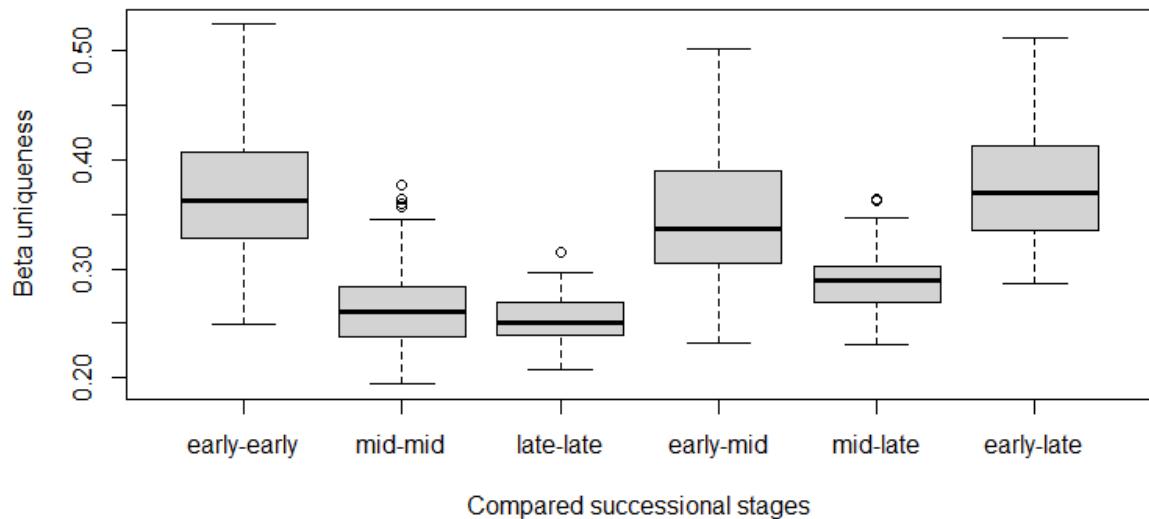
```
betaR_A <- frameDKG$betaRedundancy
```

In RutorGlacier, the object named 'Fac' indicates which level of the successional gradient a given plot belongs to ("early" = early-successional stage, "mid" = mid-successional stage and "late" = late-successional stage). We use `Fac` below to display the functional beta uniqueness between two plots of the same successional stage, and between two plots of distinct successional stages.

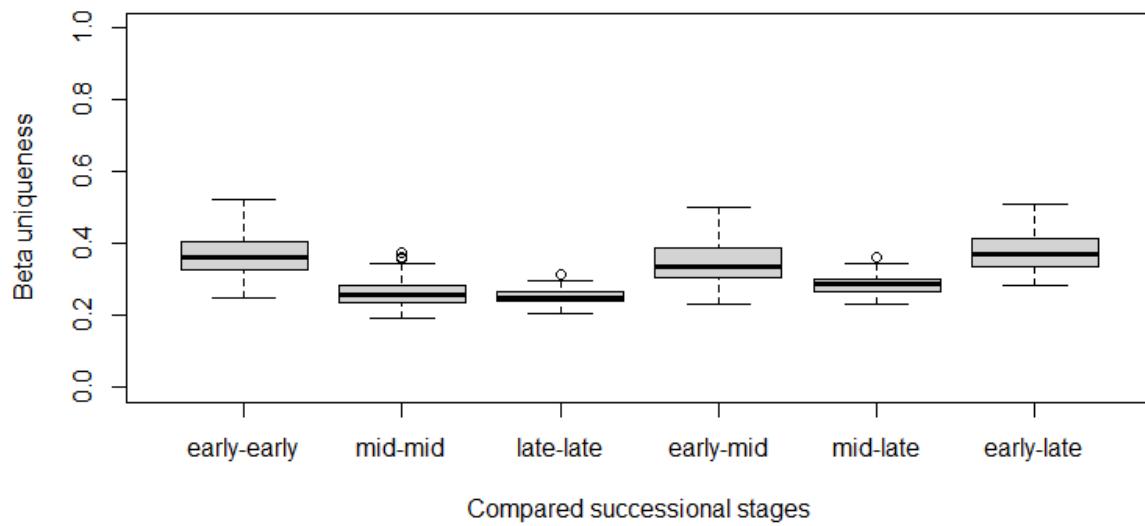
```
f1 <- unlist(sapply(1:58, function(i) rep(RutorGlacier$Fac[i], 59-i)))
f2 <- unlist(sapply(1:58, function(i) RutorGlacier$Fac[-(1:i)]))
f <- paste(f1, f2, sep="-")
F <- factor(f, levels=c("early-early", "mid-mid", "late-late", "early-mid",
"mid-late", "early-late"))

vbetaU_A <- as.vector(as.dist(betaU_A))

boxplot(vbetaU_A~F, ylab="Beta uniqueness", xlab="Compared successional stages")
```



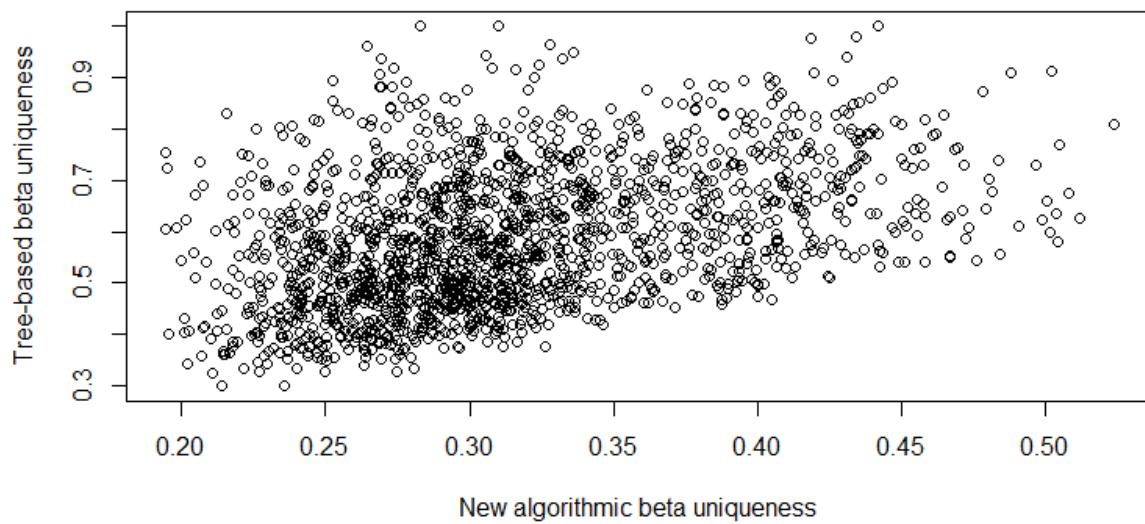
```
boxplot(vbetaU_A~F, ylab="Beta uniqueness", ylim=c(0,1), xlab="Compared successional stages") # change of scale for the Y axis
```



Below, we calculate Ricotta et al. (2020) tree-based index of beta functional uniqueness and compare the results with those provided by the new algorithmic index.

```
H <- hclust(fundis, "average")
Up <- betaTreeUniqueness(H, RutorGlacier$Abund, tol=0.00001)
vUp <- as.vector(as.dist(Up))

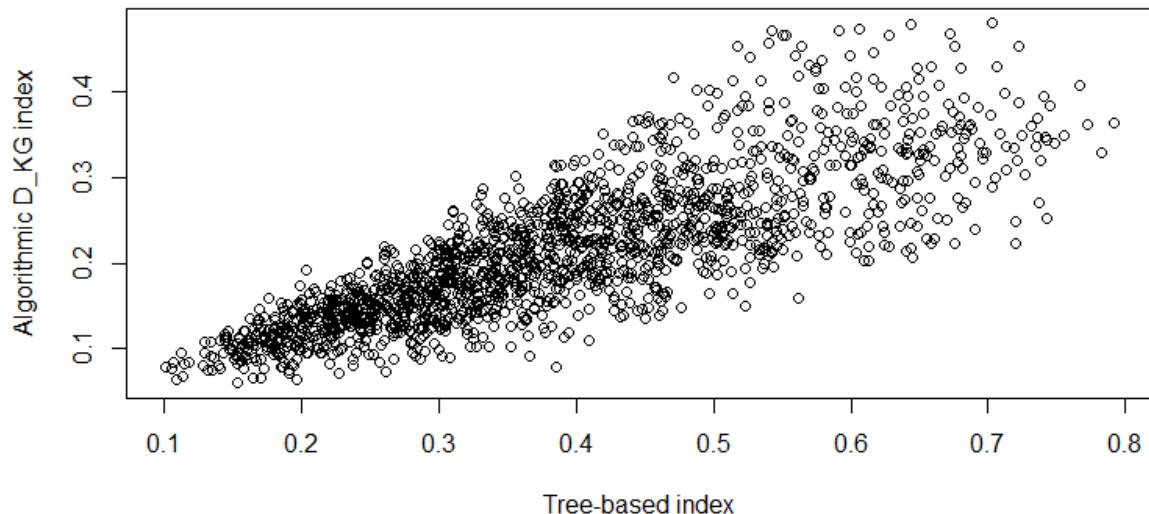
plot(vbetaU_A, vUp, xlab="New algorithmic beta uniqueness", ylab ="Tree-based beta uniqueness")
```



These beta uniqueness indices rely on plot-to-plot dissimilarity indices. Below, we compare the plot-to-plot dissimilarity indices: the tree-based index (Ricotta et al. 2020, function `DP` in package `adiv`) and Kosman (1996) & Gregorius et al. (2003) algorithmic index (D_{KG} in the main text; calculated above and stored in the list named `frameDKG`).

```
disKG <- as.dist(frameDKG$DKG)
disTree <- DP(H, RutorGlacier$Abund)

plot(as.vector(disTree), as.vector(disKG), xlab = "Tree-based index",
ylab="Algorithmic D_KG index")
```



References

- Berkelaar, M. et al. (2020) lpSolve: Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs. R package version 5.6.15. <https://CRAN.R-project.org/package=lpSolve>
- Gregorius, H.-R., Gillet, E.M., Ziehe, M. (2003) Measuring differences of trait distributions between populations. *Biometrical Journal* 8: 959–973.
- Kosman, E. (1996) Difference and diversity of plant pathogen populations: a new approach for measuring. *Phytopathology* 86: 1152–1155.
- Pavoine, S. (2020a) adiv: Analysis of Diversity. R package version 2.0. <https://CRAN.R-project.org/package=adiv>
- Pavoine, S. (2020b) adiv: an R package to analyse biodiversity in ecology. *Methods in Ecology and Evolution*. In press.
- Oksanen, J., Blanchet, F.G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., Minchin, P.R., O'Hara, R. B., Simpson, G.L., Solymos, P., Stevens, M.H.H., Szoecs, E., Wagner, H. (2019) vegan: Community Ecology Package. R package version 2.5-6. <https://CRAN.R-project.org/package=vegan>
- Ricotta, C., Laroche, F., Szeidl, L., Pavoine, S. (2020) From alpha to beta functional and phylogenetic redundancy. *Methods in Ecology and Evolution* 11: 487–493.