

Appendix 1. R scripts: manual and examples

Functions `spedisparam` and `fundisparam` available in Appendix 2 allow calculating Eq. 3, 6a, 6b, and 6c (function `spedisparam`) and 11a, 11b + the equivalents of 3 and 6c applied to species commonness (function `fundisparam`). The functions will be later integrated in package `adiv` of R (Pavoine 2020a,b).

Usage:

```
spedisparam(comm, method = c("D", "Delta", "L", "M"), abundance =  
c("relative", "absolute"), alpha = 2, tol = 1e-8, ...)
```

```
fundisparam(comm, dis, method = c("D", "Delta", "L", "M"), abundance  
= c("relative", "absolute"), alpha = 2, tol = 1e-8, ...)
```

Arguments:

`comm`: a matrix or a data frame of N plots \times S species containing the relative or absolute abundance of all species. Columns are species and plots are rows. Species must have names.

`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). *All species in `comm` must be included as row and column names in `dis`.*

`method`: a character string or a vector of character strings (if a vector is used only the first value is retained). Possible values are "D" for Eq. 11a of the main text, "Delta" for Eq. 11b of the main text, "L" for Eq. 6c (applied to species commonness rather than abundance in function `fundisparam`), and "M" for Eq. 3 (applied to species commonness rather than abundance in function `fundisparam`).

`abundance`: a character string or a vector of character strings (if a vector is used only the first value is retained). Possible values are "relative" for using relative abundances, and "absolute" for using absolute abundances.

`alpha`: a positive numeric for parameter alpha of Eq. 3, 6a,b,c, 11a,b of the main text.

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

Example of use:

Install and load package `adiv`.

```
install.packages("adiv") # Pavoine (2020a,b)
```

```
library(adiv)
```

Load in R functions `spedisparam` and `fundisparam` available in Appendix 2 (a .txt file). For that, use the following script to source the Appendix:

```
source(file.choose())
```

Load the dataset named `RutorGlacier`, available in package `adiv`.

```
data(RutorGlacier)
```

RutorGlacier\$Abund gives the abundance of plant species in plots and RutorGlacier\$Traits2 the values of traits for all observed plant species. In addition, RutorGlacier\$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).

The following commands provide the average abundance of each species in each successional stage:

```
FAC <- factor(RutorGlacier$Fac, levels = c("early","mid","late"))
MeanAbund <- apply(RutorGlacier$Abund, 2, function(x) tapply(x, FAC, mean))
```

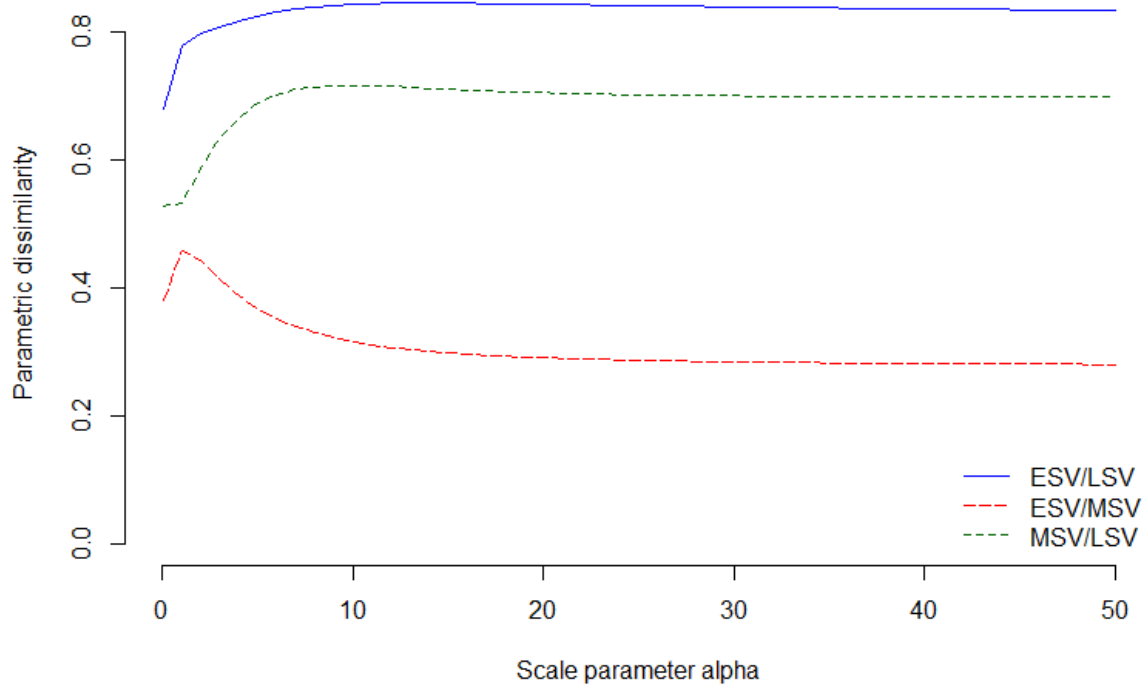
Pairwise species-based distances between successional stages were then obtained as follows:

```
# with Eq. 6a of the main text applied with relative abundances
D_EM <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("early","mid"), ], method="D", alpha = a)))

D_EL <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("early","late"), ], method="D", alpha = a)))

D_ML <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("mid","late"), ], method="D", alpha = a)))

plot(c(0.05,1:50), D_EM, ylim = c(0,max(c(D_EM, D_EL, D_ML))), col="red",
type="l", lty=5, xlab="Scale parameter alpha", ylab="Parametric
dissimilarity", bty="n")
lines(c(0.05,1:50), D_EL, col="blue")
lines(c(0.05,1:50), D_ML, col="darkgreen", lty=2)
legend("bottomright", col=c("blue","red","darkgreen"), lty=c(1,5,2),
legend=c("ESV/LSV", "ESV/MSV", "MSV/LSV"), bty="n")
```



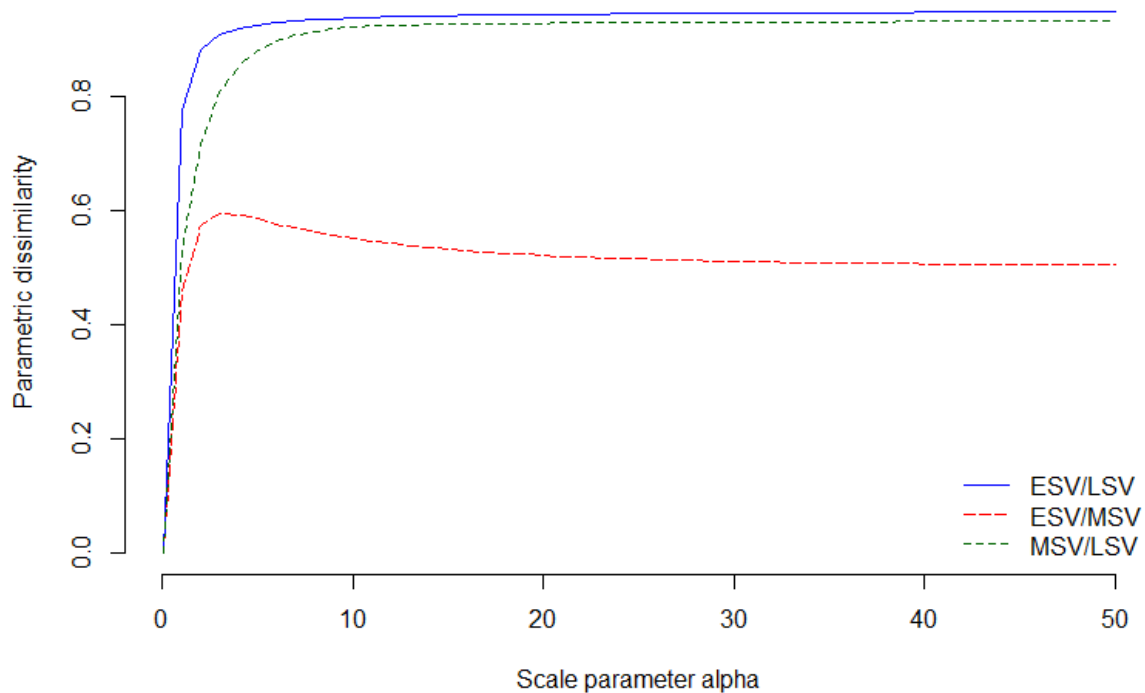
with Eq. 6b of the main text applied with relative abundances

```
Delta_EM <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("early","mid"), ], method="Delta", alpha = a)))
```

```
Delta_EL <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("early","late"), ], method="Delta", alpha = a)))
```

```
Delta_ML <- unlist(lapply(c(0.05,1:50), function(a) spedisparam(MeanAbund[
c("mid","late"), ], method="Delta", alpha = a)))
```

```
plot(c(0.05,1:50), Delta_EM, ylim = c(0,max(c(Delta_EM, Delta_EL,
Delta_ML))), col="red", type="l", lty=5, xlab="Scale parameter alpha",
ylab="Parametric dissimilarity", bty="n")
lines(c(0.05,1:50), Delta_EL, col="blue")
lines(c(0.05,1:50), Delta_ML, col="darkgreen", lty=2)
legend("bottomright", col=c("blue","red","darkgreen"), lty=c(1,5,2),
legend=c("ESV/LSV", "ESV/MSV", "MSV/LSV"), bty="n")
```



Like in Ricotta et al. (2021), the functional distances between species are calculated using the Euclidean distance applied to species' traits, each standardized to zero mean and unit standard deviation:

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

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

```
fdis <- fdis/max(fdis)
```

Pairwise functional distances between plots were then obtained as follows:

```
# with Eq. 11a of the main text
Dfun_EM <- unlist(lapply(c(1e-8,1:50), function(a) fundisparam(MeanAbund[
c("early","mid"), ], fdis, method="D", alpha = a)))

Dfun_EL <- unlist(lapply(c(1e-8,1:50), function(a) fundisparam(MeanAbund[
c("early","late"), ], fdis, method="D", alpha = a)))

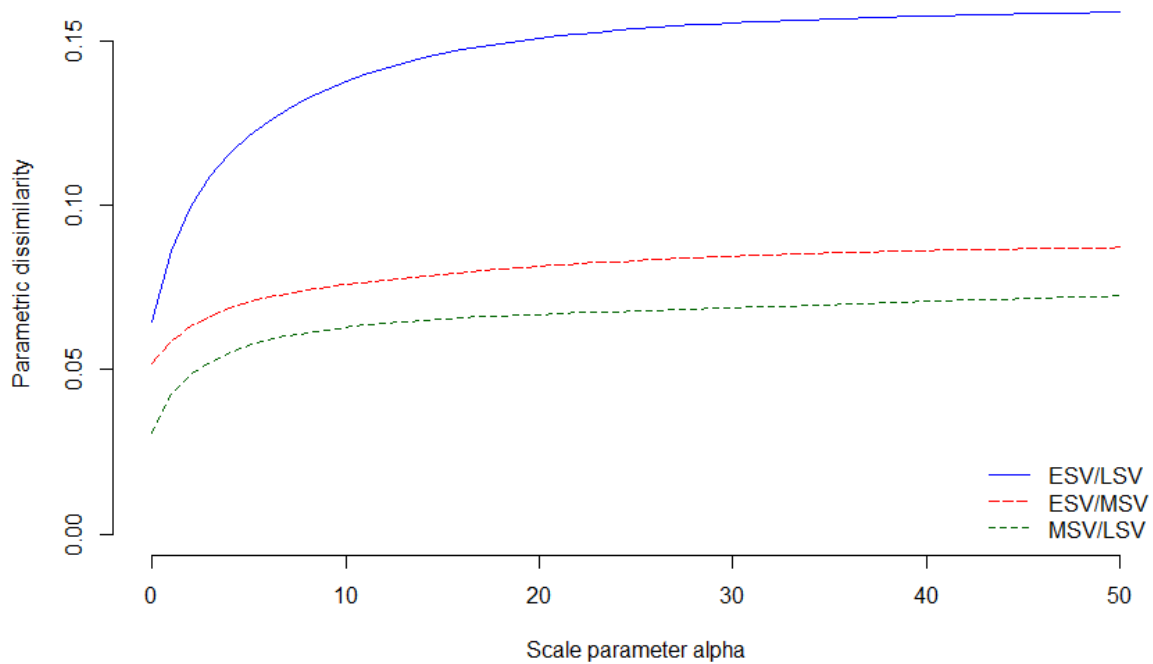
Dfun_ML <- unlist(lapply(c(1e-8,1:50), function(a) fundisparam(MeanAbund[
c("mid","late"), ], fdis, method="D", alpha = a)))

plot(c(1e-8,1:50), Dfun_EM, ylim = c(0,max(c(Dfun_EM, Dfun_EL, Dfun_ML))),
col="red", type="l", lty=5, xlab="Scale parameter alpha", ylab="Parametric
dissimilarity", bty="n")
lines(c(1e-8,1:50), Dfun_EL, col="blue")
lines(c(1e-8,1:50), Dfun_ML, col="darkgreen", lty=2)
```

```

legend("bottomright", col=c("blue","red","darkgreen"), lty=c(1,5,2),
legend=c("ESV/LSV", "ESV/MSV", "MSV/LSV"), bty="n")

```



```

# with Eq. 11b of the main text

```

```

Deltafun_EM <- unlist(lapply(c(1e-8,1:50), function(a)
fundisparam(MeanAbund[ c("early","mid"), ], fdis, method="Delta", alpha =
a)))

```

```

Deltafun_EL <- unlist(lapply(c(1e-8,1:50), function(a)
fundisparam(MeanAbund[ c("early","late"), ], fdis, method="Delta", alpha =
a)))

```

```

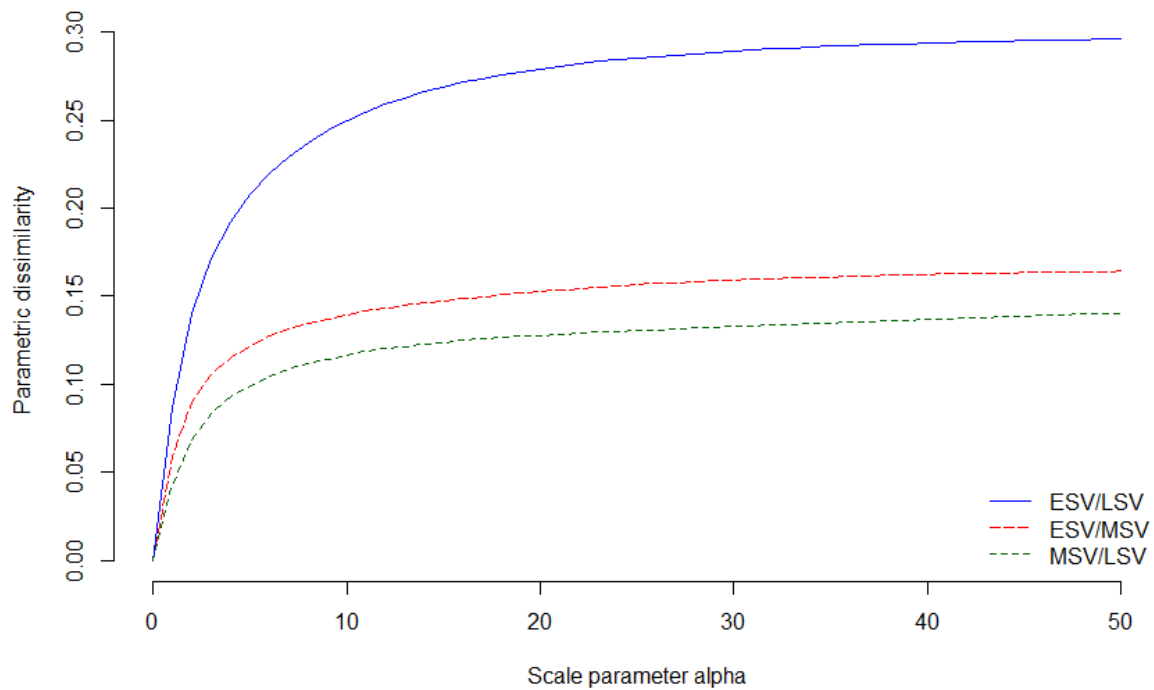
Deltafun_ML <- unlist(lapply(c(1e-8,1:50), function(a)
fundisparam(MeanAbund[ c("mid","late"), ], fdis, method="Delta", alpha =
a)))

```

```

plot(c(1e-8,1:50), Deltafun_EM, ylim = c(0,max(c(Deltafun_EM, Deltafun_EL,
Deltafun_ML))), col="red", type="l", lty=5, xlab="Scale parameter alpha",
ylab="Parametric dissimilarity", bty="n")
lines(c(1e-8,1:50), Deltafun_EL, col="blue")
lines(c(1e-8,1:50), Deltafun_ML, col="darkgreen", lty=2)
legend("bottomright", col=c("blue","red","darkgreen"), lty=c(1,5,2),
legend=c("ESV/LSV", "ESV/MSV", "MSV/LSV"), bty="n")

```



References

- 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* 11: 1106–1112. <https://doi.org/10.1111/2041-210X.13430>
- Ricotta, C., Kosman, E., Laroche, F., Pavoine, S. (2021) Beta redundancy for functional ecology. *Methods in Ecology and Evolution* 12: 1062–1069. <https://doi.org/10.1111/2041-210X.13587>