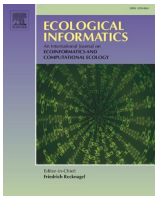
Ecological Informatics 66 (2021) 101458

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



On two dissimilarity-based measures of functional beta diversity

Carlo Ricotta [a,*], Evsey Kosman [b], Marco Caccianiga [c], Bruno E.L. Cerabolini [d], Sandrine Pavoine [e]

[a] Department of Environmental Biology, University of Rome 'La Sapienza', Rome, Italy
[b] Institute for Cereal Crops Improvement, Tel Aviv University, Tel Aviv, Israel
[c] Department of Biosciences, University of Milano, Milano, Italy
[d] Department of Biotechnology and Life Sciences, University of Insubria, Varese, Italy
[e] Centre d'Ecologie et des Sciences de la Conservation (CESCO), Muséum National d'Histoire Naturelle, CNRS, Sorbonne Université, Paris, France

ARTICLE INFO

Keywords:
Algorithmic measure
Assignment problem
Functional dissimilarity
Optimization process
Randomization test

ABSTRACT

In this paper, we propose two related versions of a dissimilarity-based measure of functional beta diversity, together with the associated tests for differences in beta diversity among different groups of samples. Both measures are based on the optimal functional matching between the species in two samples. As such, they are tightly connected to Hurlbert's seminal work on encounter-based diversity measures. The behavior of the proposed measures is illustrated with one worked example on the functional turnover of Alpine species along a successional gradient. Results show that both measures proved able to detect the functional turnover of vegetation along the chronosequence. The method, for which we provide a simple R function, further allows to evaluate the functional contribution of single sampling units to the overall beta diversity of any kind of species assemblages.

1. Introduction

Beta diversity measures the variability in species composition among a set of sampling units and is considered to be a key signature of the ecological processes that make species assemblages more or less similar to one another (Anderson et al., 2011; Bennet and Gilbert, 2016). Since the pioneering work by Whittaker (1972), there have been intense discussions on how to measure beta diversity and how to test for differences in beta diversity among different groups of samples. For reviews, see e.g. Lande (1996), Koleff et al. (2003), Anderson (2006); Anderson et al. (2011), Jost (2007), Tuomisto (2010a, 2010b), Chase et al. (2011), Chao and Chiu (2016), Legendre and De Cáceres (2013), Ricotta (2017), Chao and Ricotta (2019) and references therein.

Irrespective of how beta diversity is measured, an important requisite for diversity measures is their ecological interpretability. According to the seminal paper of Stuart Hurlbert (1971), meaningful diversity indices should have a straightforward biological interpretation: "We therefore can muddle along with a plethora of indices, each supported by at least one person's intuition and a few recommended by fashion, or we can sharpen our thoughts and rephrase our questions in terms of biologically meaningful properties [...]" (Hurlbert, 1971 p. 579).

Among these properties, the probability of intra- and interspecific encounters is a variable of interest, as it is directly related to the potential ecological interactions among all individuals and species in the community (Hurlbert, 1971; Patil and Taillie, 1982). This encounter-based approach is even more important for functional diversity where, unlike for classical diversity measures, the species are not considered equally dissimilar from each other. In a sense, dealing with functional diversity measures, the potential amount of ecological interactions among different individuals is ideally related to their functional resemblance.

In this paper, we thus propose two different versions of a dissimilarity-related index of functional beta diversity, together with the associated tests for differences among different groups of samples. Both indices are based on the optimal functional matching between the species in two samples. As such, they are tightly connected to Hurlbert's encounter-based approach.

2. A dissimilarity-based index of functional beta diversity

Given a set of N samples, let p_{jk} be the relative abundance of species $j = 1, 2, \ldots, S$ in sample $k = 1, 2, \ldots, N$ such that $0 \leq p_{jk} \leq 1$ and $\sum_j^S p_{jk} = 1$.

* Corresponding author.
E-mail address: carlo.ricotta@uniroma1.it (C. Ricotta).

https://doi.org/10.1016/j.ecoinf.2021.101458
Received 14 June 2021; Received in revised form 16 September 2021; Accepted 4 October 2021
Available online 13 October 2021

The information on the species functional organization within samples is usually represented by a symmetric $S \times S$ matrix of pairwise functional dissimilarities d_{ij} between species i and j in the range [0, 1] (with $d_{ij} = d_{ji}$ and $d_{ii} = 0$) which represent the multivariate differences in the character states among the S species.

To calculate a dissimilarity-based index of functional beta diversity, the first step consists in calculating the pairwise functional dissimilarity D_{hk} between any pair of samples h and k. To this end, Ricotta et al. (2021) first used an algorithmic measure originally developed by Kosman (1996) and Gregorius et al. (2003) to calculate genetic distances between populations. The measure is based on the optimal matching between the species abundances in h and k so as to minimize the overall functional dissimilarity between both samples.

The dissimilarity index D_{hk} is calculated as follows: given two samples h and k, with n individuals in both samples, each individual in h is matched to an individual in k in order to get n pairs that minimize the sum of functional dissimilarities between the individuals in each pair (Kosman and Leonard, 2007). The pairs are built such that all individuals in both samples are used only once. The overall functional dissimilarity between the two samples is then obtained as the mean dissimilarity between each pair of individuals (i.e. by dividing the sum of functional dissimilarities by the n pairs of individuals). However, since the number of individuals in h and k is generally not the same, to get a complete matching between the samples, this procedure is usually performed on the species relative abundances in both samples. The algorithmic dissimilarity D_{hk} can be thus interpreted as the minimum cost per individual needed to change the character states of the species in sample h to the states of the species in k (Gregorius et al., 2003).

Finding the optimal matching between the species abundances in h and k is known as the assignment problem, a special type of linear programming or linear optimization problem (Dantzig and Thapa, 1997). Dealing with species relative abundances, the functional dissimilarity between samples h and k can be formulated as (Gregorius et al., 2003):

$$D_{hk} = \min_{\pi} \sum_{i}^{S} \sum_{j}^{S} d_{ij} \times \pi(i,j) \tag{1}$$

where $\pi(i,j)$ is the relative abundance of species i in sample h that is matched with species j in sample k. Since D_{hk} is essentially a mean dissimilarity between matched pairs of individuals, if the functional dissimilarity d_{ij} between each pair of individuals is in the range [0, 1], the resulting mean dissimilarity also ranges between 0 and 1. Kosman (2014) further showed that if all species in h and k are considered maximally dissimilar from each other (i.e. if $d_{ij} = 1$ for all species i in sample h and species j in sample k), D_{hk} will be equal to $D_{hk} = \frac{1}{2}\sum_{i,j}^{S} \left| p_{ih} - p_{jk} \right|$.

A simple way to generalize D_{hk} to more than two samples, which is usually adopted in community ecology for calculating the beta diversity of a set of N samples (but see e.g. Diserud and Ødegaard, 2007), consists in calculating the mean value of D_{hk} for all possible pairs of samples:

$$\beta_N = \frac{\sum_{k>h}^{N} D_{hk}}{N(N-1)/2} \tag{2}$$

Once beta diversity has been calculated, the next step is how to test for differences in beta diversity among different groups of samples. To this end, Anderson (2006) proposed a multivariate analogue of Levene's (1960) test, which is directly connected to the way β_N is calculated. The test can be considered in two steps: first, starting from the functional dissimilarities between all pairs of sampling units D_{kn}, the dissimilarity $D_{k\cdot}$ of each individual sample from its group centroid in multivariate space is calculated according to McArdle and Anderson (2001). Next, the average of these dissimilarities among groups is compared using ANOVA. A P-value can be then obtained with either the traditional tables on F-distribution or by using a permutation procedure (Anderson, 2006).

A drawback of this method is that the dissimilarity of individual samples from the group centroid depends on the number of samples in each group. Take for example a group composed of five maximally dissimilar samples, i.e. with $D_{hk} = 1$ for all $h \neq k$. In this case, the dissimilarity $D_{k\cdot}$ of each individual sample from its group centroid is equal to $D_{k\cdot} = 0.632$. By contrast, for ten maximally dissimilar samples, $D_{k\cdot} = 0.671$ (for details, see Anderson, 2006). Accordingly, this test works correctly only with fully balanced designs with the same number of samples in each group.

To overcome this problem, a possible solution may consist in substituting $D_{k\cdot}$ with the mean dissimilarity of each individual sample k from all other $N - 1$ samples in the same group:

$$D_{\bar{k}} = \frac{\sum_{h \neq k}^{N} D_{hk}}{N-1} \tag{3}$$

The same approach was used by Violle et al. (2017) and Kosman et al. (2019) to calculate the mean distance in trait space of a species to all other species in a community. The main advantage of $D_{\bar{k}}$ over $D_{k\cdot}$ is that $D_{\bar{k}}$ is not influenced by the number of samples in each group. Like for the Anderson (2006) test, the average of these dissimilarities among groups can be then compared using standard ANOVA (see the example in Supplementary material, Appendix 1).

3. A second index of beta diversity

A second method for deriving a measure of multiple-site functional dissimilarity among sampling units may consist in calculating the dissimilarity of Kosman (1996) and Gregorius et al. (2003) $D_{k\eta}$ between the species relative abundances in sample k and the species relative abundances in an hypothetical complementary sample η. This complementary sample is obtained by pooling together the species relative abundances of all $N - 1$ samples that are different from k such that the relative abundance of species j in η is calculated as:

$$p_{j\eta} = \frac{\sum_{h \neq k}^{N} p_{jh}}{N-1} \tag{4}$$

According to this leave-one-out approach, η can be interpreted as the compositional centroid of the $N - 1$ samples that differ from k in Euclidean space (see Champely and Chessel, 2002). A multiple-site measure of beta diversity can be then obtained by taking the mean of the dissimilarities $D_{k\eta}$ over the N samples:

$$\beta_{\eta} = \frac{\sum_{k}^{N} D_{k\eta}}{N} \tag{5}$$

If beta diversity is calculated according to Eq. 5, a test for differences in beta diversity among different groups of samples can be then performed in the usual way, by comparing the mean values of $D_{k\eta}$ within each group with ANOVA.

4. Worked example

4.1. Data

To illustrate the behavior of the proposed measures, we used a data set of Alpine vegetation sampled by Caccianiga et al. (2006) along a primary succession at the foreland of the Rutor Glacier (Northern Italy). The data set has been already used in previous studies on community structure and diversity (Ricotta et al., 2016; Ricotta et al., 2020) and is composed of 45 species in 59 plots of approximately 25 m^2. All data are available in Ricotta et al. (2016, Appendix S2). The species abundances in each plot were measured with a five-point ordinal scale transformed to ranks. The plots were classified into three successional stages based on the age of the glacial deposits: early-successional stage (17 plots), mid-successional stage (32 plots), and late-successional stage (10 plots).

For all 45 species sampled at the three successional stages, we used

six quantitative traits that are related to their successional status along the primary succession: canopy height (CH; mm), leaf dry mass content (LDMC; %), leaf dry weight (LDW; mg), specific leaf area (SLA; $mm^2 \times mg^{-1}$), leaf nitrogen content (LNC; %), and leaf carbon content (LCC; %). All traits can be found in Caccianiga et al. (2006, Table 2).

First, we used the Euclidean distance to compute a matrix of pairwise functional distances between the 45 species from the six functional traits. For this purpose, all trait values for the 45 species were standardized to zero mean and unit standard deviation. The output functional distances were then scaled in the range $[0, 1]$ by dividing each distance by the maximum value in the distance matrix.

Using the algorithmic approach of Kosman (1996) and Gregorius et al. (2003), we next calculated the beta diversity components (i.e. dissimilarities) $D_{\bar{k}}$ and $D_{k\eta}$ for each sample in each successional stage. All calculations were performed with a new R script (available in the electronic Supplementary material, Appendix 1 and 2 of this paper) that modifies the R function `dislptransport` in Ricotta et al. (2021, Appendix S3). We finally tested for differences in beta diversity among the three successional stages by comparing the average of these dissimilarities among groups with ANOVA. *P*-values were obtained by using a permutation procedure. Among the many available permutation procedures in ANOVA designs (Anderson, 2004; Anderson and Ter Braak, 2003), we used the simplest approach, which consists in permuting individual observation units among the three successional stages of the Rutor chronosequence. To this end, we reshuffled $17 + 32 + 10 = 59$ observed dissimilarities $D_{\bar{k}}$ and $D_{k\eta}$ into random groups of 17, 32, and 10 units, respectively (9999 permutations) and recalculated the F-values for each permutation. The same permutation procedure, was then used to perform a post-hoc pairwise *t*-test with Holm correction of the values of $D_{\bar{k}}$ and $D_{k\eta}$ between the three successional stages.

4.2. Results

The results of the permutational ANOVA on the values of $D_{\bar{k}}$ and $D_{k\eta}$ among the three successional stages were in both cases highly significant ($F(D_{\bar{k}}) = 71.56$, $p < 0.001$ and $F(D_{k\eta}) = 18.91$, $p < 0.001$). For both dissimilarity coefficients $D_{\bar{k}}$ and $D_{k\eta}$, the within-group dispersion (or beta diversity) progressively decreased along the primary succession (Fig. 1). As shown by Caccianiga et al. (2006) and Ricotta et al. (2016), the significantly higher beta diversity of the early-successional samples may be due to the random dispersal mechanisms that drive the colonization of the moraine ridges in the first successional stages (abiotic filter). In contrast, the lower beta diversity of the mid- and late successional samples is associated to a lower level of stochasticity in the colonization process of the later successional stages and hence to an increased level of functional homogeneity among different sampling units (biotic filter).

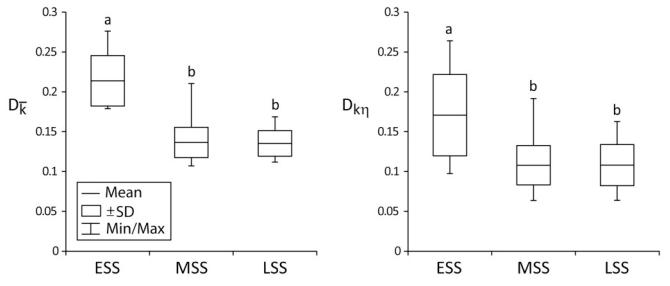


Fig. 1. Box plots of the beta diversity components (dissimilarity coefficients) $D_{\bar{k}}$ and $D_{k\eta}$ for the three successional stages of the Alpine vegetation of the Rutor glacier. ESS = early-successional stage; MSS = mid-successional stage; LSS = late-successional stage. Different letters a and b indicate significantly different distributions at $p < 0.001$ for $D_{\bar{k}}$ and $p < 0.01$ for $D_{k\eta}$ (permutational *t*-test with Holm adjustment for multiple tests based on 9999 randomizations).

Note that, since $D_{\bar{k}}$ is essentially an average dissimilarity between pairs of samples, while $D_{k\eta}$ is the dissimilarity between a given sample k and a complementary sample η that is obtained by pooling together the species relative abundances of all samples that are different from k, the values of $D_{k\eta}$ are generally lower than the values of $D_{\bar{k}}$ (see Fig. 1).

5. Discussion

In this paper we proposed two measures of functional beta diversity, β_N and β_η which originate from Whittaker's (1972) suggestion that beta can be summarized from a dissimilarity coefficient between pairs of samples (see also Chao and Chiu, 2016). The proposed measures are tightly connected to each other to the point that both of them can be considered '*variazioni sul tema*' of the same approach. In particular, $D_{k\eta}$ represents the dissimilarity of sample k from the pooled set of species in the $N - 1$ samples that differ from k. Therefore, this index, together with the corresponding beta diversity β_η, is directly related to the notion of originality (or distinctiveness, Pavoine et al., 2017). A sample is functionally original if its functional characteristics are rare in the pooled set of samples. The index is also related to the notion of the complementarity of a sample compared to a reference set of samples: complementarity being the gain in biodiversity units provided by adding an area (or sample) to a set of areas (samples) (Faith et al., 2004). These two notions (originality and complementarity) are used in conservation biology to identify sites with distinct species/functional/phylogenetic composition (and thus sites for which conservation actions should be a priority because of their distinct composition) (e.g. Mishler et al., 2014).

From the perspective of conservation biology, Kosman et al. (2019) recently proposed an additional indicator for estimating functional differences among samples: functional uniqueness, or singularity. Based on this approach, a sample that is on average quite distant from most samples but functionally similar to another sample has a lower conservation priority compared to a sample with the same average distance to other samples but without a close neighbor in functional space. To summarize this property, Violle et al. (2017) calculated the minimum pairwise distance between a focal sample and all other samples, while the singularity measure of Kosman et al. (2019) is based on variation in distances of the focal sample to all other samples, not just the nearest neighbor in trait space. Nonetheless, irrespective of how singularity is calculated, it can be easily derived from the distances D_{hk} in Eq. 1.

Unlike the vast majority of functional dissimilarity measures used in community ecology, the algorithmic index of Kosman (1996) and Gregorius et al. (2003), is not based on the excess of among-sample diversity compared to within-sample diversity (e.g. Chao et al., 2014; Chiu and Chao, 2014; Pavoine and Ricotta, 2014). Therefore, it is very flexible as it can be based on any between-species dissimilarity measure of choice without restrictions on their geometrical properties (see e.g. Pavoine and Ricotta, 2014). Also, the index of Kosman (1996) and Gregorius et al. (2003) satisfies an important requisite for functional dissimilarity measures which requires that dissimilarity remains unchanged if a given species j is replaced by two functionally identical species with the same total abundance of j. For mathematical details, see Leinster and Cobbold (2012); Pavoine and Ricotta (2019). From an ecological viewpoint, this means that the measures that conform to this requisite summarize the functional dissimilarity among samples irrespective of the identity of the species that support these functions. Accordingly, this algorithmic dissimilarity is closer to the essence of functional dissimilarity than the measures that do not conform to this requisite.

Regarding the test for differences in functional beta diversity among different groups of samples, the principle is the same as that of Anderson (2006). However, the values of $D_{\bar{k}}$ and $D_{k\eta}$ are not influenced by the number of samples in each group. In addition, we do not need to calculate the functional centroid of each group, and this renders the test much easier to perform, especially if the dissimilarities D_{hk} and $D_{k\eta}$ are not embeddable in Euclidean space without distortion (for details, see

McArdle and Anderson, 2001).

Nonetheless, alongside the pros, there are also a few potential cons for this test: like for the Anderson test, the values of $D_{\bar{k}}$ and $D_{k\eta}$ are not fully independent of each other. This is because, for a given sample k the quantity $D_{\bar{k}}$ ($D_{k\eta}$) is obtained by averaging all dissimilarities D_{hk} (all species relative abundances p_{jh}) over all $N-1$ samples that are different from k (see Eq. 3 and 4, respectively). This nonindependence may become relevant for small numbers of samples such that in the most critical situation of $N=2$, the values of $D_{\bar{k}}$ and $D_{k\eta}$ are identical for both samples.

Even more importantly, the randomization process associated to this test, while being statistically sound, has only little biological foundation. Beta diversity describes the spatial variability in species composition and is considered to be a key signature of a number of community assembly processes, such as dispersal, habitat filtering, intra- and interspecific competition, or the species responses to environmental conditions (Bennet and Gilbert, 2016). Therefore, while the permutation of the dissimilarities $D_{\bar{k}}$ and $D_{k\eta}$ among sampling units has no clear biological meaning, a biologically sound null model should provide some indication on whether differences in beta diversity among groups of samples are actually related to deterministic assembly processes that deviate from stochastic patterns of species co-occurrence (Chase et al., 2011). This may be achieved, for example, by restricted permutation of species occurrences among the samples in each group. However, to construct an adequate randomization test that correctly addresses the ecological questions under study without confounding within group heterogeneity with between group heterogeneity, some additional work is needed. In the meantime, the tests described in this paper may represent an acceptable, though ecologically imperfect solution to the problem.

Author contributions

CR: Conceptualization; Methodology; Data analysis; Writing - original draft. EK: Methodology; Data analysis; Writing - review & editing. MC: Data collection; Data analysis; Writing - review & editing. BC: Data collection; Data analysis; Writing - review & editing. SP: Methodology; Data analysis; Software; Writing - review & editing.

Funding

This research received no external funding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ecoinf.2021.101458.

References

Anderson, M.J., 2004. PERMDISP: A FORTRAN Computer Program for Permutational Analysis of Multivariate Dispersions (for any Two-Factor ANOVA Design) Using Permutation Tests. Department of Statistics, University of Auckland, New Zealand.
Anderson, M.J., 2006. Distance-based tests for homogeneity of multivariate dispersions. Biometrics 62, 245–253.
Anderson, M.J., Ter Braak, C.T.F., 2003. Permutation tests for multi-factorial analysis of variance. J. Stat. Comput. Simul. 73, 85–113.
Anderson, M.J., Crist, T.O., Chase, J.M., Vellend, M., Inouye, B.D., Freestone, A.L., Sanders, N.J., Cornell, H.V., Comita, L.S., Davies, K.F., Harrison, S.P., Kraft, N.J.B., Stegen, J.C., Swenson, N.G., 2011. Navigating the multiple meanings of beta diversity: a road map for the practicing ecologist. Ecol. Lett. 14, 19–28.
Bennet, J.R., Gilbert, B., 2016. Contrasting beta diversity among regions: how do classical and multivariate approaches compare? Glob. Ecol. Biogeogr. 25, 368–377.
Caccianiga, M., Luzzaro, A., Pierce, S., Ceriani, R.M., Cerabolini, B.E.L., 2006. The functional basis of a primary succession resolved by CSR classification. Oikos 112, 10–20.
Champely, S., Chessel, D., 2002. Measuring biological diversity using Euclidean metrics. Environ. Ecol. Stat. 9, 167–177.
Chao, A., Chiu, C.H., 2016. Bridging the variance and diversity decomposition approaches to beta diversity via similarity and differentiation measures. Methods Ecol. Evol. 7, 919–928.
Chao, A., Ricotta, C., 2019. Quantifying evenness and linking it to diversity, beta diversity, and similarity. Ecology 100, e02852.
Chao, A., Chiu, C.H., Jost, L., 2014. Unifying species diversity, phylogenetic diversity, functional diversity, and related similarity and differentiation measures through Hill numbers. Annu. Rev. Ecol. Evol. Syst. 45, 297–324.
Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M., Inouye, B.D., 2011. Using null models to disentangle variation in community dissimilarity from variation in α-diversity. Ecosphere 2, art24. https://doi.org/10.1890/ES10-00117.1.
Chiu, C.H., Chao, A., 2014. Distance-based functional diversity measures and their decomposition: a framework based on Hill numbers. PLoS One 9, e100014.
Dantzig, G.B., Thapa, M.N., 1997. Linear Programming. Springer, New York.
Diserud, O.H., Ødegaard, F., 2007. A multiple-site similarity measure. Biol. Lett. 3, 20–22.
Faith, D.P., Reid, C.A.M., Hunter, J., 2004. Integrating phylogenetic diversity, complementarity and endemism for conservation assessment. Conserv. Biol. 18, 255–261.
Gregorius, H.R., Gillet, E.M., Ziehe, M., 2003. Measuring differences of trait distributions between populations. Biom. J. 45, 959–973.
Hurlbert, S.H., 1971. The nonconcept of species diversity: a critique and alternative parameters. Ecology 52, 577–586.
Jost, L., 2007. Partitioning diversity into independent alpha and beta components. Ecology 88, 2427–2439.
Koleff, P., Gaston, K.J., Lennon, J.J., 2003. Measuring beta diversity for presence-absence data. J. Anim. Ecol. 72, 367–382.
Kosman, E., 1996. Difference and diversity of plant pathogen populations: a new approach for measuring. Phytopathology 86, 1152–1155.
Kosman, E., 2014. Measuring diversity: from individuals to populations. Eur. J. Plant Pathol. 138, 467–486.
Kosman, E., Leonard, K.J., 2007. Conceptual analysis of methods applied to assessment of diversity within and distance between populations with asexual or mixed mode of reproduction. New Phytol. 174, 683–696.
Kosman, E., Burgio, K.R., Presley, S.J., Willig, M.R., Scheiner, S.M., 2019. Conservation prioritization based on trait-based metrics illustrated with global parrot distributions. Divers. Distrib. 25, 1156–1165.
Lande, R., 1996. Statistics and partitioning of species diversity, and similarity among multiple communities. Oikos 76, 5–13.
Legendre, P., De Cáceres, M., 2013. Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. Ecol. Lett. 16, 951–963.
Leinster, T., Cobbold, C.A., 2012. Measuring diversity: the importance of species similarity. Ecology 93, 477–489.
Levene, H., 1960. Robust tests for equality of variances. In: Olkin, I., Ghurye, S.G., Hoeffding, W., Madow, W.G., Mann, H.B. (Eds.), Contributions to Probability and Statistics. Stanford University Press, Stanford, pp. 278–292.
McArdle, B.H., Anderson, M.J., 2001. Fitting multivariate models to community data: a comment on distance-based redundancy analysis. Ecology 82, 290–297.
Mishler, B.D., Knerr, N., González-Orozco, C.E., Thornhill, A.H., Laffan, S.W., Miller, J.T., 2014. Phylogenetic measures of biodiversity and neo-and paleo-endemism in Australian Acacia. Nat. Commun. 5, 1–10.
Patil, G.P., Taillie, C., 1982. Diversity as a concept and its measurement. J. Am. Stat. Assoc. 77, 548–561.
Pavoine, S., Ricotta, C., 2014. Functional and phylogenetic similarity among communities. Methods Ecol. Evol. 5, 666–675.
Pavoine, S., Ricotta, C., 2019. Measuring functional dissimilarity among plots: adapting old methods to new questions. Ecol. Indic. 97, 67–72.
Pavoine, S., Bonsall, M.B., Dupaix, A., Jacob, U., Ricotta, C., 2017. From phylogenetic to functional originality: guide through indices and new developments. Ecol. Indic. 82, 196–205.
Ricotta, C., 2017. Of beta diversity, variance, evenness, and dissimilarity. Ecol. Evol. 7, 4835–4843.
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 Ecol. Evol. 7, 1386–1395.
Ricotta, C., Acosta, A.T.R., Caccianiga, M., Cerabolini, B.E.L., Godefroid, S., Carboni, M., 2020. From abundance-based to functional-based indicator species. Ecol. Indic. 118, 106761.
Ricotta, C., Kosman, E., Laroche, F., Pavoine, S., 2021. Beta redundancy for functional ecology. Methods Ecol. Evol. https://doi.org/10.1111/2041-210X.13587.

Tuomisto, H., 2010a. A diversity of beta diversities: straightening up a concept gone awry. Part 1. Defining beta diversity as a function of alpha and gamma diversity. Ecography 33, 2–22.
Tuomisto, H., 2010b. A diversity of beta diversities: straightening up a concept gone awry. Part 2. Quantifying beta diversity and related phenomena. Ecography 33, 23–45.
Violle, C., Thuiller, W., Mouquet, N., Munoz, F., Kraft, N.J.B., Cadotte, M.W., Livingstone, S.W., Mouillot, D., 2017. Functional rarity: the ecology of outliers. Trends Ecol. Evol. 32, 356–367.
Whittaker, R., 1972. Evolution and measurement of species diversity. Taxon 21, 213–251.

Appendix 1. R scripts and examples

The R function `distorefpool` calculates, in a set of *N* plots, the functional dissimilarity between a given plot and a theoretical plot obtained by pooling all other plots (like in Eq. 4 and 5 of the main text). When the plots are distributed between defined groups, the R function `rtestsampledis` performs the ANOVA-like approach that evaluates how different the average dissimilarity between two plots of a group (see argument `fac`) is from one group to another. The function `posthoc` completes function `rtestsampledis` by performing pair-wise tests: an ANOVA-like test is performed to evaluates how different the average dissimilarity between two plots of a group is from the average dissimilarity between two plots of another group (all possible combinations of two groups are considered). 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 a new version of the adiv package of R: https://cran.r-project.org/web/packages/adiv/index.html. The functions were checked and applied with R-4.0.5.

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: ade4 (Thioulouse et al. 2018)

Function Syntax: Functions are available in Appendix 2 (.txt file)

Usage:
```
distorefpool(comm, dis, fundis, ...)
rtestsampledis(comm, dis, fac, fundis, option = 1:2, nrep = 999,
...)
posthoc(Xtest, p.adjust.method = "none", output = c("light",
"full"))
```

Arguments:
`comm`: a matrix or a data frame of *N* plots × *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***.
`fundis`: a character string giving the name of a function to calculate plot to plot dissimilarity. The first argument of this function must be a matrix or a data frame of plots × species containing the relative or absolute abundance of all species. The second argument of this function must be 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).
`...`: further arguments that can be passed to function fundis.
`fac`: a factor with the length equal to the number of rows in comm. The factor indicates which group each plot belongs to.
`option`: either 1 or 2. If 1, then the functional distinctiveness of a plot is measured as its average functional dissimilarity to another plot (like in Eq. 2 of the main text). If 2, then the functional distinctiveness of a plot is measured as its functional dissimilarity to a theoretical plot obtained by pooling all other plots (like in Eq. 5 of the main text).

nrep: the number of permutations to be done in the ANOVA-like approach that evaluates how different the average dissimilarity between two plots of a group (see argument fac) is from one group to another.
Xtest: an object of class "rtestsampledis" obtained by function rtestsampledis.
p.adjust.method: a character string. See function p.adjust for details and choice of adjustment methods.
output: a logical. If TRUE, all simulated values are saved and returned.

Value:

The function distorefpool returns a vector with, for each studied plot, the average functional dissimilarity between the plot and the other plots (Eq. 2 of the main text).
The function rtestsampledis returns an object of class "randtest" (from package ade4, Thioulouse et al. 2018).
The function posthoc returns an object of class "krandtest" (also from package ade4).

Example:

Install and load packages lpSolve, adiv, ade4 and ggplot2.

```
install.packages("lpSolve") # Berkelaar et al. (2020)
install.packages("adiv") # Pavoine (2020a,b)
install.packages("ade4") # Thioulouse et al. (2018)
install.packages("ggplot2") # Wickham (2016)

library(lpSolve)
library(adiv)
library(ade4)
library(ggplot2)
```

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

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

Load also function dislptransport available in Ricotta et al. (2021, Appendix S3). For that, copy and paste the function directly in your R console or copy and paste each of them in a .txt file and use the script source(file.choose()) to source it in R

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

```
data(RutorGlacier)
```

RutorGlacier is a list. RutorGlacier$Abund is an object of class data.frame that contains the abundance of plant species in plots (plots in rows; species in columns). RutorGlacier$Traits2 is an object of class data.frame that contains the values of traits for all observed plant species.

As 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 were 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 can be obtained by using the following command:

```
Dhk <- dislptransport(RutorGlacier$Abund, fdis) # Eq. 1 of the main text
```

The associated overall beta diversity over all plots simply is

```
BetaN <- mean(Dhk) # Eq. 2 of the main text
BetaN
[1] 0.2116677
```

The average dissimilarity of each individual plot to all other plots can be obtained as follows:

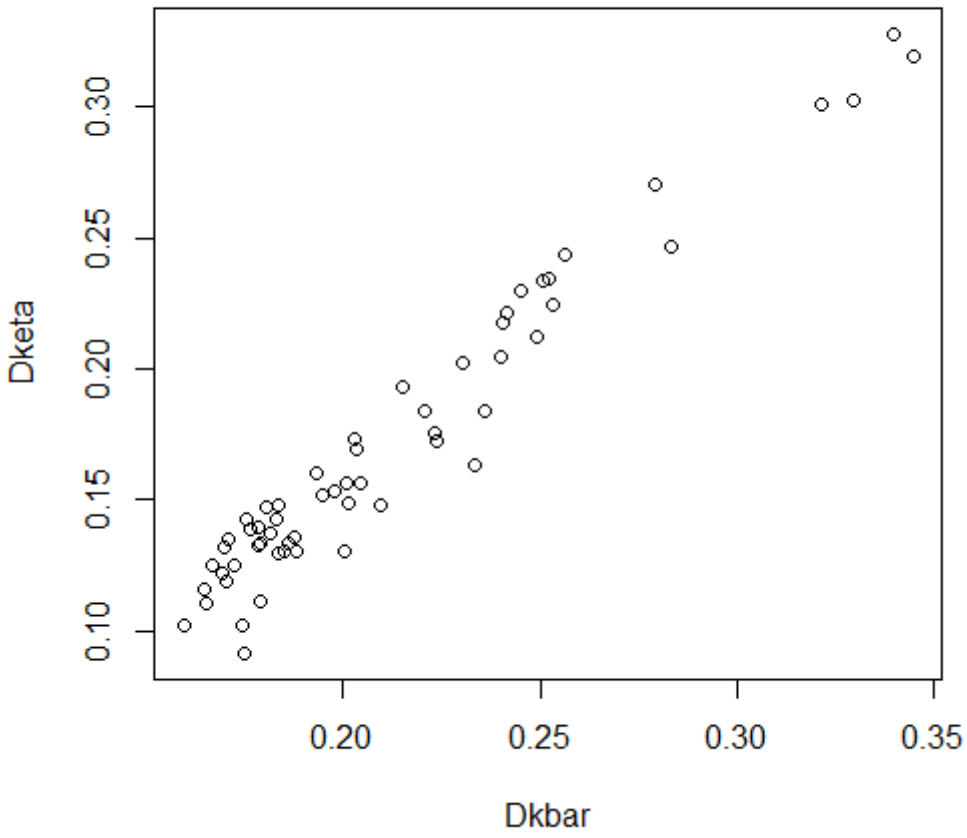
```
Dkbar <- sapply(1:attributes(Dhk)$Size, function(i) mean(as.matrix(Dhk)[i,
-i])) # Eq. 3 of the main text
names(Dkbar) <- attributes(Dhk)$Labels
Dkbar
       X1        X2        X3        X4        X5        X6        X7        X8
0.2833631 0.1880202 0.2097167 0.2007451 0.2332128 0.3294609 0.2238146 0.2014741
       X9       X10       X11       X12       X13       X14       X15       X16
0.3444822 0.2001922 0.2533755 0.2398871 0.2361672 0.2504668 0.3215201 0.3394210
      X17       X18       X19       X20       X21       X22       X23       X24
0.2490277 0.1813965 0.1875205 0.1754742 0.1976459 0.1667445 0.1695133 0.1838696
      X25       X26       X27       X28       X29       X30       X31       X32
0.1787682 0.1700641 0.1649776 0.1791913 0.1596490 0.1834967 0.1744900 0.1767666
      X33       X34       X35       X36       X37       X38       X39       X40
0.1655225 0.1863529 0.1790495 0.1805052 0.1725616 0.1831648 0.1749797 0.1933605
      X41       X42       X43       X44       X45       X46       X47       X48
0.1706312 0.2033942 0.1708969 0.1784742 0.1836892 0.1949550 0.2230722 0.2029705
      X49       X50       X51       X52       X53       X54       X55       X56
0.1852885 0.2417135 0.2404009 0.2521855 0.2150808 0.2044498 0.2792292 0.2207165
      X57       X58       X59
0.2303822 0.2561274 0.2453327
```

The average dissimilarity of each individual plot to the theoretical plot obtained by pooling all other plots can be obtained as follows:

```
Dketa <- distorefpool(RutorGlacier$Abund, fdis, "dislptransport")
Dketa
        X1         X2         X3         X4         X5         X6         X7
0.24676468 0.13054320 0.14790770 0.15652715 0.16337355 0.30240280 0.17244626
        X8         X9        X10        X11        X12        X13        X14
0.14835707 0.31946384 0.13039521 0.22406177 0.20454009 0.18382233 0.23347194
       X15        X16        X17        X18        X19        X20        X21
0.30088606 0.32799898 0.21218436 0.13750160 0.13538275 0.14228780 0.15342645
       X22        X23        X24        X25        X26        X27        X28
0.12467190 0.12195372 0.14763775 0.13931504 0.13184800 0.11590768 0.13326021
       X29        X30        X31        X32        X33        X34        X35
0.10180148 0.12946702 0.10236718 0.13893242 0.11049052 0.13377701 0.11115636
       X36        X37        X38        X39        X40        X41        X42
0.14724159 0.12469215 0.14273816 0.09118223 0.16022342 0.11851015 0.16944553
       X43        X44        X45        X46        X47        X48        X49
0.13484530 0.13268298 0.14774691 0.15206365 0.17579620 0.17311839 0.13041099
       X50        X51        X52        X53        X54        X55        X56
0.22158769 0.21760442 0.23463467 0.19284376 0.15629914 0.27028423 0.18411935
       X57        X58        X59
0.20210019 0.24369769 0.22946792
```

Link between Dketa and Dkbar:

```
plot(Dkbar, Dketa)
```



The overall beta diversity over all plots associated with Dketa simply is

```
BetaEta <- mean(Dketa) # Eq. 5 of the main text
BetaEta
[1] 0.1706723
```

`RutorGlacier$Fac` is a vector that 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).

```
FAC <- factor(RutorGlacier$Fac, levels = c("early","mid","late"))
```

The same calculation of the functional distinctiveness of a plot can thus also be done within each successional stage:

```
Listcomm <- split(1:nrow(RutorGlacier$Abund), FAC)

funx <- function(x) {
funi <- function(i) mean((as.matrix(Dhk)[x, x])[i, -i])
RES <- sapply(1:length(x), funi)
names(RES) <- rownames(RutorGlacier$Abund)[x]
```

```
return(RES)
}

LDkbar <- lapply(Listcomm, funx)
LDkbar
$early
       X1        X2        X3        X4        X5        X6        X7        X8
0.2793514 0.2063564 0.2146637 0.2145762 0.1902042 0.2452806 0.1980554 0.1963205
       X9       X10       X11       X12       X13       X14       X15       X16
0.2663816 0.1839177 0.2005795 0.1834980 0.1820301 0.2263759 0.2358982 0.2760978
      X17
0.1926447

$mid
      X18       X19       X20       X21       X22       X23       X24       X25
0.1421271 0.1504958 0.1306955 0.1609434 0.1237694 0.1233812 0.1472974 0.1387185
      X26       X27       X28       X29       X30       X31       X32       X33
0.1150523 0.1174132 0.1419706 0.1069922 0.1350532 0.1246678 0.1200482 0.1132186
      X34       X35       X36       X37       X38       X39       X40       X41
0.1368429 0.1490411 0.1279568 0.1240154 0.1311320 0.1355208 0.1443436 0.1225057
      X42       X43       X44       X45       X46       X47       X48       X49
0.1563800 0.1197213 0.1292627 0.1328761 0.1536369 0.2102809 0.1542701 0.1452713

$late
      X50       X51       X52       X53       X54       X55       X56       X57
0.1191709 0.1334624 0.1510494 0.1325934 0.1473606 0.1686762 0.1262438 0.1119260
      X58       X59
0.1395953 0.1234304

DkbarWithinGroup <- unlist(LDkbar, use.names = FALSE)
names(DkbarWithinGroup) <- unlist(lapply(LDkbar, names))
DkbarWithinGroup <- DkbarWithinGroup[rownames(RutorGlacier$Abund)]
DkbarWithinGroup
       X1        X2        X3        X4        X5        X6        X7        X8
0.2793514 0.2063564 0.2146637 0.2145762 0.1902042 0.2452806 0.1980554 0.1963205
       X9       X10       X11       X12       X13       X14       X15       X16
0.2663816 0.1839177 0.2005795 0.1834980 0.1820301 0.2263759 0.2358982 0.2760978
      X17       X18       X19       X20       X21       X22       X23       X24
0.1926447 0.1421271 0.1504958 0.1306955 0.1609434 0.1237694 0.1233812 0.1472974
      X25       X26       X27       X28       X29       X30       X31       X32
0.1387185 0.1150523 0.1174132 0.1419706 0.1069922 0.1350532 0.1246678 0.1200482
      X33       X34       X35       X36       X37       X38       X39       X40
0.1132186 0.1368429 0.1490411 0.1279568 0.1240154 0.1311320 0.1355208 0.1443436
      X41       X42       X43       X44       X45       X46       X47       X48
0.1225057 0.1563800 0.1197213 0.1292627 0.1328761 0.1536369 0.2102809 0.1542701
      X49       X50       X51       X52       X53       X54       X55       X56
0.1452713 0.1191709 0.1334624 0.1510494 0.1325934 0.1473606 0.1686762 0.1262438
      X57       X58       X59
0.1119260 0.1395953 0.1234304

Listab <- split(RutorGlacier$Abund, FAC)
LDketa     <-     lapply(Listab,     function(x)     distorefpool(x,     fdis,
"dislptransport"))
LDketa
$early
        X1         X2         X3         X4         X5         X6         X7
0.26392207 0.15402068 0.16524452 0.18909005 0.10639756 0.20661496 0.14685590
        X8         X9        X10        X11        X12        X13        X14
0.14954525 0.23729735 0.11137968 0.15482078 0.11607055 0.09729446 0.20217095
       X15        X16        X17
0.20733192 0.26076225 0.13230378

$mid
       X18        X19        X20        X21        X22        X23        X24
0.11767510 0.12624949 0.11017017 0.13823905 0.09834954 0.09531792 0.12972948
       X25        X26        X27        X28        X29        X30        X31
0.11711503 0.08477043 0.08440223 0.12207607 0.06362562 0.10044876 0.07859484
```

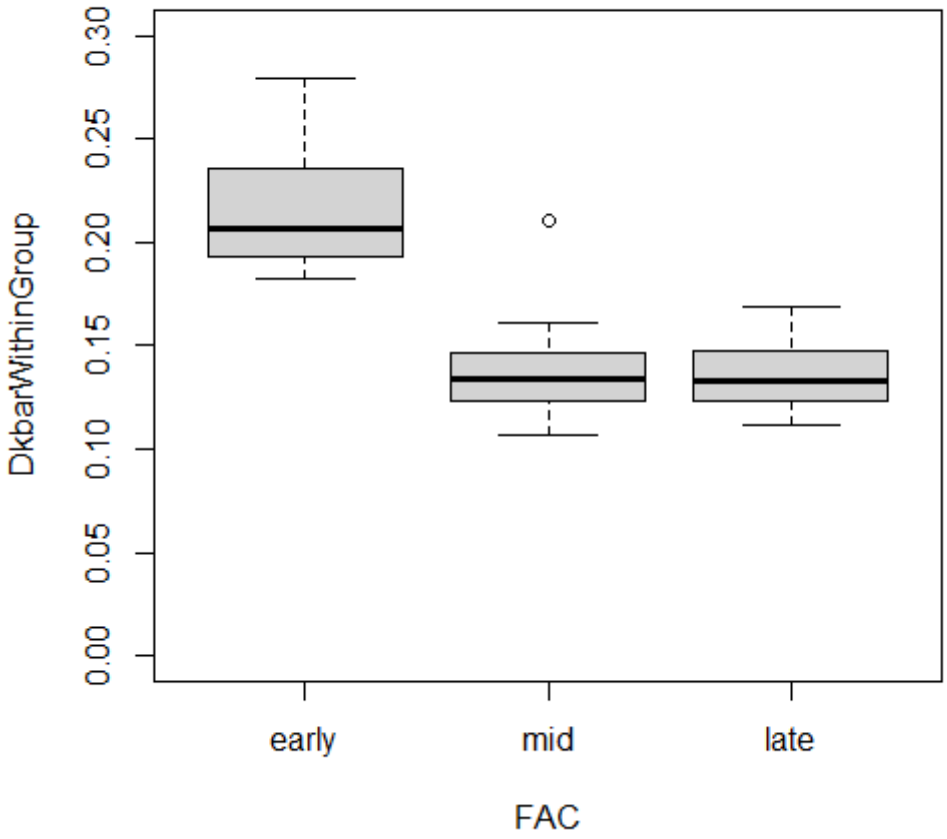
```
       X32        X33        X34        X35        X36        X37        X38
0.08973293 0.07011797 0.10849854 0.11154361 0.10510035 0.09342271 0.10286584
       X39        X40        X41        X42        X43        X44        X45
0.08521716 0.12458756 0.08391134 0.13600940 0.09119526 0.09812298 0.10750062
       X46        X47        X48        X49
0.13347024 0.19162813 0.13794545 0.11570316

$late
       X50        X51        X52        X53        X54        X55        X56
0.08910403 0.10888189 0.13266509 0.10950880 0.10701116 0.16268995 0.08632015
       X57        X58        X59
0.06387000 0.12289391 0.09773765
```

```
DketaWithinGroup <- unlist(LDketa, use.names = FALSE)
names(DketaWithinGroup) <- unlist(lapply(LDketa, names))
DketaWithinGroup <- DketaWithinGroup[rownames(RutorGlacier$Abund)]
DketaWithinGroup
```

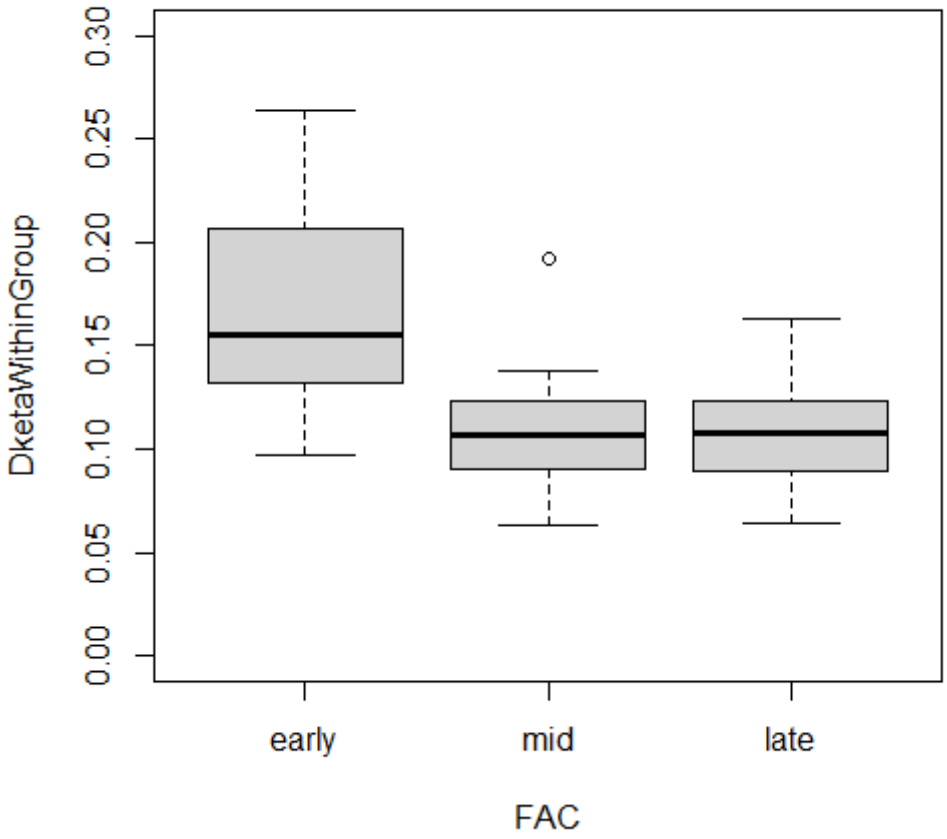
```
        X1         X2         X3         X4         X5         X6         X7
0.26392207 0.15402068 0.16524452 0.18909005 0.10639756 0.20661496 0.14685590
        X8         X9        X10        X11        X12        X13        X14
0.14954525 0.23729735 0.11137968 0.15482078 0.11607055 0.09729446 0.20217095
       X15        X16        X17        X18        X19        X20        X21
0.20733192 0.26076225 0.13230378 0.11767510 0.12624949 0.11017017 0.13823905
       X22        X23        X24        X25        X26        X27        X28
0.09834954 0.09531792 0.12972948 0.11711503 0.08477043 0.08440223 0.12207607
       X29        X30        X31        X32        X33        X34        X35
0.06362562 0.10044876 0.07859484 0.08973293 0.07011797 0.10849854 0.11154361
       X36        X37        X38        X39        X40        X41        X42
0.10510035 0.09342271 0.10286584 0.08521716 0.12458756 0.08391134 0.13600940
       X43        X44        X45        X46        X47        X48        X49
0.09119526 0.09812298 0.10750062 0.13347024 0.19162813 0.13794545 0.11570316
       X50        X51        X52        X53        X54        X55        X56
0.08910403 0.10888189 0.13266509 0.10950880 0.10701116 0.16268995 0.08632015
       X57        X58        X59
0.06387000 0.12289391 0.09773765
```

Box plots of the functional distinctiveness of plots (as defined by $D_{\bar{k}}$ and $D_{k\eta}$, respectively) within each successional stage:

```
boxplot(DkbarWithinGroup~FAC, ylim=c(0,0.3))
```



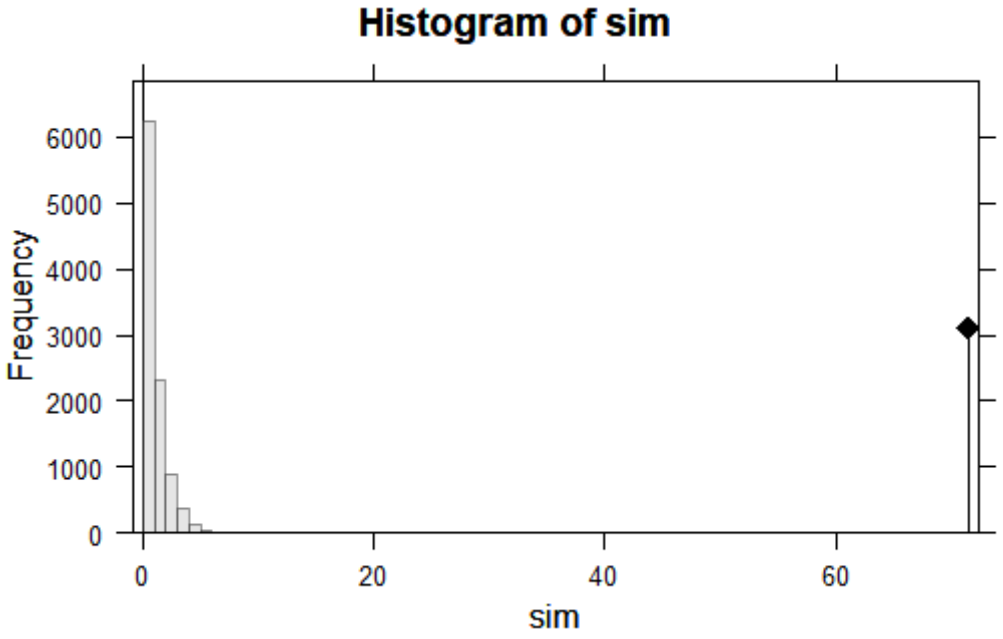

```
boxplot(DketaWithinGroup ~FAC, ylim=c(0,0.3))
```



ANOVA-like approach to evaluate how different the average dissimilarity between two plots of a group:

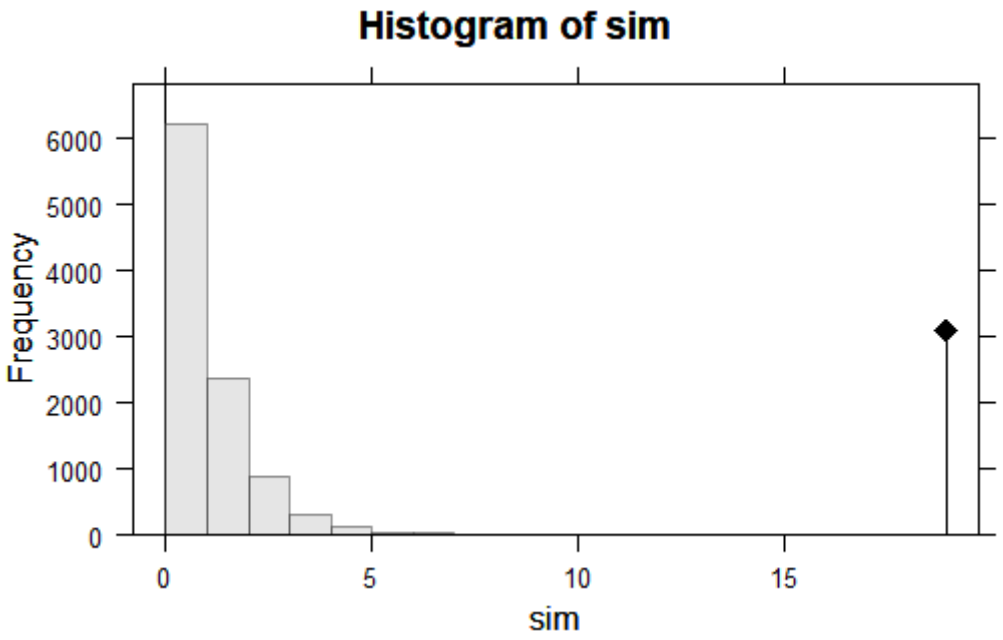
```
Rkbar <- rtestsampledis(RutorGlacier$Abund, fdis, FAC, "dislptransport",
option = 1, nrep = 9999) # Here with index D_k̄
Rkbar
Monte-Carlo test
Call: rtestsampledis(comm = RutorGlacier$Abund, dis = fdis, fac = FAC,
    fundis = "dislptransport", option = 1, nrep = 9999)

Observation: 71.56196 # F statistic

Based on 9999 replicates
Simulated p-value: 1e-04 # P-value
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
  65.043802    1.039703    1.175548

# Below the histogram of simulated values with an arrow indicating the
observed value of explained sum of squares divided by residual sum of
squares.
```

```
plot(Rkbar)
```



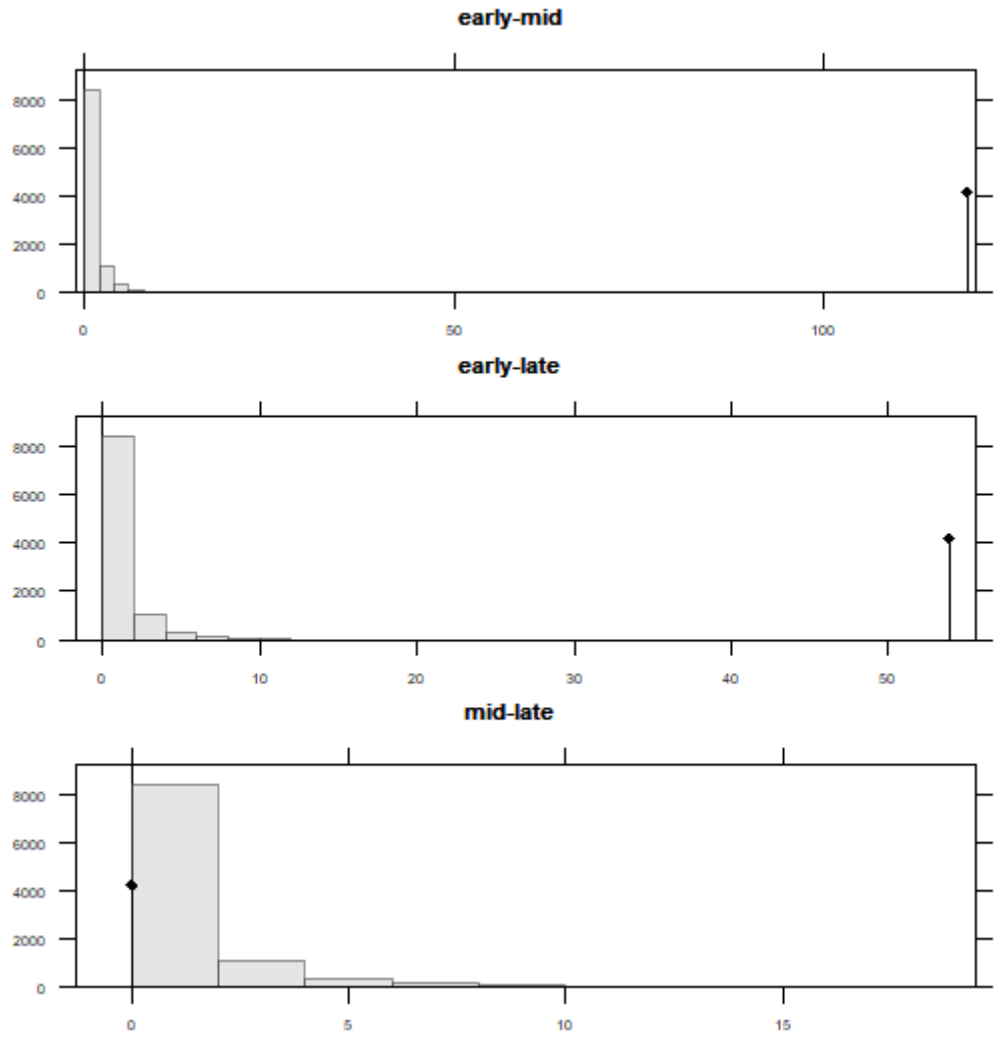

```
Rketa <- rtestsampledis(RutorGlacier$Abund, fdis, FAC, "dislptransport",
option = 2, nrep = 9999) # Here with index D_kη
```

```
Rketa
Monte-Carlo test
Call: rtestsampledis(comm = RutorGlacier$Abund, dis = fdis, fac = FAC,
    fundis = "dislptransport", option = 2, nrep = 9999)

Observation: 18.91216 # F statistic

Based on 9999 replicates
Simulated p-value: 1e-04 # P-value
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
  16.744722    1.033480    1.140026

# Below the histogram of simulated values with an arrow indicating the
observed value of explained sum of squares divided by residual sum of
squares.

plot(Rketa)
```

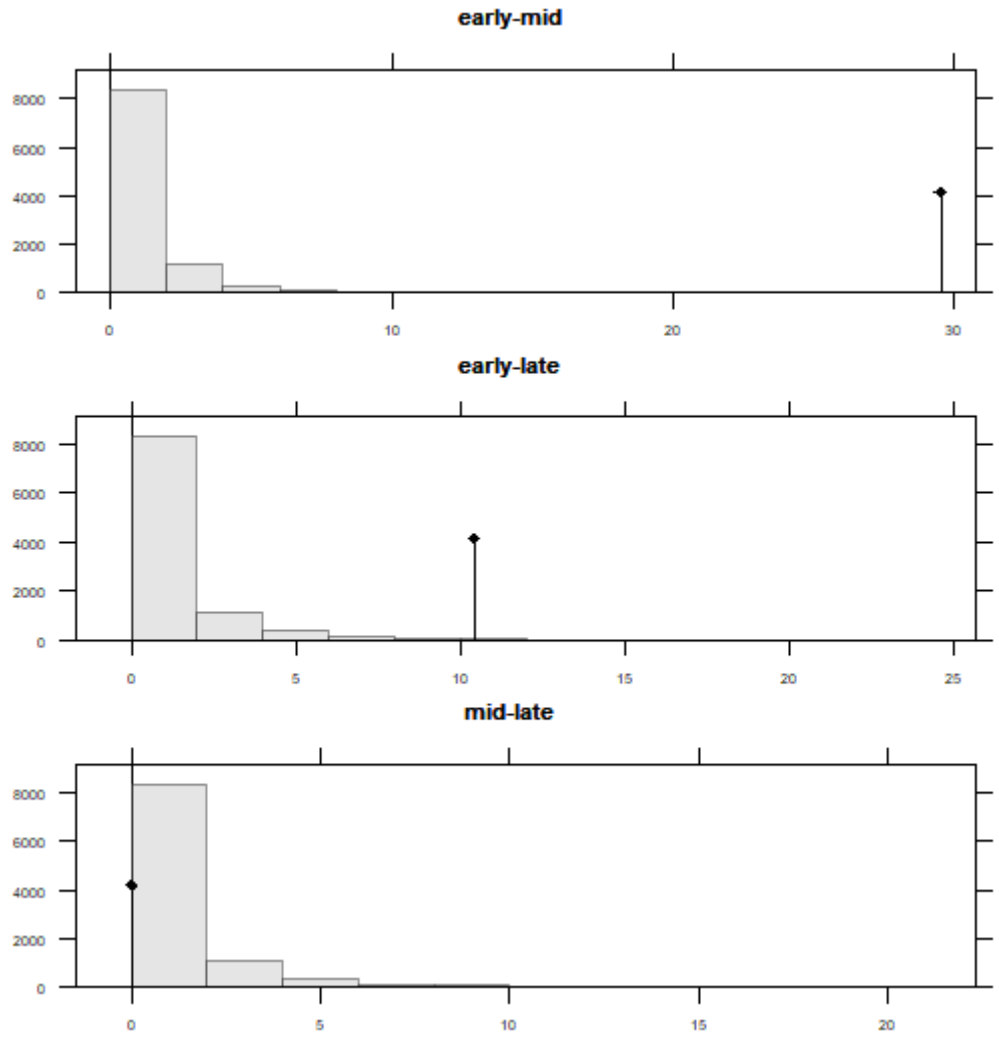


Below are the pair-wise tests: ANOVA-like tests performed to evaluates how different the average dissimilarity between two plots of a group is from the average dissimilarity between two plots of another group (all possible combinations of two groups are considered)

```
PHkbar <- posthoc(Rkbar, p.adjust.method = "holm")
PHkbar # Here with index D_k̄
class: krandtest lightkrandtest
Monte-Carlo tests
Call: posthoc(Xtest = Rkbar, p.adjust.method = "holm")

Number of tests:   3

Adjustment method for multiple comparisons:   holm
Permutation number:   9999
        Test          Obs    Std.Obs   Alter Pvalue Pvalue.adj
1  early-mid 119.34000057 78.1224147 greater 0.0001     0.0003
2 early-late  53.87378756 31.4392769 greater 0.0001     0.0003
3   mid-late   0.02383877 -0.6623418 greater 0.8805     0.8805

# Column Test indicates which successional stages are compared; Obs = F
statistic; Pvalue.adj = adjusted P-value

# Below, for each pair of compared succession stages, the histogram of
simulated values with an arrow indicating the observed F value.

plot(PHkbar)
```




```
PHketa <- posthoc(Rketa, p.adjust.method = "holm")
PHketa # Here with index D_kl
class: krandtest lightkrandtest
Monte-Carlo tests
Call: posthoc(Xtest = Rketa, p.adjust.method = "holm")

Number of tests:   3

Adjustment method for multiple comparisons:   holm
Permutation number:   9999
        Test         Obs   Std.Obs   Alter Pvalue Pvalue.adj
1  early-mid 29.54092333 19.531254 greater 0.0001     0.0003
2 early-late 10.45608502  5.507140 greater 0.0036     0.0072
3   mid-late  0.02462257 -0.666635 greater 0.8825     0.8825

# Column Test indicates which successional stages are compared; Obs = F
statistic; Pvalue.adj = adjusted P-value

# Below, for each pair of compared succession stages, the histogram of
simulated values with an arrow indicating the observed F value.

plot(PHketa)
```



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

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. https://doi.org/10.1111/2041-210X.13587

Thioulouse, J., Dray, S., Dufour, A.-B., Siberchicot, A., Jombart, J., Pavoine, S. (2018) Multivariate Analysis of Ecological Data with ade4. Springer, New York.

Wickham, H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer, New York.

```
distorefpool <- function(comm, dis, fundis, ...){

    if (!(inherits(comm, "data.frame") | inherits(comm, "matrix")))
        stop("comm must be a data.frame or a matrix")
    if(nrow(comm) < 2) stop("comm must have at least two rows")
    if (!(inherits(dis, "dist") | inherits(dis, "matrix")))
        stop("dis must be an object of class dist or a matrix")
    dis <- as.dist(dis)
    if (!inherits(fundis, "character"))
        stop("fundis must be a character string")
    if (!exists(fundis))
        stop("the function specified in argument fundis does not exist")
    thefundis <- match.fun(fundis)
    ncom <- nrow(comm)
    funi <- function(i){
        commrel <- sweep(comm, 1, rowSums(comm), "/")
        Veta <- colSums(commrel[-i, ])/(ncom-1)
        comi <- cbind.data.frame(unlist(comm[i, ]/sum(comm[i, ])), Veta)
        comi <- as.data.frame(t(comi))
        colnames(comi) <- colnames(comm)
        rownames(comi) <- c("k","eta")
        return(as.vector(thefundis(comi, dis, ...)))
    }
    RES <- sapply(1:ncom, funi)
    names(RES) <- rownames(comm)
    return(RES)

}

rtestsampledis <- function(comm, dis, fac, fundis, option = 1:2, nrep =
999, ...){

    if (!(inherits(comm, "data.frame") | inherits(comm, "matrix")))
        stop("comm must be a data.frame or a matrix")
    if (!(inherits(dis, "dist") | inherits(dis, "matrix")))
        stop("dis must be an object of class dist or a matrix")
    dis <- as.dist(dis)
    if(ncol(comm)!=attributes(dis)$Size) stop("Species in comm must be
similar and in the same order as species in dis")
    if(!inherits(fac, "factor") | length(fac)!=nrow(comm)) stop("Incorrect
definition of argument fac")
    ngroups <- length(levels(fac))
    if(min(table(fac))<2) stop("Each group in fac must contain at least
two communities")
    ncom <- nrow(comm)
    if (!inherits(fundis, "character"))
        stop("fundis must be a character string")
    if (!exists(fundis))
        stop("the function specified in argument fundis does not exist")
    thefundis <- match.fun(fundis)
    if(!option[1]%in%(1:2))  stop("Incorrect  definition  of  argument
option")
    if(option[1] == 1){
        Dhk <- as.matrix(thefundis(comm, dis, ...))
        LDhk <- list()
        for(i in 1:ngroups) {
           LDhk[[i]] <- Dhk[fac==levels(fac)[i], fac==levels(fac)[i]]
        }
        LDk  <-  lapply(LDhk,  function(x)  sapply(1:nrow(x),  function(j)
mean(as.matrix(x)[j, -j])))
```

```
    }
    else {
        LDk <- list()
        for(i in 1:ngroups) {
            LDk[[i]] <- distorefpool(comm[fac==levels(fac)[i], ], dis,
fundis, ...)
        }
    }
    EffRES <- unlist(lapply(LDk, length)) - 1
    RES <- sum(unlist(lapply(LDk, var)) * EffRES) / (ncom - ngroups)
    EffMOD <- unlist(lapply(LDk, length))
    MOD <- sum(EffMOD*(unlist(lapply(LDk, mean))-mean(unlist(LDk)))^2)
/(ngroups - 1)
    OBS <- MOD / RES
    simul <- function(i) {

        LL <- split(sample(unlist(LDk)), fac)
        EffRES <- unlist(lapply(LL, length)) - 1
        RES <- sum(unlist(lapply(LL, var)) * EffRES) / (ncom - ngroups)
        EffMOD <- unlist(lapply(LL, length))
        MOD <- sum(EffMOD*(unlist(lapply(LL, mean))-mean(unlist(LL)))^2)
/(ngroups - 1)
        return(MOD / RES)

    }
    SIM <- sapply(1:nrep, simul)
    RES <- as.randtest(obs = OBS, sim = SIM, alter = "greater", output =
"full")
    RES$call <- match.call()
    class(RES) <- c(class(RES), "rtestsampledis")
    return(RES)

}

posthoc <- function(Xtest, p.adjust.method = "none", output = c("light",
"full")){

    if(!inherits(Xtest, "rtestsampledis")) stop("X must be of class
rtestsampledis")
    objectsX <- as.list(Xtest$call)
    Comm <- eval.parent(objectsX$comm)
    Dis <- eval.parent(objectsX$dis)
    Fac <- eval.parent(objectsX$fac)
    Fundis <- eval.parent(objectsX$fundis)
    Option <- eval.parent(objectsX$option)
    Nrep <- eval.parent(objectsX$nrep)
    FFundis <- match.fun(Fundis)
    if(length(objectsX)>7) {
        FF <- formals(FFundis)
        objectsXdots <- objectsX[-(1:7)]
        for(i in 1:length(objectsXdots )){
        FF[names(objectsXdots)[i]] <- objectsXdots[[i]]
        }
        formals(FFundis) <- FF
    }
    levFac <- levels(Fac)
    setFac <- combn(levFac, 2)

    Distorefpool <- function(comm, dis) {
        ncom <- nrow(comm)
```

```
        funi <- function(i){
        Veta <- colSums(comm[-i, ])/(ncom-1)
        comi <- cbind.data.frame(unlist(comm[i, ]), Veta)
        comi <- as.data.frame(t(comi))
        colnames(comi) <- colnames(comm)
        rownames(comi) <- c("k","eta")
        return(as.vector(FFundis(comi, dis)))
        }
        RES <- sapply(1:ncom, funi)
        names(RES) <- rownames(comm)
        return(RES)
    }

    Rtestsampledis <- function(comm, dis, fac, option = 1:2, nrep = 999,
...) {

        dis <- as.dist(dis)
        ngroups <- length(levels(fac))
        ncom <- nrow(comm)

        if(option[1] == 1){
        Dhk <- as.matrix(FFundis(comm, dis))
        LDhk <- list()
        for(i in 1:ngroups) {
           LDhk[[i]] <- Dhk[fac==levels(fac)[i], fac==levels(fac)[i]]
        }
        LDk <- lapply(LDhk, function(x) sapply(1:nrow(x), function(j)
mean(as.matrix(x)[j, -j])))
        }
        else {
        LDk <- list()
        for(i in 1:ngroups) {
            LDk[[i]] <- Distorefpool(comm[fac==levels(fac)[i], ], dis)
        }
        }

        EffRES <- unlist(lapply(LDk, length)) - 1
        RES <- sum(unlist(lapply(LDk, var)) * EffRES) / (ncom - ngroups)
        EffMOD <- unlist(lapply(LDk, length))
        MOD <- sum(EffMOD*(unlist(lapply(LDk, mean))-mean(unlist(LDk)))^2)
/(ngroups - 1)
        OBS <- MOD / RES
        simul <- function(i) {

        LL <- split(sample(unlist(LDk)), fac)
        EffRES <- unlist(lapply(LL, length)) - 1
        RES <- sum(unlist(lapply(LL, var)) * EffRES) / (ncom - ngroups)
        EffMOD <- unlist(lapply(LL, length))
        MOD <- sum(EffMOD*(unlist(lapply(LL, mean))-mean(unlist(LL)))^2)
/(ngroups - 1)
        return(MOD / RES)

        }
        SIM <- sapply(1:nrep, simul)
        return(list(obs=OBS, sim=SIM))

    }

    FUNN <- function(Vi) {
        FAC <- factor(as.character(Fac)[as.character(Fac)%in%Vi])
```

```
        COMM <- Comm[as.character(Fac)%in%Vi, ]
        res <- Rtestsampledis(COMM, Dis, FAC, option = Option, nrep = Nrep)
        return(res)
    }
    RESlist <- apply(setFac, 2, FUNN)
    OBS <- unlist(lapply(RESlist, function(x) x$obs))
    SIM <- cbind.data.frame(lapply(RESlist, function(x) x$sim))
    colnames(SIM) <- apply(setFac, 2, function(x) paste(x, collapse="-"))
    RES <- as.krandtest(sim = SIM, obs = OBS,
        alter = "greater",
        p.adjust.method = p.adjust.method, output = output)
    RES$call <- match.call()
    return(RES)

}
```