

# Beta diversity computed as Var(Y)

## Description

Compute estimates of total beta diversity as the total variance in a community data matrix **Y**, as well as derived SCBD and LCBD statistics, for 20 dissimilarity coefficients or the raw data table. Computing beta diversity as Var(**Y**) for raw, untransformed community composition data is not recommended. Tests of significance of the LCDB indices are also produced.

## Usage

```
beta.div(Y,method="hellinger",sqrt.D=FALSE,samp=TRUE,nperm=999,save.D=FALSE)
```

## Arguments

|               |  |
|---------------|--|
| <b>Y</b>      | Community composition data. The file class can be either <code>data.frame</code> or <code>matrix</code> .  |
| <b>method</b> | One of the 20 dissimilarity coefficients, or "none". See Details. Names can be abbreviated to a non-ambiguous set of first letters. Default: <code>method="hellinger"</code> .   |
| <b>sqrt.D</b> | If <code>sqrt.D=TRUE</code> , the dissimilarities in matrix <b>D</b> are square-rooted before computation of <code>SStotal</code> , <code>BDtotal</code> and <code>LCBD</code> . This transformation may be useful for methods {"manhattan", "modmeanchardiff", "whittaker", "divergence", "canberra", "percentagedifference", "wishart"} since square-root transformation of the dissimilarities makes these <b>D</b> matrices Euclidean.<br><i>Note 1</i> – Euclideanarity is useful for ordination by principal coordinate analysis; lack of this property does not adversely affect <code>SStotal</code> , <code>BDtotal</code> and <code>LCBD</code> .<br><i>Note 2</i> – The logical value given to parameter <code>sqrt.D</code> has no incidence on calculations through methods {"euclidean", "profiles", "hellinger", "chord", "chisquare", "none"} since no <b>D</b> matrix is computed in those cases. |
| <b>samp</b>   | If <code>samp=TRUE</code> , the abundance-based distances ( <code>ab.jaccard</code> , <code>ab.sorensen</code> , <code>ab.ochiai</code> , <code>ab.simpson</code> ) are computed for sample data. If <code>samp=FALSE</code> , they are computed for true population data.   |
| <b>nperm</b>  | Number of permutations for the tests of significance of LCBD indices.  |
| <b>save.D</b> | If <code>save.D=TRUE</code> , the distance matrix will appear in the output list.  |
| <b>clock</b>  | If <code>clock=TRUE</code> , the computation time is printed. Useful when <code>nperm</code> is large.   |

## Details

Calculations may be carried out in two ways, depending on the selected method.

- For untransformed or transformed raw data, the total sum of squares (`SStotal`) is first computed, then the total beta diversity (`BDtotal`), which is `SStotal` divided by  $(n - 1)$ , is calculated. This algorithm is used for methods {"euclidean", "profiles", "hellinger", "chord", "chisquare", "none"}. No dissimilarity matrix nor transformation of the data is computed when the method is

"euclidean" or "none" (no transformation). For methods "profiles", "hellinger", "chord" and "chisquare", the algorithm begins with computation of the same-name transformation of the community data (Legendre and Gallagher 2001; Legendre and Legendre 2012, Section 7.7);  $SS_{total}$  and  $BD_{total}$  are then computed for the transformed data.

- Calculations can also be conducted from a dissimilarity matrix.  $SS_{total}$  is computed by summing the squared dissimilarities in the lower triangular dissimilarity matrix and dividing by  $n$ ; then, total beta diversity ( $BD_{total}$ ) is computed as above. Choices are: `method = {"manhattan", "modmeanchardiff", "whittaker", "divergence", "canberra", "percentagedifference", "wishart", "kulczynski", "ab.jaccard", "ab.sorensen", "ab.ochiai", "ab.simpson", "jaccard", "sorensen", "ochiai"}`. Equations for these dissimilarities are presented in Table 1 of Legendre and De Cáceres (subm.). See Chao et al. (2006) for details about the abundance-based (ab) coefficients.

The Jaccard, Sørensen and Ochiai coefficients are the binary forms of 9 of the 11 dissimilarity coefficients that are suitable for beta diversity assessment. The equivalences are described in Legendre and De Cáceres (submitted, Table 1). These popular coefficients can be computed directly using function `beta.div()` without going to the trouble of applying the quantitative forms of these coefficients to data reduced to presence-absence form. The transformation to presence-absence is done directly by function `dist.binary()` of package `ade4`, which is used by `beta.div()` to compute these coefficients. That function produces the dissimilarity matrix in the form  $\sqrt{D}$ , which is Euclidian. Hence for these three coefficients, function `beta.div()` should be used with option `sqrt.D=FALSE`.

(1) *Species contributions to beta diversity* (SCBD indices for the species) are computed for the untransformed or transformed raw data, but not for dissimilarity matrices. (2) *Local contributions to beta diversity* (LCBD indices) represent the degree of uniqueness of the sites in terms of their species compositions. They can be computed in all cases: raw (not recommended) or transformed data, as well as dissimilarity matrices. See Legendre and De Cáceres (submitted) for details.

LCBD indices are tested for significance by random, independent permutations within the columns of  $\mathbf{Y}$ . This permutation method tests  $H_0$  that the species are distributed at random, independently of one another, among the sites, while preserving the species abundance distributions in the observed data. See Legendre and De Cáceres (submitted) for discussion.

## Value

Function `beta.div` returns a list containing the following results:

|                              |   |
|------------------------------|---|
| <code>SStotal_BDtotal</code> | Total sum of squares and total beta diversity [= $\text{Var}(\mathbf{Y})$ ] of the data matrix. $BD_{total}$ statistics computed with the same $\mathbf{D}$ index are comparable among data sets having the same or different numbers of sampling units ( $n$ ), provided that they are of the same size or represent the same sampling effort. |
| <code>SCBD</code>            | Vector of <i>Species contributions to beta diversity</i> (SCBD), if computed.   |
| <code>LCBD</code>            | Vector of <i>Local contributions to beta diversity</i> (LCBD) for the sites.  |
| <code>p.LCBD</code>          | P-values associated with the LCBD indices.  |
| <code>method</code>          | Method selected.  |
| <code>note</code>            | Notes indicate whether the selected coefficient is Euclidean or not.  |
| <code>D</code>               | The distance matrix if <code>save.D=TRUE</code> .   |

## References

Chao, A., R. L. Chazdon, R. K. Colwell and T. J. Shen. 2006. Abundance-based similarity indices and their estimation when there are unseen species in samples. *Biometrics* 62: 361–371.

Legendre, P. and M. De Cáceres. Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. (Submitted).

Legendre, P. and E. D. Gallagher, E.D. 2001. Ecologically meaningful transformations for ordination of species data. *Oecologia* 129: 271–280.

Legendre, P. and Legendre, L. 2012. *Numerical Ecology*. 3rd English edition. Elsevier Science BV, Amsterdam.

## Author

Pierre Legendre, Département de sciences biologiques, Université de Montréal.  
License: GPL-2.

## Example

```
### Example: mite data available in the vegan package
```

```
require(vegan)
```

```
data(mite)
```

```
res = beta.div(mite, "hellinger", nperm=999)
```

```
# Plot a map of the LCDB indices
```

```
# First, load the file of Cartesian coordinates of the 70 mite sampling sites
```

```
data(mite.xy)
```

```
plot(mite.xy, asp=1, type="n", xlab="x coordinates (m)", ylab="y coordinates (m)", main="Map  
of mite LCBD")
```

```
points(mite.xy, pch=21, col="white", bg="brown", cex=120*res$LCBD)
```

```
### Example using the mite abundance data and the percentage difference dissimilarity
```

```
res = beta.div(mite, "percentage", nperm=999, clock=TRUE)
```

```
# Plot a map of the LCDB indices
```

```
# First, load the file of cartesian coordinates of the 70 mite sampling sites
```

```
data(mite.xy)
```

```
signif = which(res$p.LCBD <= 0.05)      # Which are the significant LCDB indices?
```

```
nonsignif = which(res$p.LCBD > 0.05)   # Which are the non-significant LCDB indices?
```

```
plot(mite.xy, asp=1, type="n", xlab="x coordinates (m)", ylab="y coordinates (m)", main="Map  
of mite LCBD (red = significant indices)")
```

```
points(mite.xy[nonsignif,], pch=21, col="white", bg="blue", cex=100*res$LCBD[nonsignif])
```

```
points(mite.xy[signif,], pch=21, col="white", bg="red", cex=100*res$LCBD[signif])
```