

Suppl. Doc. 3 – Analysis of simulated coenocline data with directional indices

The objective of this exercise is to illustrate the interpretation of the new directional indices through an example that would be familiar to ecologists. Most community ecologists are familiar with the representation of species abundances along a transect or an ecological gradient (including time) by a graph showing the succession of species along the gradient, like the one presented at the bottom of this page.

Note – Overlap is the similarity α . It will not be analysed in detail because it is not, as such, a directional index. Overlap is, however, included in the numerator of Nestedness indices.

Basic reference on coenoclines:

Whittaker, R. H. 1972. Evolution and measurement of species diversity. *Taxon* 21: 213–251.

Part 1 – Coenocline with increasing tolerances

Simulate a coenocline (matrix **Y**) *with increasing tolerances* (standard deviations): 51 sites, 51 species with identical maximum abundances of 20 (function argument *h*).

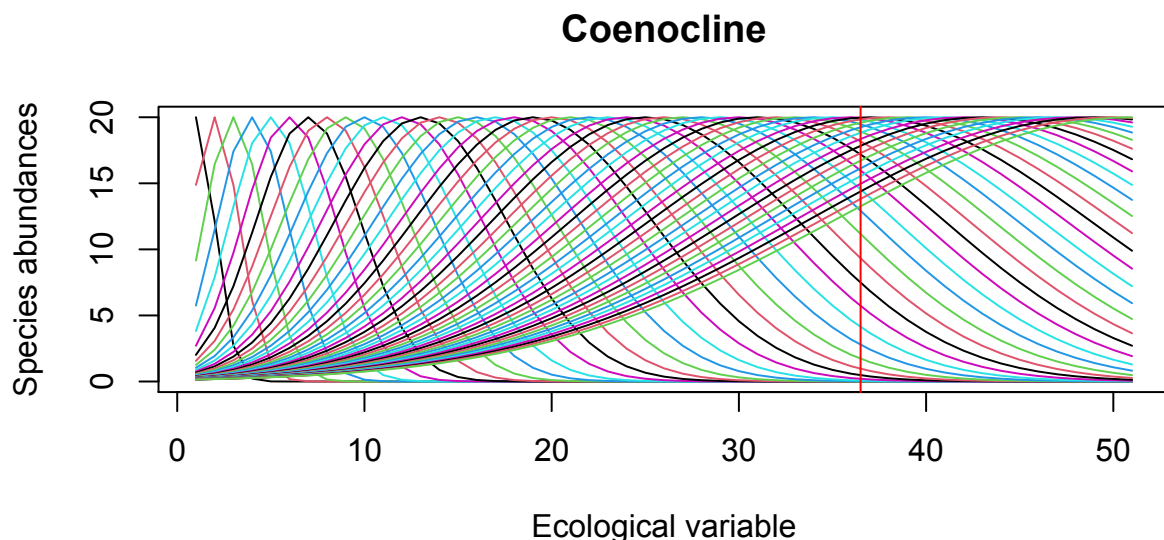
After cut-off at abundances >2, **Y** contains a large proportion of zeros in the first 36 sites

library(coenocliner)

```
# Coenocline with increasing tolerances. Y contains about 43% zeros: length(which(Y<2))
x <- seq(from=1, to=51, by=1)      # Coenocline variable or transect positions (graph abscissa)
opt <- seq(from=1, to=51, by=1)    # Positions of species optima along transect x
tol <- seq(1,16.2, by=0.3)          # Species tolerances, increasing along the species list
h <- rep(20, 51)                    # Maximum abundances of individual species (all equal)
```

```
Y <- coenocline(x, responseModel = "gaussian",
  params = cbind(opt = opt, tol = tol, h = h),
  countModel = "poisson", expectation = TRUE)
```

```
plot(Y, type = "l", lty = "solid", xlab="Ecological variable",
  ylab="Species abundances", main="Coenocline")
abline(v=36.5, col="red")          # Marks the cut-off point along the transect at abscissa=36.5
```



```

Y2.1 <- ifelse(Y > 2, 1, 0)      # Transform to binary (absence-presence) data
# Species richness at the 51 sites
apply(Y2.1, 1, sum)
[1] 7 9 12 14 16 19 20 23 25 27 30 31 34 36 38 41 42 41 41 40 40 39 38 38 37 37 36 35 35
[30] 34 34 33 32 32 31 31 30 29 29 28 27 27 26 26 25 24 24 23 23 22 21

```

```

# For illustration of calculation of the directional indices, we use a subset of sites (8 sites)
# well spaced-out along the coenocline, 5 sampling units apart
( sample.seq = seq(1, 40, by=5) )
[1] 1 6 11 16 21 26 31 36

```

```

Y2.2 = Y2.1[sample.seq,]      # Binary data at 8 spaced-out sites along the coenocline
apply(Y2.2, 1, sum)
[1] 7 19 30 41 40 37 34 31    # Species richness at the 8 selected sites

```

```

# Compute matrices with a, b and c to check the calculations on the following pages

```

```

a = Y2.2 %*% t(Y2.2)
b = (1 - Y2.2) %*% t(Y2.2)
c = Y2.2 %*% (1 - t(Y2.2))

```

b

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0	14	28	41	40	37	34	31
[2,]	2	0	14	28	30	30	30	30
[3,]	5	3	0	14	16	16	16	16
[4,]	7	6	3	0	2	2	2	2
[5,]	7	9	6	3	0	0	0	0
[6,]	7	12	9	6	3	0	0	0
[7,]	7	15	12	9	6	3	0	0
[8,]	7	18	15	12	9	6	3	0

c

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0	2	5	7	7	7	7	7
[2,]	14	0	3	6	9	12	15	18
[3,]	28	14	0	3	6	9	12	15
[4,]	41	28	14	0	3	6	9	12
[5,]	40	30	16	2	0	3	6	9
[6,]	37	30	16	2	0	0	3	6
[7,]	34	30	16	2	0	0	0	3
[8,]	31	30	16	2	0	0	0	0

```

# Sub-diagonal (.sd) vectors, from positions (2,1) to (8,7) in the square matrices

```

```

a.sd = c(5, 16, 27, 38, 37, 34, 31)
b.sd = c(2, 3, 3, 3, 3, 3, 3)
c.sd = c(14, 14, 14, 2, 0, 0, 0)

```

```

(b.sd+c.sd)      # c(16, 17, 17, 5, 3, 3, 3)  # Total turnover
2*pmin(b.sd,c.sd) # c(4, 6, 6, 4, 0, 0, 0)  # May be computed as (b.sd+c.sd) - abs(b.sd-c.sd)
(a.sd+b.sd+c.sd) # c(21, 33, 44, 43, 40, 37, 34)

```

Examine the partitioning of turnover into gain.t, loss.t and neutral.t

The vectors of sub-diagonal indices could be used to draw a **Figure** showing all results below

The following function requires function *directional.response.R* provided in *Suppl Doc 2*

Function *sub.diag.R*

```
sub.diag <- function(mat, method="gaining.turnover", relativize="J")
#
# Compute directional.response(mat); extract sub-diagonal of output matrices
{
x <- directional.response(mat, method=method, relativize=relativize)
n.sd <- nrow(x$mat.out)          # Number of values in sub-diagonal of mat.out
tmp <- cbind( 2:n.sd, 1:(n.sd-1) )
  # print(tmp)
s.diag <- x$mat.out[tmp]
if(is.na(x$total.t[[1]])) tt <- NA else tt <- x$total.t[tmp]
if(is.na(x$total.n[[1]])) tn <- NA else tn <- x$total.n[tmp]
if(is.na(x$total.strict.n[[1]])) tn2 <- NA else tn2 <- x$total.strict.n[tmp]
#
list(sub.diag=s.diag, total.t=tt, total.n=tn, total.strict.n=tn2)
}
```

Examine the partitioning of turnover into gain.t, loss.t and neutral.t

Turnover – The change in species composition between adjacent sites along a gradient.

```
( total.t = sub.diag(Y2.2, method= "gaining.turnover", relativize=NULL)$total.t )
[1] 16 17 17 5 3 3 3          # Total turnover = (b+c)
```

```
( gain.t = sub.diag(Y2.2, method= "gaining.turnover", relativize=NULL)$sub.diag )
# Method: gaining.turnover
[1] 16 17 17 4 0 0 0          # if(c>b then b+c, else 2*min(b,c))
```

```
( loss.t = sub.diag(Y2.2, method= "losing.turnover", relativize=NULL)$sub.diag )
# Method: losing.turnover
[1] 4 6 6 5 3 3 3          # if(c<b then b+c, else 2*min(b,c))
```

```
( neutral.t = sub.diag(Y2.2, method= "neutral.turnover", relativize=NULL)$sub.diag )
# Method: neutral.turnover      # 2*min(b,c)
[1] 4 6 6 4 0 0 0
```

=> Note: total.t = max(gain.t, loss.t); neutral.t = min(gain.t, loss.t)

The results show that total turnover is high between sites 1-2 (16 species), 2-3 (17 species) and 3-4 (17 species); then it drops to a lower level (5 and 3 species). Total turnover is dominated by gaining turnover between sites 1-2 (16 species), 2-3 (17 species) and 3-4 (17 species), and by losing turnover between sites 4-5 (5 species), 5-6, 6-7 and 7-8 (3 species).

Compute **standardized turnover indices**, controlling for total turnover.

=> It is easier to appreciate the ecological importance of individual turnover fractions by computing the **fraction** of total turnover attributed to the three indices.

gain.t / total.t

[1] 1.0 1.0 1.0 0.8 0.0 0.0 0.0

loss.t / total.t

[1] 0.2500000 0.3529412 0.3529412 1.0000000 1.0000000 1.0000000 1.0000000

neutral.t / total.t

[1] 0.2500000 0.3529412 0.3529412 0.8000000 0.0000000 0.0000000 0.0000000

When controlling for total turnover, the results clearly show that total turnover is dominated by gaining turnover between sites 1-2 (16 species), 2-3 (17 species) and 3-4 (17 species), and by losing turnover between sites 4-5 (5 species), 5-6, 6-7 and 7-8 (3 species), as predicted by the relationship $\text{total.t} = \max(\text{gain.t}, \text{loss.t})$ mentioned in the previous subsection.

Neutral turnover is the highest (80%) in relation to total turnover between sites 4-5. This means that the balance between gain and loss is the largest between sites 4-5 in relation to the total turnover, as predicted by the relationship $\text{neutral.t} = \min(\text{gain.t}, \text{loss.t})$ mentioned in the previous subsection.

=====

Examine the partitioning of nestedness into gain.n, loss.n and neutral.n

Nestedness – A type of richness difference pattern characterized by the species at a site being a strict subset of the species at a richer site.

```
( total.n = sub.diag(Y2.2, method= "gaining.nestedness", relativize=NULL)$total.n )
[1] 17 27 38 39 40 37 34
```

```
( gain.n = sub.diag(Y2.2, method= "gaining.nestedness", relativize=NULL)$sub.diag )
# Method: gaining.nestedness
[1] 17 27 38 38 37 34 31          # Maximum at (3,4) and (4,5)
```

```
( loss.n = sub.diag(Y2.2, method= "losing.nestedness", relativize=NULL)$sub.diag )
# Method: losing.nestedness
[1] 5 16 27 39 40 37 34          # Maximum at (5,6)
```

```
( neutral.n = sub.diag(Y2.2, method= "neutral.nestedness", relativize=NULL)$sub.diag )
# Method: neutral.nestedness
[1] 5 16 27 38 37 34 31          # Maximum at (4,5)
```

=> Note: total.n = max(gain.n, loss.n); neutral.n = min(gain.n, loss.n)

Compute **standardized nestedness indices**, controlling for total nestedness.

=> It is easier to appreciate the ecological importance of individual nestedness fractions by computing the **fraction** of total nestedness attributed to the three indices.

```
( gain.n / total.n )
[1] 1.0000000 1.0000000 1.0000000 0.9743590 0.9250000 0.9189189 0.9117647 # Decreasing
=> gain.n decreases along the series of pairs of sites.
```

```
( loss.n / total.n )
[1] 0.2941176 0.5925926 0.7105263 1.0000000 1.0000000 1.0000000 1.0000000 # Increasing
=> loss.n increases along the series of pairs of sites.
```

```
( neutral.n / total.n )
[1] 0.2941176 0.5925926 0.7105263 0.9743590 0.9250000 0.9189189 0.9117647 #Max at (4,5)
=> neutral.n increases along the first 4 pairs, then it decreases very slightly.
```

When controlling for total nestedness, the results clearly show that total nestedness is dominated by gaining nestedness between sites 1-2 (17 species), 2-3 (27 species) and 3-4 (38 species), and by losing nestedness between sites 4-5 (39 species), 5-6 (40 species), 6-7 (37 species) and 7-8 (34 species), as predicted by the relationship $\text{total.n} = \max(\text{gain.n}, \text{loss.n})$ mentioned in the previous subsection.

Neutral nestedness is the highest (97%) with respect to total nestedness between sites 4-5. This means that the balance between gain and loss is the largest between sites 4-5 in relation to the total nestedness, as predicted by the relationship $\text{neutral.n} = \min(\text{gain.n}, \text{loss.n})$ mentioned above.

Examine the partitioning of strict.nestedness into gaining.strict.n and losing.strict.n.

Strict nestedness – Here nestedness is interpreted in the strict sense, meaning that nestedness does not exist for sampling units with equal numbers of species.

```
(total.strict.n = sub.diag(Y2.2, method="gaining.strict.nestedness", relativize=NULL)$total.strict.n)
[1] 17 27 38 39 40 37 34
```

```
( gain.strict.n = sub.diag(Y2.2, method="gaining.strict.nestedness",relativize=NULL)$sub.diag )
# Method: gaining.strict.nestedness
[1] 17 27 38 38 37 34 31
```

```
( gain.strict.n / total.strict.n )
[1] 1.0000000 1.0000000 1.0000000 0.9743590 0.9250000 0.9189189 0.9117647
```

```
( loss.strict.n = sub.diag(Y2.2, method= "losing.strict.nestedness", relativize=NULL)$sub.diag )
# Method: losing.strict.nestedness
[1] 5 16 27 39 40 37 34
```

```
( loss.strict.n / total.strict.n )
[1] 0.2941176 0.5925926 0.7105263 1.0000000 1.0000000 1.0000000 1.0000000
```

For this coenocline, the interpretation is the same as with simple nestedness of the previous section. It is based on the relationship $\text{total.strict.n} = \max(\text{gain.strict.n}, \text{loss.strict.n})$. For other data sets, the interpretation could differ between the two types of nestedness.

=====

Part 2 – Coenocline with constant change in community composition

A coenocline *with constant change in community composition* can easily be generated with the `coenocline.R` function, but it would not produce a signature that could be identified by the directional indices described in the present paper. This is an example of the “null situation” in the paper. It shows what the new indices cannot do.

```
library(coenocliner)
```

Example of a coenocline with constant tolerances: 51 sites, 51 species with identical maximum abundances. After cut-off at values >2 (first 36 sites), Y contains a large proportion of zeros.

```
# Coenocline with constant tolerances,
x <- seq(from=1, to=51, by=1)    # Coenocline variable or transect positions (graph abscissa)
opt <- seq(from=1, to=51, by=1)  # Positions of species optima along transect x
tol <- rep(3.0, 51)              # Equal species tolerances
h <- rep(20, 51)                 # Maximum abundance of individual species (all equal)
```

```
Y.const <- coenocline(x, responseModel = "gaussian",
  params = cbind(opt = opt, tol = tol, h = h),
  countModel = "poisson", expectation = TRUE)
```

```
# Y.const contains about 46% zeros: length(which(Y.const<1e-5)) / length(Y.const)
```

```
Y.const.1 <- ifelse(Y.const > 2, 1, 0)
rich = apply(Y.const.1, 1, sum)    # Species richness at the 51 sites
[1] 7 8 9 10 11 12 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
[31] 13 13 13 13 13 13 13 13 13 13 13 13 13 13 12 11 10 9 8 7
plot(rich)                        # Plot species richness at the 51 sites
```

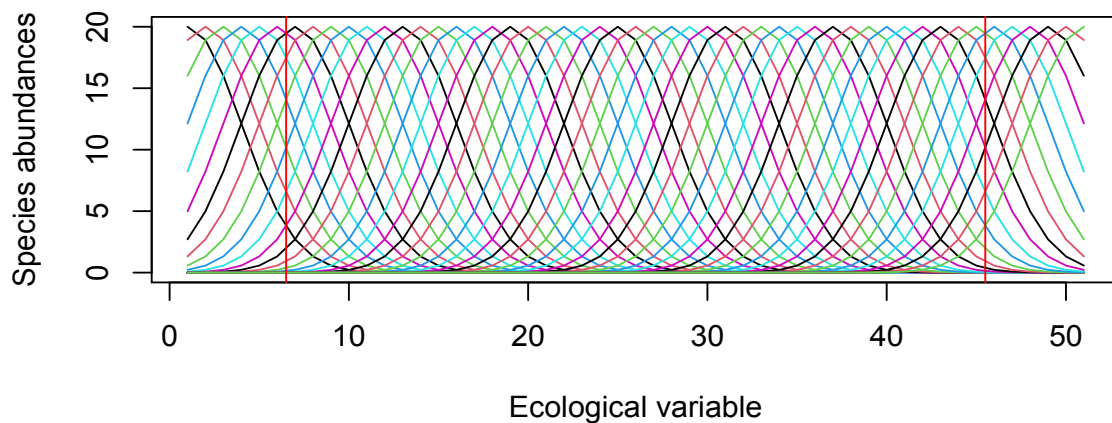
```
# The portion with constant richness (rich=13) is in the interval [7,45] along the ecological variable.
```

```
# Plot the coenocline with constant tolerances
```

```
plot(Y.const, type = "l", lty = "solid", xlab="Ecological variable",
  ylab="Species abundances", main="Coenocline, constant tolerances")
```

```
# Materialize this interval, in the graph, with vertical red lines at positions 6.5 and 45.5
abline(v=6.5, col="red") ; abline(v=45.5, col="red")
```

Coenocline, constant tolerances



Save the presence-absence data in that interval to a data file containing these 39 sites:

```
Y.const.39 = Y.const.1[7:45,]
```

We could run the function `directional.response` (see *Suppl Doc 2*) on the whole data file. Here we select the sites with spacing of 5, as in the first simulations above:

```
samp.seq = seq(1, 39, by=5)
```

We obtain 8 sites again:

```
samp.seq
[1] 1 6 11 16 21 26 31 36
```

```
Y.const.8 = Y.const.39[samp.seq,]      # or: Y.const.8 = Y.const.1[seq(7, 45, by=5),]
```

```
dim(Y.const.8)
[1] 8 51
```

Load functions `directional.response.R` and `sub.diag.R`, as in part 1 of this Supplement

Examine the partitioning of turnover into gain.t, loss.t and neutral.t

Turnover – The change in species composition between adjacent sites along a gradient.

```
( total.t = sub.diag(Y.const.8, method= "gaining.turnover", relativize=NULL)$total.t )
[1] 10 10 10 10 10 10 10      # Total turnover = = max(gain.t, loss.t) (computed below)
```

```
( gain.t = sub.diag(Y.const.8, method= "gaining.turnover", relativize=NULL)$sub.diag )
Method: gaining.turnover
[1] 10 10 10 10 10 10 10      # if(c>b then b+c, else 2*min(b,c)
```

```
( loss.t = sub.diag(Y.const.8, method= "losing.turnover", relativize=NULL)$sub.diag )
Method: losing.turnover
[1] 10 10 10 10 10 10 10      # if(c<b then b+c, else 2*min(b,c)
```



```
( neutral.t = sub.diag(Y.const.8, method= "neutral.turnover", relativize=NULL)$sub.diag )
# Method: neutral.turnover      # 2*min(b,c)
[1] 10 10 10 10 10 10 10 10
```

=> Note: total.t = max(gain.t, loss.t); neutral.t = min(gain.t, loss.t)

Examine the partitioning of nestedness into gain.n, loss.n and neutral.n.

Nestedness – A type of richness difference pattern characterized by the species at a site being a strict subset of the species at a richer site.

```
( total.n = sub.diag(Y.const.8, method= "gaining.nestedness", relativize=NULL)$total.n )
[1] 8 8 8 8 8 8 8 8
```

```
( gain.n = sub.diag(Y.const.8, method= "gaining.nestedness", relativize=NULL)$sub.diag )
# Method: gaining.nestedness
[1] 8 8 8 8 8 8 8 8      # Constant values
```

```
( loss.n = sub.diag(Y.const.8, method= "losing.nestedness", relativize=NULL)$sub.diag )
# Method: losing.nestedness
[1] 8 8 8 8 8 8 8 8      # Constant values
```

```
( neutral.n = sub.diag(Y.const.8, method= "neutral.nestedness", relativize=NULL)$sub.diag )
# Method: neutral.nestedness
[1] 8 8 8 8 8 8 8 8      # Constant values
```

=> Note: total.n = max(gain.n, loss.n); neutral.n = min(gain.n, loss.n)

Analysis of these data (all 39 sites, or reduced to 8 sites as in this example) with the new directional indices did not produce any signal because there is no variation in the *rates of change* between sites. It does not mean that there is no directionality, but that our indices are not meant to capture community composition changes with constant rates.

The directionality of the broad-scale data structure could still be shown by a PCoA of the Jaccard or Sørensen dissimilarity matrix. Indeed, in the PCoA ordination, the sites are ordered following their entry positions in the matrix. If the replacement of species is regular along the coenocline, an ordination should show a standard horseshoe shape. An example follows.

Demonstration – The coenocline produces a regular horseshoe shape in PCoA

```
library(ape) # where functions pcoa and biplot.pcoa are found
library(adespatial) # where function dist.ldc is found
pcoa.res = pcoa(dist.ldc(Y.const.8, "sorensen"))
biplot(pcoa.res, type="n", main=c("PCoA ordination of Yconst.8", " Sørensen dissimilarity"))
lines(pcoa.res$vectors[,1:2], type="l", col="red")
points(pcoa.res$vectors[,1:2], type="p", pch=21, bg="white")
```

