

The species chromatogram, a new graphical method to represent, characterise and compare the ecological niches of different species

Application with R

Loïck Kléparski^{1,2} and Grégory Beaugrand²

¹Univ. Littoral Côte d'Opale, CNRS, Univ. Lille, UMR 8187 - LOG - Laboratoire d'Océanologie et de Géosciences, F-62930 Wimereux, France

²Marine Biological Association, Citadel Hill, Plymouth PL1 2PB, United Kingdom.

October 19, 2021

In this document, we describe how to use the R functions available on Github in order to reproduce a part of our results. Only three main functions are used : `chromato_env16.R`, `opti_eury_niche2.R` and `opti_eury_niche2.R`. The following packages have to be installed: `abind`, `colorRamps`, `ggplot2` and `reshape2`. Noted that the same procedure is applicable with Matlab.

Function `chromato_env16.R`

The function `chromato_env16.R` estimates and displays the chromatogram of a given species. This function takes six arguments in the following order :

```
chromato_env16(z,y,alpha,m,k,order_smth)
```

With `z` a matrix with `n` samples by `p` environmental variables (i.e. the value of each environmental variable in each sample), `y` a vector with the abundance of a species in the `n` samples, `alpha` an integer corresponding to the number of category along each environmental variable, `m` an integer corresponding to the lowest number of samples needed in a category in order to have an estimation of the mean abundance, `k` an integer corresponding to the percentage of samples with the highest abundance values to use to estimate the mean abundance in a given category and `order_smth` an integer corresponding the order of the simple moving average applied along each niche dimension.

`chromato_env16.R` used the functions `nanmean4.R`, which estimates the mean of the `k`% of the samples with the highest abundance and `moymob1.R` which applies the moving average.

Example with CPR data:

Read the code of the function `chromato_env16.R`:

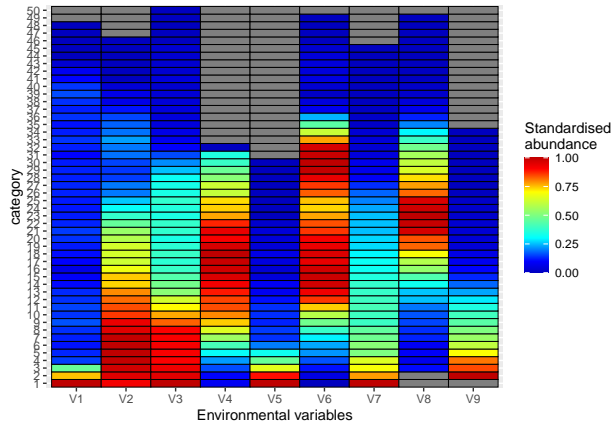
```
source("chromato_env16.R")
```

Load the files `cpr_env.csv` which contained the 90,732 CPR samples by 10 environmental variables (i.e. bathymetry, nitrate, phosphate, silicate, MLD, temperature, PAR, salinity, chlchlorophyll-a and euphotic depth) and `data_phyto.csv` which contained the abundance of 45 diatoms species in each sample (90,732 samples by 45 species). (The abundance of *Paralia sulcata*, *Skeletonema costatum*, *Rhizosolenia styliformis* and *R. bergonii* are stored in the column number 1, 2, 5 and 28 respectively):

```
cpr_env<-read.csv("cpr_env.csv",header = FALSE,sep = ",",dec = ".",na='NaN')
data_phyto<-read.csv("data_phyto.csv",header = FALSE,sep = ",",dec = ".")
```

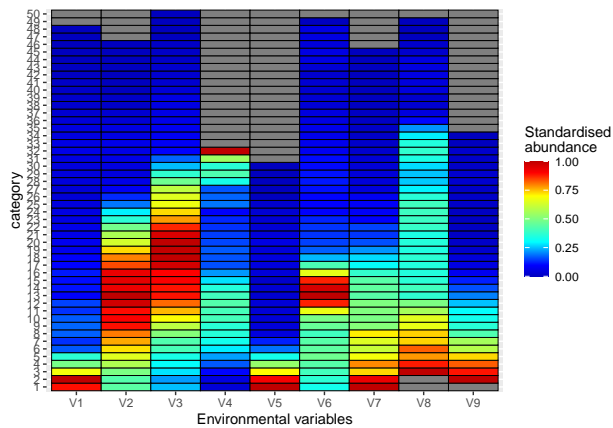
Then apply the function `chromato_env16.R` for *Paralia sulcata*:

```
y_psulcata<-chromato_env16(cpr_env[,c(1:8,10)],data_phyto[,1],50,20,5,2)
```



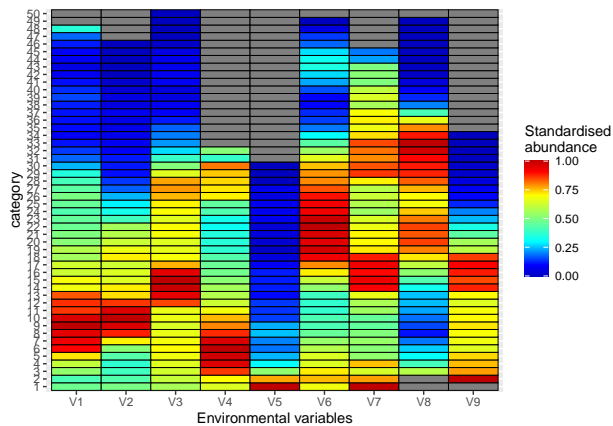
Skeletonema costatum:

```
y_scosta<-chromato_env16(cpr_env[,c(1:8,10)],data_phyto[,2],50,20,5,2)
```



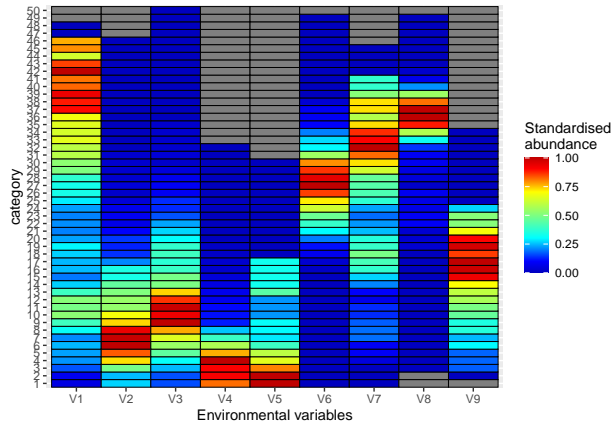
Rhizosolenia styliiformis:

```
y_rstyli<-chromato_env16(cpr_env[,c(1:8,10)],data_phyto[,5],50,20,5,2)
```



and *Rhizosolenia bergonii*:

```
y_rbergonii<-chromato_env16(cpr_env[,c(1:8,10)],data_phyto[,28],50,20,5,2)
```



Function `opti_eury_niche2.R`

The function `opti_eury_niche2.R` estimates the niche optimums and breadths of each species along each niche dimension, i.e. each environmental variable. The mean niche breadth is also estimated. This function takes six arguments in the following order :

```
opti_eury_niche2(sp_chr,T,z,y,k)
```

With `sp_chr` a three dimensional matrices with the species chromatograms (alpha category by p environmental variables by species), `T` the threshold of minimal abundance in a category for the niche breadth estimation, `z` a matrix with n samples by p environmental variables (i.e. the value of each environmental variable in each sample), `y` a matrix with the abundance of the species in the n samples and `k` the percentage of samples with the highest abundance used for the mean abundance estimation.

Example with CPR data:

Read the code of the function `opti_eury_niche2.R`:

```
source("opti_eury_niche2.R")
```

The function `abind` can be used to combine the four species chromatograms along a third dimension:

```
library(abind)
test_phyto<-abind(y_psulcata,y_scosta,y_rstyli,y_rbergonii,along=3)
```

The function `opti_eury_niche2.R` can then be applied:

```
opti_eury_niche2(test_phyto,0.05,cpr_env[,c(1:8,10)],data_phyto[,c(1,2,5,28)],5)
```

The degree of euryoecy (i.e. niche breadth) of each species along each dimension are stored in `deg_eury`, a table with in line the p environmental variables (here nine, i.e. 1=bathymetry, 2=nitrate, 3=phosphate, 4=silicate, 5=MLD, 6=temperature, 7=PAR, 8=salinity and 9=euphotic depth) and in column the species (here four, i.e. 1=*Paralia sulcata*, 2=*Skeletonema costatum*, 3=*Rhizosolenia styliiformis* and 4=*R. bergonii*):

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] 87.50000 72.91667 100.00000 95.83333
## [2,] 84.78261 76.08696  84.78261 52.17391
## [3,] 68.00000 78.00000  90.00000 56.00000
## [4,] 96.87500 96.87500 100.00000 43.75000
## [5,] 60.00000 66.66667  90.00000 56.66667
## [6,] 81.63265 97.95918  97.95918 40.81633
```

```
## [7,] 66.66667 57.77778 100.00000 82.22222
## [8,] 74.46809 89.36170 80.85106 42.55319
## [9,] 51.51515 48.48485 78.78788 66.66667
```

Mean degree of euryoecy of each species is stored in `mean_deg_eury`:

```
## [1] 74.60446 76.01431 91.37564 59.63137
```

Niche optimums for each species (i.e. four) along each niche dimension (i.e. nine) are stored in `opti_val`:

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 43.71342275 164.3214263 1024.7535885 4924.597523
## [2,]  1.85316914   4.2368076   3.2344707   1.857350
## [3,]  0.00387912   0.3957169   0.2935805   0.172401
## [4,]  5.57933491   9.8454771   1.9849569   1.383753
## [5,] 18.26105085  18.3356583   17.8111185   43.180671
## [6,] 16.45252437   6.3075089   11.6712827   14.658172
## [7,]  0.49425302   0.4489858   0.3879702   25.029722
## [8,] 33.17704454  29.4797783   35.0331347   35.576847
## [9,] 10.44985542   9.5844252   10.5505980   51.686962
```

Function `combina_niche3.R`

The function `combina_niche3.R` estimates the index of niche overlapping (D) among species niche. This function takes two arguments in the following order :

```
combina_niche3(sp_chr,T)
```

With `sp_chr` a three dimensional matrix with the species chromatograms (alpha category by p environmental variables by species) and `T` the threshold of minimal abundance in a category for the niche breadth estimation.

`combina_niche3.R` used the functions `niche_difer_sp.R` and `niche_difer2.R`.

Example with CPR data:

Read the code of the function `combina_niche3.R` and apply the function on the previous three-dimensional matrix `test_phyto`:

```
source("combina_niche3.R")
combi_dim_phyto<-combina_niche3(test_phyto,0.05)
```

The mean degree of niche overlapping is stored in `results1` (renamed here as `combi_dim_phyto`). The first column displays the number of dimensions considered simultaneously, columns 2 to 10 display the combinations of dimensions (i.e. 1=bathymetry, 2=nitrate, 3=phosphate, 4=silicate, 5=MLD, 6=temperature, 7=PAR, 8=salinity and 9=euphotic depth). The last column displays index D associated with the combination of environmental dimensions. D=0% when species niches are fully different and D=100% when species niches are identical; the higher the number of dimensions, the lower the value of index D. Only the combinations of environmental variables that minimise values of index D are displayed :

```
##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## [1,]      1     6    NA    NA    NA    NA    NA    NA    NA    NA 66.66667
## [2,]      2     6     7    NA    NA    NA    NA    NA    NA    NA 47.74995
## [3,]      3     6     7     8    NA    NA    NA    NA    NA    NA 36.26314
## [4,]      4     6     7     8     9    NA    NA    NA    NA    NA 28.26879
## [5,]      5     5     6     7     8     9    NA    NA    NA    NA 22.83315
## [6,]      6     3     5     6     7     8     9    NA    NA    NA 18.95416
## [7,]      7     3     4     5     6     7     8     9    NA    NA 16.06675
## [8,]      8     1     3     4     5     6     7     8     9    NA 13.90961
## [9,]      9     1     2     3     4     5     6     7     8     9 12.49415
```