dispRity demo for ecologists

Thomas Guillerme
guillert@tcd.ie

November 28, 2015

This demo aims to give quick overview of the dispRity package (v.0.1.2) for ecological analysis. Please refer to GitHub page: github.com/TGuillerme/dispRity for other vignettes, namely the dispRity tutorial that explains the functions in more detail.

To keep it short, this package allows the use of all of the dimensions of ordinated matrices (i.e. PCA, MDS, PCO) for statistical analysis rather than just a sub-set of dimensions. For example, one might want to know whether the sort of water treatment at certain depth alters invertebrate communities and composition in natural habitats.

Contents

1 Before starting

1.1 Installing dispRity

You can install this package easily if you are using the latest version of R and devtools.

```
install.packages("devtools")
library(devtools)
install_github("TGuillerme/dispRity", ref = "release")
library(dispRity)
```

This is a quick demo for using the dispRity package (v.0.1.2) in ecological analysis. See the other dispRity demos for a general demo of the dispRity package.

1.2 Data

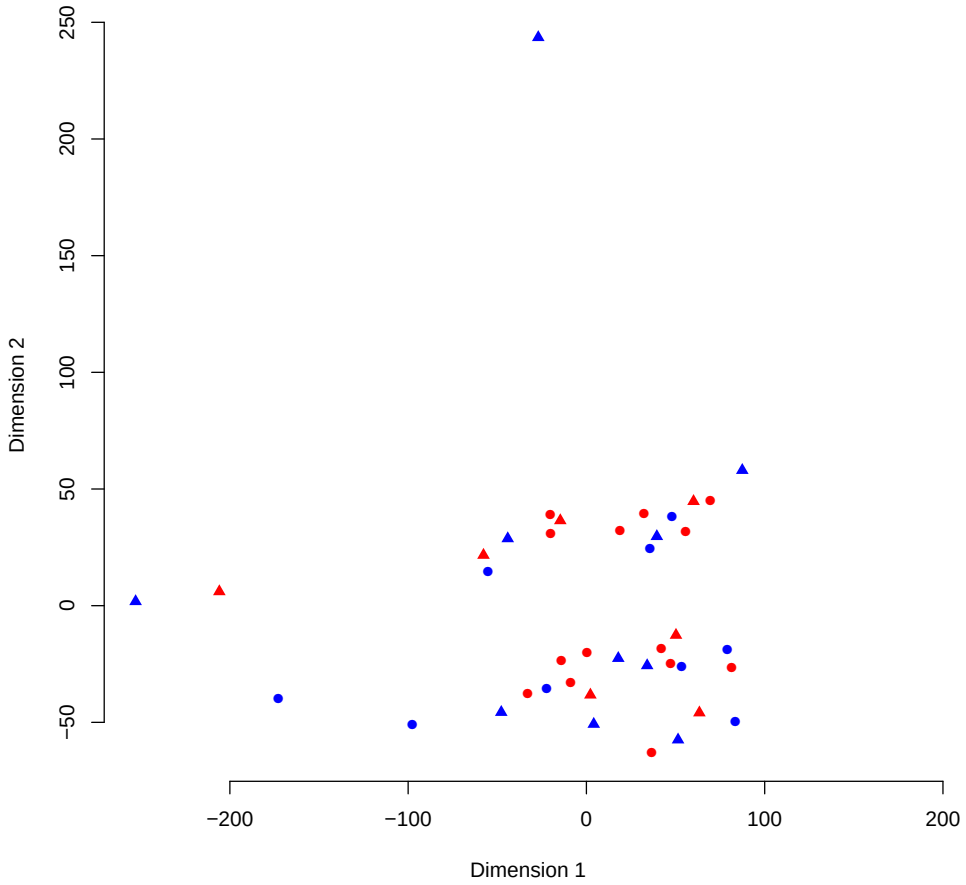
For this example with ecological data we are going to use data from McClean (unpubl.). This data is the ordination of a distance matrix based on nutrient enrichment and depth and freshwater benthic invertebrates.

```
## Loading demo and the package data
library(dispRity)

## Loading required package:  paleotree

## Setting the random seed for repeatability
set.seed(123)

## Loading the data
data(McClean_data)

## This dataset contains an ordinated matrix (20 dimensions) of the distance
## between 40 experimental plots.
ord_matrix <- McClean_data$ordination

## As well as two list of different factors affecting each experimental plot:
## the treatment and the depth.
treatments <- McClean_data$treatment
depth <- McClean_data$depth
```

2 A classical two dimensional approach

A classical way to represent this ordinated data will be to use two dimensional plots.

```
## The x and y axis represent the two first dimensions of the ordination
x <- ord_matrix[, 1]
y <- ord_matrix[, 2]
## The colors will represent the treatments
cols <- sub("a", "red", treatments)
cols <- sub("b", "blue", cols)
## The symbols will represent the depth
pchs <- sub(1, 16, depth)
pchs <- as.numeric(sub(2, 17, pchs))

## Graphical option
par(bty = "n")
## A classic 2D ordination plot
plot(x, y, col = cols, pch = pchs, xlab = "Dimension 1", ylab = "Dimension 2",
    xlim = range(x) + c(0, 100))
```



This shows the distribution of the experimental plots along the two first axis of variance of the ordinated distance matrix (i.e. the first two dimensions). At a first glance, it seems difficult to see a clear effect from the treatments or the depth since the different experimental plots with the same parameters don't seem to cluster. However, the problem, is that this plot ignores the 18 other dimensions of the ordination! Additionally, these two represented dimensions do not represent a biological reality *per se*; i.e. the values on the first dimension do not represent some continuous traits (e.g. depth) but just the ordinations of correlations between the data and some factors.

Therefore, one might want to approach this problem without getting stuck in only two dimensions and consider the whole dataset as a n-dimensional object. To explore our hypothesis, we could look to see if there are differences in the hyper-volume that different treatments occupy in the ordinated space to see the effect of water treatments on the communities.

3 A multidimensional approach with `dispRity`

3.1 Splitting the data

The first step in such analysis will be to create different series that are two sub-samples of the ordinated space (i.e. the n-dimensional object). Each of these series contain a certain number of elements (i.e. the 40 experimental field plots) that have some attributes in common. In our example, we are going to group the elements according to their depth and treatment.

```
## Creating the table that contain the elements and their attributes
factors <- as.data.frame(matrix(data = c(treatments, depth), nrow = nrow(ord_matrix),
    ncol = 2, byrow = FALSE, dimnames = list(rownames(ord_matrix))))
names(factors)<-c("Treat", "Depth")
head(factors)

##     Treat Depth
## 1       a     1
## 1.1     a     2
## 2       b     1
## 2.1     b     2
## 3       a     1
## 3.1     a     1
```

Second, let's split the data according to these factors to create the series of the ordinated space by using the `cust.series` function:

```
## Splitting the ordinated space into four subsamples
customised_series <- cust.series(ord_matrix, factors)

## Note that the output of dispRity functions are dispRity objects
class(customised_series)

## [1] "dispRity"

## These objects are automatically printed in a summary method (calling S3 print.dispRity)
## giving information about the object
customised_series

## 4 custom series for 40 elements
## Series:
## Treat.a, Treat.b, Depth.1, Depth.2.
```

For more details on the `dispRity` objects, see the `dispRity` manual. Basically the idea is to avoid jamming the `R` console such as when using:

```
## Summarising the object
str(customised_series)

## Displaying the full object
print.dispRity(customised_series, all=TRUE)
```

3.2 Calculating disparity

Disparity can be calculated in many ways, and therefore, the `dispRity` function allows to measure disparity *as defined by the user*. For more details on disparity, see the other vignettes.

In this example, we are going to define disparity as the median distance between the different experimental plots and the centroid of the ordinated space. High values of disparity will indicate a general high dispersal from this centroid (i.e. on average, the experimental plots are far apart in the ordinated space). We can define the metrics easily in the `dispRity` function by feeding them to the `metric` argument. Here we are going to feed the functions `median::stats` and `centroids::dispRity` which calculates distances between elements and centroid.

```
## Calculating disparity
disparity <- dispRity(customised_series, metric = c(median, centroids))

## Note that disparity is a dispRity object and printing it just gives details
## on the object, not the results. We need to use summary.dispRity (S3) to get
## the results.
disparity

## Disparity measurements across 4 series for 40 elements
## Series:
## Treat.a, Treat.b, Depth.1, Depth.2.
## Disparity calculated as: c(median, centroids) for 20 dimensions.
## Data was split using custom method.

summary(disparity)

##    series  n observed
## 1 Treat.a 21    82.38
## 2 Treat.b 19    94.68
## 3 Depth.1 23    83.98
## 4 Depth.2 17    89.72
```

3.3 Bootstrapping the data

One might also want to bootstrap the data to test the robustness of the measured disparity to outliers. Also, as we can see, each series has different numbers of elements. It might be interesting to rarefy the data as well to have only series with the same number of elements. Both steps are easily doable through the `boot.matrix` function.

```
## Simple bootstrapping (1000 times)
bootstrapped_data <- boot.matrix(customised_series, bootstraps = 100)
## Bootstrapping with rarefaction (i.e. only re-sampling 17 elements each time)
rarefied_data <- boot.matrix(customised_series, bootstraps = 100, rarefaction = 17)
## Note that the output is a dispRity object giving some details on the series and the bootstraps
bootstrapped_data

## Bootstrapped ordinated matrix with 40 elements
## Series:
## Treat.a, Treat.b, Depth.1, Depth.2.
## Data was split using custom method.
## Data was bootstrapped 100 times, using the full bootstrap method.

rarefied_data

## Bootstrapped ordinated matrix with 40 elements
## Series:
## Treat.a, Treat.b, Depth.1, Depth.2.
## Data was split using custom method.
## Data was bootstrapped 100 times, using the full bootstrap method.
## Data was rarefied with a maximum of 17 elements
```

We can now rerun a more robust disparity analysis using the bootstrapped data:

```
## Bootstrapped disparity
disparity_BS <- dispRity(bootstrapped_data, metric = c(median, centroids))
## Rarefied disparity
disparity_rare <- dispRity(rarefied_data, metric = c(median, centroids))
```

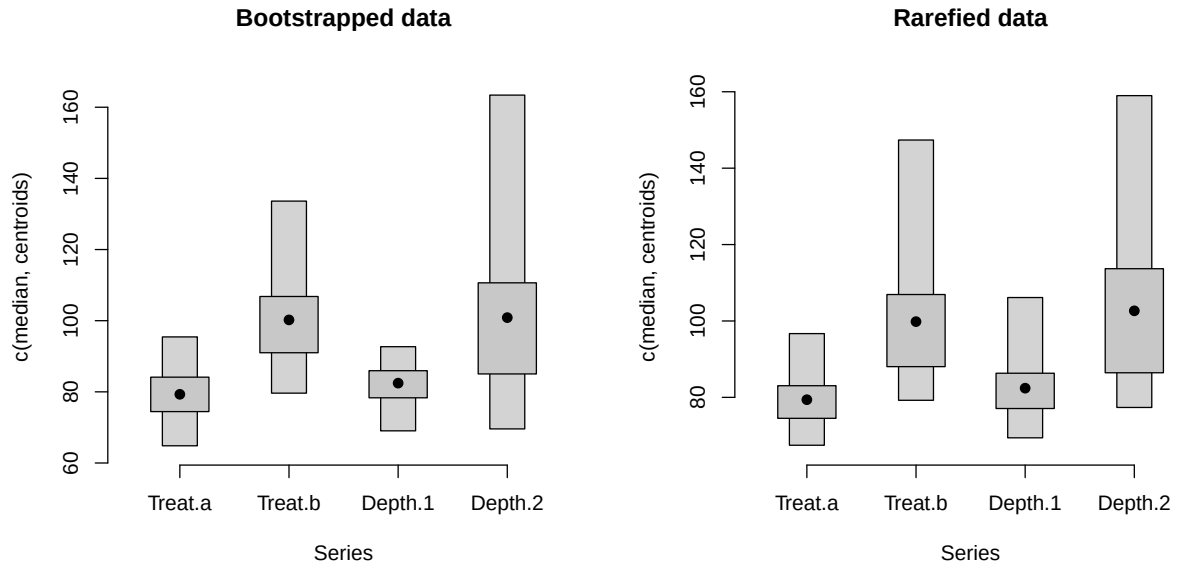
3.4 Summarising the data

We can now summarise the data using various options such as the confidence intervals levels and the central tendency.

```
## Summarising disparity (default)
summary(disparity_BS)

##    series  n observed  mean  2.5%   25%   75% 97.5%
## 1 Treat.a 21    82.38  79.3 64.83 74.45  84.1  95.4
## 2 Treat.b 19    94.68 100.2 79.63 90.99 106.8 133.6
## 3 Depth.1 23    83.98  82.4 69.04 78.32  85.9  92.7
## 4 Depth.2 17    89.72 100.9 69.58 85.03 110.6 163.4

## The quantiles are calculated as 50 and 95 and the central tendency is the mean by default.
## But we can specify different options
summary(disparity_BS, quantile=90, cent.tend=median)

##    series  n observed median    5%   95%
## 1 Treat.a 21    82.38  78.50 68.34  91.2
## 2 Treat.b 19    94.68  97.02 83.17 128.3
## 3 Depth.1 23    83.98  83.21 72.79  91.3
## 4 Depth.2 17    89.72  93.04 75.77 149.4

## Finally we can see the results of the rarefaction analysis:
summary(disparity_rare)

##    series  n observed  mean  2.5%   25%   75% 97.5%
## 1 Treat.a 17       NA  79.4 67.47 74.52  83.1  96.7
## 2 Treat.b 17       NA  99.8 79.24 88.04 106.9 147.4
## 3 Depth.1 17       NA  82.4 69.41 77.08  86.3 106.1
## 4 Depth.2 17    89.72 102.7 77.36 86.44 113.7 159.0

## Note that for the first three series, there is no observed disparity since the data was
## rarefied for these series.
```

We can also plot the results and have a look at the effect of the number of experimental plots:

```
## Graphical options
quartz(width = 10, height = 5) ; par(mfrow = (c(1,2)), bty = "n")
## Plotting the bootstrapped disparity
plot(disparity_BS, main="Bootstrapped data")
## Plotting the rarefied disparity
plot(disparity_rare, main="Rarefied data")
```



As we can see, there seems to be no strong effect of the number of experimental plots in each series (i.e. the rarefied plot is really similar to the bootstrapped plot) which is a good thing!

3.5 Testing the hypothesis

Finally, we can test our hypothesis (whether the sort of water treatment at certain depth alters invertebrate communities and composition in natural habitats) by using the `test.dispRity` function.

```
## Testing the effect of our factors on the bootstrapped data
summary(test.dispRity(disparity_BS, test = aov, comparisons = "all"))

##              Df Sum Sq Mean Sq F value Pr(>F)
## series        3  39184   13061   61.17 <2e-16 ***
## Residuals   396  84553     214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Testing the effect of our factors on the rarefied data
summary(test.dispRity(disparity_rare, test = aov, comparisons = "all"))

##              Df Sum Sq Mean Sq F value Pr(>F)
## series        3  42229   14076   63.85 <2e-16 ***
## Residuals   396  87301     220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Post-hoc testing of the effect of the two different factors
## Note that the comparison contains the list of the pairs of series to compare
test.dispRity(disparity_BS, test = wilcox.test, comparison = list(c(1,2), c(3,4)))

##                       W      p.value
## Treat.a - Treat.b   695 7.170082e-26
## Depth.1 - Depth.2  2064 7.360379e-13

## Testing the effect of the two different factors for the rarefied data
test.dispRity(disparity_rare, test = wilcox.test, comparison = list(c(1,2), c(3,4)))
```

```
##                       W      p.value
## Treat.a - Treat.b   880 7.845518e-24
## Depth.1 - Depth.2  1599 9.672040e-17
```

As we can see, there is strong support for an effect of the treatment and the depth on the median distance between each experimental plots and the centroid of the ordinated space. In other words, it seems that with the second treatment and the second level of depth, the invertebrate communities and composition were further apart (i.e. more dispersed in the ordinated space) than with the first treatment and the first level of depth.