`dispRity` demo for palaeobiologists

Thomas Guillerme
guillert@tcd.ie

November 24, 2015

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

To keep it short, this package allows to easily perform **disparity-through-time** analysis. This type of analysis often starts by ordinating **morphometric** or **cladistic** data into a multidimensional object hereafter called the **morphospace**. One might then be interested how the occupancy of the morphospace changes through time by measuring a summary metric of this morphospace through time, called **disparity**. In the morphospace case, disparity might be seen as a value summarising the diversity of morphologies.

This demo showcases a typical disparity-through-time analyse with the following steps:

1. Installing `dispRity`
2. The morphospace
3. Splitting the morphospace through time
4. Bootstrapping the data
5. Calculating disparity
6. Plotting the results
7. Testing differences

In this demo we are going to test whether the disparity changed through time in this dataset since the last 100 million years.

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)
```

2 The morphospace

In this example, we are going to use a subset of the data from Beck and Lee (2014) that contains an ordinated cladistic matrix and a phylogenetic tree containing both the same 50 taxa along side a table that contains some first and last occurence data for several taxa. The ordinated matrix will represent our full morphospace, i.e. all the morphologies that ever existed through time (for this dataset).

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

```
## Loading required package:  paleotree
```

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

## The morphospace:
data(BeckLee_mat50)
head(BeckLee_mat50[,1:5])
```

```
##                    [,1]          [,2]        [,3]       [,4]      [,5]
## Cimolestes   -0.5319679  0.1117759259  0.09865194 -0.1933148 0.2035833
## Maelestes    -0.4087147  0.0139690317  0.26268300  0.2297096 0.1310953
## Batodon      -0.6923194  0.3308625215 -0.10175223 -0.1899656 0.1003108
## Bulaklestes  -0.6802291 -0.0134872777  0.11018009 -0.4103588 0.4326298
## Daulestes    -0.7386111  0.0009001369  0.12006449 -0.4978191 0.4741342
## Uchkudukodon -0.5105254 -0.2420633915  0.44170317 -0.1172972 0.3602273
```

```
dim(BeckLee_mat50)
```

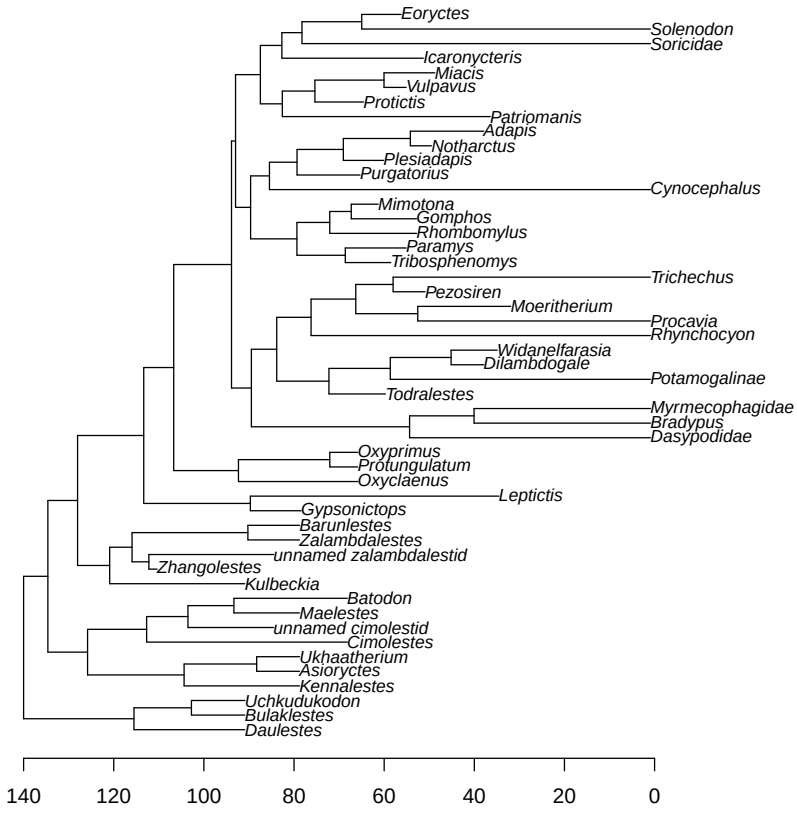
```
## [1] 50 48
```

```
## The morphospace contains 50 taxa and has 48 dimensions (or axis)

## A list of first and last occurrences data for some fossils
data(BeckLee_ages)
head(BeckLee_ages)
```

```
##             FAD  LAD
## Adapis     37.2 36.8
## Asioryctes 83.6 72.1
## Leptictis  33.9 33.3
## Miacis     49.0 46.7
## Mimotona   61.6 59.2
## Notharctus 50.2 47.0
```

```
## And a phylogeny:
data(BeckLee_tree)
plot(BeckLee_tree, cex=0.8); axisPhylo(root=140)
```



Note that you can have an even nicer tree if you use the `strap` package:

```
install.packages(strap)
library(strap)
geoscalePhylo(BeckLee_tree, cex.tip=0.8, cex.ts=0.6)
```

3 Splitting the morphospace through time

One of the crucial steps in disparity-through-time analysis is to split the full morphospace into smaller sub-samples that contain the total amount of morphologies at certain points in time (time-slicing) or during certain periods in time (time-binning). Basically, the full morphospace in the previous step represents the total amount of morphologies in all time and is likely to be greater than any sub-sample of the morphospace.

The `dispRity` package provides a `time.series` function that allows to split the morphospace time slices (using `method="continuous"`) or into time bins (using `method="discrete"`). In this example, we are going to split the morphospace into 5 equal time bins of 20 million years long from 100 million years ago (Mya) to the present. Note that we can also provide here the table containing the first and last occurrences data for some fossils to take into account some fossils that might span between our different bins.

```
## Creating the vector of time bins ages
time_bins <- rev(seq(from=0, to=100, by=20))

## Splitting the morphospace using the time.series function
binned_morphospace <- time.series(data = BeckLee_mat50, tree = BeckLee_tree,
    method = "discrete", time = time_bins, inc.nodes = FALSE,
    FADLAD = BeckLee_ages)

## Some tips have no FAD/LAD and are assumed to be single points in time.
```

The warning message tells us that some taxa were considered as single points in time because we didn't provide any temporal range information for some taxa. The output object is a `dispRity` object. For details about this object class, please refer to the other vignettes. In brief however, `dispRity` objects are lists of different elements (i.e. disparity results, morphospace sub-samples, etc...) that display only a summary of the object when calling the object for visualisation reasons.

```
## The object class
class(binned_morphospace)

## [1] "dispRity"

## What's in the object (messy version for the console!)
str(binned_morphospace)

## List of 3
##  $ data    :List of 5
##   ..$ 100-80: num [1:8, 1:48] -0.739 -0.68 -0.511 -0.473 -0.459 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:8] "Daulestes" "Bulaklestes" "Uchkudukodon" "Asioryctes" ...
##   .. .. ..$ : NULL
##   ..$ 80-60 : num [1:15, 1:48] -0.477 -0.473 -0.474 -0.532 -0.409 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:15] "Kennalestes" "Asioryctes" "Ukhaatherium" "Cimolestes" ...
##   .. .. ..$ : NULL
##   ..$ 60-40 : num [1:13, 1:48] -0.165 0.254 0.622 0.753 0.671 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:13] "Todralestes" "Pezosiren" "Tribosphenomys" "Paramys" ...
##   .. .. ..$ : NULL
##   ..$ 40-20 : num [1:6, 1:48] -0.121 -0.203 -0.158 0.696 0.355 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "Leptictis" "Dilambdogale" "Widanelfarasia" "Moeritherium" ...
##   .. .. ..$ : NULL
##   ..$ 20-0  : num [1:10, 1:48] 0.3079 0.6531 0.5089 -0.0419 0.235 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:10] "Dasypodidae" "Bradypus" "Myrmecophagidae" "Potamogalinae" ...
##   .. .. ..$ : NULL
##  $ elements: chr [1:50] "Cimolestes" "Maelestes" "Batodon" "Bulaklestes" ...
##  $ series  : chr [1:6] "discrete" "100-80" "80-60" "60-40" ...
##  - attr(*, "class")= chr "dispRity"

names(binned_morphospace)

## [1] "data"     "elements" "series"

## How it's printed by default
binned_morphospace
```

```
## 5 discrete series for 50 elements
## Series:
## 100-80, 80-60, 60-40, 40-20, 20-0.
```

Note that these objects will gradually contain more information when reaching higher steps in the disparity-through-time analysis.

4 Bootstrapping the data

Once we obtain our different subsamples, we might want to bootstrap or/and rarefy it (i.e. pseudo-replicating the data). The bootstrap will allow us to make each subsample more robust to outliers and the rarefaction will allow us to compare slices with the same number of taxa to get rid of eventual sampling problems. For that, we can use the `boot.matrix` function that allows to bootstrap the `dispRity` object along with the `rarefaction` option that will rarefy it.

```
## Bootstrapping each sub-sample 100 times
boot_bin_morphospace <- boot.matrix(binned_morphospace, bootstraps = 100)

## Getting the minimum number of rows (=taxa) in the sub-samples
min(unlist(lapply(binned_morphospace$data, nrow)))

## [1] 6

## Bootstrapping each sub-sample 100 times and rarefying them
rare_bin_morphospace <- boot.matrix(binned_morphospace, bootstraps = 100,
    rarefaction = 6)

## Note that the output objects are both from the "dispRity" class ...
class(boot_bin_morphospace); class(rare_bin_morphospace)

## [1] "dispRity"
## [1] "dispRity"

## ... and display some more information on the previous step when called
boot_bin_morphospace

## Bootstrapped ordinated matrix with 50 elements
## Series:
## 100-80, 80-60, 60-40, 40-20, 20-0.
## Data was split using discrete method.
## Data was bootstrapped 100 times, using the full bootstrap method.

rare_bin_morphospace

## Bootstrapped ordinated matrix with 50 elements
## Series:
## 100-80, 80-60, 60-40, 40-20, 20-0.
## Data was split using discrete method.
## Data was bootstrapped 100 times, using the full bootstrap method.
## Data was rarefied with a maximum of 6 elements
```

5 Calculating disparity

We can now calculate the disparity within each sub-sample along side with some confidence intervals generated by the pseudo-replication step above (bootstrap/rarefaction). Disparity can be calculated in many ways and this package allows user to come up with their own disparity metrics. For more info, please refer to the other vignettes.

In this example, we are going to calculate the spread of the data in each sub-sample by calculating the disparity as the sum of the variance of each dimension of the morphospace in each sub-sample using the `dispRity` function.

```
## Disparity for the bootstrapped data
boot_disparity <- dispRity(boot_bin_morphospace, metric = c(sum, variances))

## Disparity for the rarefied data
rare_disparity <- dispRity(rare_bin_morphospace, metric = c(sum, variances))

## Again, these objects are both from the "dispRity" class and just display
## some information when called, note the results!
boot_disparity
```

```
## Disparity measurements across 5 series for 50 elements
## Series:
## 100-80, 80-60, 60-40, 40-20, 20-0.
## Disparity calculated as: c(sum, variances) for 48 dimensions.
## Data was split using discrete method.
## Data was bootstrapped 100 times, using the full bootstrap method.
```

```
rare_disparity
```

```
## Disparity measurements across 5 series for 50 elements
## Series:
## 100-80, 80-60, 60-40, 40-20, 20-0.
## Disparity calculated as: c(sum, variances) for 48 dimensions.
## Data was split using discrete method.
## Data was bootstrapped 100 times, using the full bootstrap method.
## Data was rarefied with a maximum of 6 elements
```

```
## To display the actual results we can use the summary function
summary(boot_disparity)
```

```
##   series  n observed  mean  2.5%   25%   75% 97.5%
## 1 100-80  8    1.675 1.460 1.087 1.389 1.568 1.648
## 2  80-60 15    1.782 1.678 1.538 1.631 1.728 1.792
## 3  60-40 13    1.913 1.773 1.607 1.734 1.826 1.886
## 4  40-20  6    2.022 1.673 1.212 1.537 1.822 1.942
## 5   20-0 10    1.971 1.773 1.598 1.716 1.842 1.890
```

```
summary(rare_disparity)
```

```
##   series n observed  mean  2.5%   25%   75% 97.5%
## 1 100-80 6       NA 1.440 1.067 1.339 1.575 1.726
## 2  80-60 6       NA 1.670 1.276 1.614 1.798 1.922
## 3  60-40 6       NA 1.763 1.458 1.704 1.846 1.986
## 4  40-20 6     2.02 1.721 1.392 1.633 1.828 1.992
## 5   20-0 6       NA 1.762 1.459 1.642 1.880 2.009
```

```
## See more options on the summary function:
?summary.dispRity
```

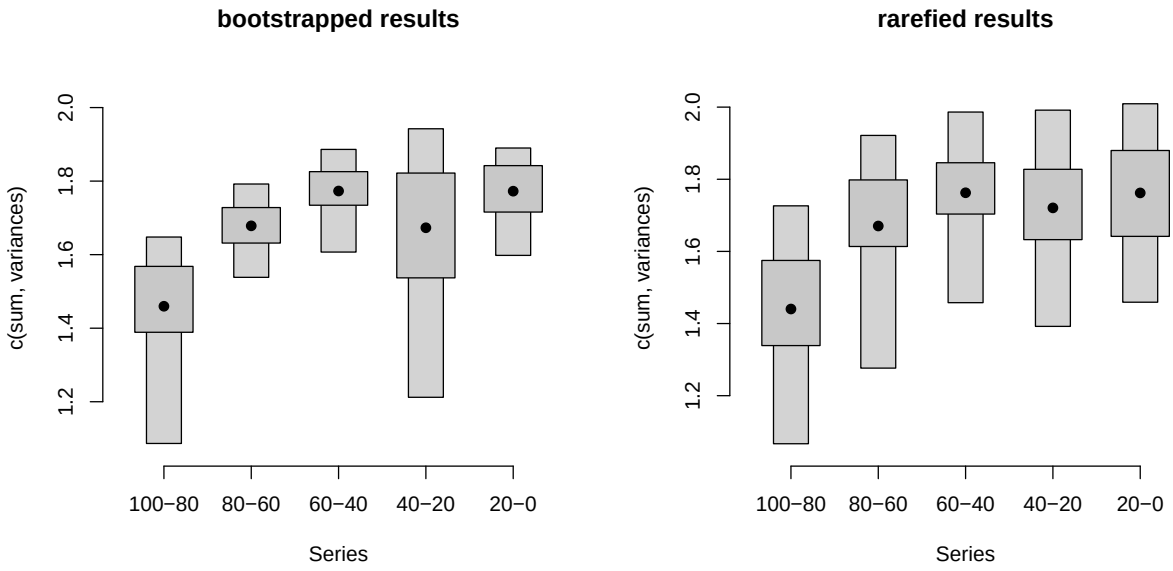
6 Plotting the results

It is sometimes easier to visualise the results in a plot than in a table. For that we can use the `plot` function to plot the `dispRity` objects.

```
## Graphical options
quartz(width = 10, height = 5) ; par(mfrow = (c(1,2)), bty = "n")

## Plotting the bootstrapped and rarefied results
plot(boot_disparity, main = "bootstrapped results")
plot(rare_disparity, main = "rarefied results")

## See more options on the plot function:
?plot.dispRity
```



As we can see in this case, the rarefaction tends to increase the confidence intervals around the mean.

7 Testing differences

Finally, to draw some conclusions from these results, we need to apply some statistical testing. As stated before, we wanted to test whether the disparity changed through time in this dataset since the last 100 million years. To do so, we can simply apply some comparisons of the means between each time-bins in a sequential manner to see whether the disparity in bin X is equal to the disparity in bin Y which is in turn equal to the disparity in bin Z, etc. Because our data is temporally auto-correlated (i.e. what happens in bin Y is dependent on what happened in bin X) and was pseudo-replicated (i.e. each bootstrap draw creates non-independent sub-samples because they are all based on the same sub-sample), we are going to apply a non-parametric mean comparison: the `wilcox.test`. Also, we are going to apply a p-value correction to avoid type I error inflation due to performing multiple test for the same hypothesis.

```
## testing the differences between bins in the bootstrapped dataset.
test.dispRity(boot_disparity, test = wilcox.test, comparison = "sequential",
    correction = "bonferroni")

##                   W      p.value
```

```
## 100-80 - 80-60  471 7.427563e-28
## 80-60 - 60-40  1562 1.798899e-16
## 60-40 - 40-20  6250 9.061511e-03
## 40-20 - 20-0   3725 7.379715e-03

## testing the differences between bins in the rarefied dataset.
test.dispRity(rare_disparity, test = wilcox.test, comparison = "sequential",
    correction = "bonferroni")

##                   W      p.value
## 100-80 - 80-60 1695 2.717026e-15
## 80-60 - 60-40  3424 4.733196e-04
## 60-40 - 40-20  5669 4.095267e-01
## 40-20 - 20-0   4254 2.740887e-01
```

Here our results show significant changes in disparity through time between all time bins during the last 100 million years. However, when looking at the rarefied results, there is no significant difference between the time bins in the Cenozoic (60-40, 40-20 and 20-0 Mya), suggesting that the differences detected in the first test might just be due to the differences in number of taxa sample (respectively 13, 6 and 10 in each time bin).

References

Beck, R. M. and M. S. Lee. 2014. Ancient dates or accelerated rates? Morphological clocks and the antiquity of placental mammals. Proceedings of the Royal Society B: Biological Sciences 281:1–10.