`dispRity` metrics

Thomas Guillerme

March 17, 2016

This vignette details how the disparity metrics are implemented in the `dispRity` package (v.0.2).

1 Disparity metrics

There are many ways for measuring disparity. In brief, disparity is a summary metric that will represent an aspect of an ordinated space (e.g. MDS, PCA, PCO, PCoA). For example, one can look at ellipsoid hyper-volume of the ordinated space (Donohue et al., 2013), the sum and the product of the ranges and variances of the ordinated space dimensions (Wills et al., 1994) or the median position of the elements in the ordinated space relative to its centroid (Guillerme and Cooper, in prep.). Of course, there are many more examples of metrics one can use for describing some aspect of the ordinated space, with some performing better than other ones at particular descriptive tasks or some being more generalist.

Because of this great diversity of metric, the package `dispRity` has not a unique way to measure disparity but rather proposes to facilitate users to define their own disparity metric that will best suit to their particular analysis. In fact, the core function of the package, `dispRity` , allows the user to define any metric in the `metric` argument. However, because of the infrastructure of the package (especially the one of the `dispRity` objects – see the `dispRity` manual), the `metric` argument has to follow certain rules:

1. it must be composed of one to three `function` objects;
2. these functions can only have as input a `matrix` or a `vector`;
3. each of these functions must be of one of the three levels described below;
4. at least one of the functions must be a level 1 function (see below).

2 The function levels

The metric functions levels determine the dimensional level of decomposition of the input `matrix`. In other words, each level designate the dimensions of the output: either three (a `matrix`); two (a `vector`); or one (a single `numeric` value) dimensions (see 1).

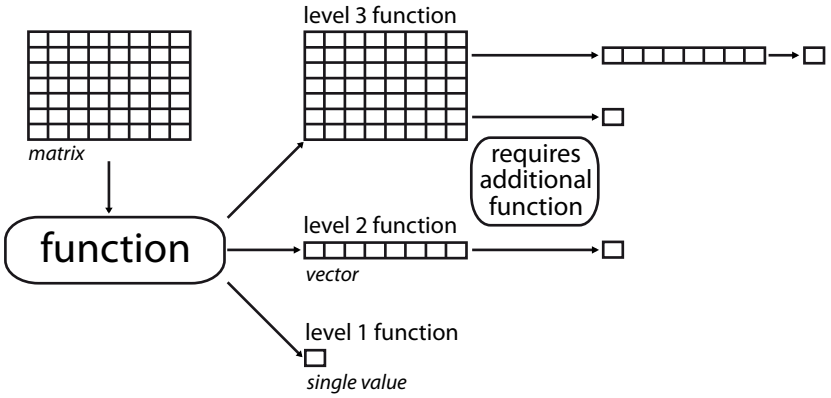

Figure 1: Illustration of the different levels of output of a function with an input `matrix`

2.1 A level 1 function

A level 1 function for example, will decompose a `matrix` or a `vector` into a single value:

```
## First, let's create a dummy matrix
dummy_matrix <- matrix(rnorm(12), 4, 3)

## Any summary metric such as mean or median are good examples of level 1
## functions as they reduce the matrix to a single dimension (i.e. one value):
mean(dummy_matrix)

## [1] -0.3103284

median(dummy_matrix)

## [1] -0.370667
```

2.2 A level 2 function

A level 2 function will decompose a `matrix` into a `vector`. Several level 2 functions are implemented in `dispRity` (see `?dispRity.metric`) such as the `variances` function that calculates the variance of each dimension of the ordinated matrix.

```
## We can define a function that outputs the products of each row of the matrix:
prod.rows <- function(matrix) apply(matrix, 1, prod)

## This function is a level 2 function as it reduces the matrix into a two
## dimensional element (a vector of values):
prod.rows(dummy_matrix)

## [1] -0.04339738  1.54026837  0.77351869  0.19139270
```

2.3 A level 3 function

Finally a level 3 function will transform the `matrix` into another `matrix`. Note that the dimension of the output `matrix` don't need to match the the input `matrix`:

```
## The var functions computes the variance/covariance matrix which will be a
## three by three matrix
var(dummy_matrix)

##              [,1]        [,2]       [,3]
## [1,]  0.56954681  0.03596576 -0.2348924
## [2,]  0.03596576  0.49175752 -0.1251905
## [3,] -0.23489237 -0.12519048  0.1284110
```

3 make.metric

Of course, functions can be more complex and involve multiple operations such as the `centroids` function (see `?dispRity.metric`) that calculates the euclidean distance between each elements and the ordinated space centroid. The `make.metric` function implemented in `dispRity` is designed to help testing and finding the level of the functions. This function tests:

1. if your function can deal with a `matrix` or a `vector` as an input;
2. which is your function's level according to it's output (level 1, 2 or 3, see above);
3. whether the function can properly be implemented in the `dispRity` function (the function is fed into a `lapply` loop.

For example, let's see if the functions described above are the right levels:

```
## First we need to load the package
library(dispRity)

## Warning in .doLoadActions(where, attach):  trying to execute load actions without 'methods'
package

## Which level is the mean function? And is able to be used in the dispRity?
make.metric(mean)

## mean outputs a single value.
## mean is detected as being a level 1 function.

## Same questions from the prod.rows function:
make.metric(prod.rows)

## prod.rows outputs a matrix object.
## prod.rows is detected as being a level 2 function.
## Additional level 1 function will be needed.

## Note that the function also tells us that we will need a level 1 function as well.
## We'll cover this below.

## Same questions for the var function:
make.metric(var)

## var outputs a matrix object.
## var is detected as being a level 3 function.
## Additional level 2 and/or level 1 function(s) will be needed.
```

4 metric argument in dispRity

Using this metric structure, on can easily use any disparity metric in the dispRity function as follows:

```
## First we need to load the data
data(BeckLee_mat50)

## Measuring disparity as the standard deviation of all the value of the
## ordinated matrix (level 1 function).
summary(dispRity(BeckLee_mat50, metric = sd))

##   series  n observed
## 1      1 50    0.201

## Measuring disparity as the standard deviation of the variance of each axis of
## the ordinated matrix (level 1 and level 2 functions).
summary(dispRity(BeckLee_mat50, metric = c(sd, variances)))

##   series  n observed
## 1      1 50    0.028

## Measuring disparity as the standard deviation of the variance of each axis of
## the variance covariance matrix (level 1, level 2 and level 3 functions).
summary(dispRity(BeckLee_mat50, metric = c(sd, variances, var)), round=10)

##   series  n     observed
## 1      1 50 0.0001025857
```

Note that the order of each function in the metric argument does not matter, the dispRity function will automatically detect the function levels (using make.metric) and apply them to the data in decreasing order (level 3 > level 2 > level 1).

```
## Disparity as the standard deviation of the variance of each axis of the
## variance covariance matrix:
disparity1 <- summary(dispRity(BeckLee_mat50, metric = c(sd, variances, var)),
  round=10)

## Same but using a different functions order for the metric argument
disparity2 <- summary(dispRity(BeckLee_mat50, metric = c(variances, sd, var)),
  round=10)

## Both ways output the same disparity values:
disparity1$observed == disparity2$observed # is TRUE

## [1] TRUE
```

References

Donohue, I., O. L. Petchey, J. M. Montoya, A. L. Jackson, L. McNally, M. Viana, K. Healy, M. Lurgi, N. E. O'Connor, and M. C. Emmerson. 2013. On the dimensionality of ecological stability. Ecology Letters 16:421–429.

Guillerme, T. and N. Cooper. in prep. Mammalian morphological diversity does not increase in response to the cretaceous-paleogene mass extinction and the extinction of the (non-avian) dinosaurs. .

Wills, M. A., D. E. G. Briggs, and R. A. Fortey. 1994. Disparity as an evolutionary index: A comparison of Cambrian and recent arthropods. Paleobiology 20:93–130.