dispRity metrics

Thomas Guillerme

October 15, 2015

This vignette details how the disparity metrics are implemented in the `dispRity` package (v.0.1.2).

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 [1], the sum and the product of the ranges and variances of the ordinated space dimensions [2] or the median position of the elements in the ordinated space relative to its centroid [3]. Of course, there is 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 on of the `dispRity` objects – see the other dispRity demos), the `metric` argument has to follow certain slightly restrictive rules described in detail below.

1. `metric` must be between one or three `function` objects
2. `metric` must intake a `matrix` or `vector` object
3. `metric` individual functions must be classified in one of the three levels
4. `metric` one of the functions must be a level 1 function

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).

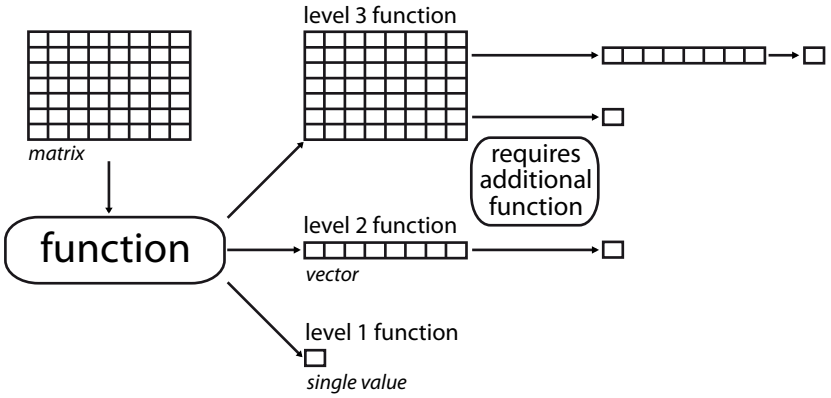

Figure 1: Illustration of the different levels of output of a function with an input `matrix`

2.1 A level 1 function

A level 1 function for example, will decompose the `matrix` (or a `vector` output from a level 2 function, see below) 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 dimensions (value):
mean(dummy_matrix)

## [1] -0.2854926

median(dummy_matrix)

## [1] -0.3311845
```

2.2 A level 2 function

A level 2 function for example, will decompose the `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 column of the ordinated matrix (or axis/dimensions):

```
## We can define a different function that outputs the products of the rows:
rows <- function(matrix) apply(matrix, 1, prod)
## Rows is a level 2 function as well (outputs a vector of 4 values):
rows(dummy_matrix)

## [1]  0.05173447  0.57024219  0.02722488 -0.04706524
```

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,]  0.38649973 -0.04075162 -0.11251152
## [2,] -0.04075162  0.60776451 -0.09710222
## [3,] -0.11251152 -0.09710222  0.06777280
```

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. Concretely, This function tests:

1. if your function can deal with a matrix as an `input`.
2. which level is your function (level 1, 2 or 3, see `?dispRity.metric`).
3. whether the function can properly be implemented in the `dispRity` function.

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

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

## Loading required package:  paleotree

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

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

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

## rows outputs a matrix object.
## 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.
## Which level is the var function? And is able to be used in the dispRity 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 algorithm always classifies them by levels and computes first level 3,
## then level 2 and finally level 1.
summary(dispRity(BeckLee_mat50, metric = c(variances, sd, var)), round=10)

##   series  n     observed
## 1      1 50 0.0001025857

## Both ways output the same disparity values.
```

References

[1] Donohue I, Petchey OL, Montoya JM, Jackson AL, McNally L, Viana M, et al. On the dimensionality of ecological stability. Ecology Letters. 2013;16(4):421–429. Available from: http://dx.doi.org/10.1111/ele.12086.

[2] Wills MA, Briggs DEG, Fortey RA. Disparity as an Evolutionary Index: A Comparison of Cambrian and Recent Arthropods. Paleobiology. 1994 04;20(2):93–130. Available from: http://www.jstor.org/stable/2401014.

[3] Guillerme T, Cooper N. Mammalian morphological diversity does not increase in response to the Cretaceous-Paleogene mass extinction and the extinction of the (non-avian) dinosaurs. in prep;Available from: https://github.com/TGuillerme/SpatioTemporal_Disparity/raw/master/Writing/STD_draft.pdf.