Short Communication

Measuring Rao’s Q diversity index from remote sensing: An open source solution

Duccio Rocchinia,⁎, Matteo Marcantonioa,b, Carlo Ricottic

a Fondazione Edmund Mach, Research and Innovation Centre, Department of Biodiversity and Molecular Ecology, Via E. Mach 1, 38010 S. Michele all’Adige (TN), Italy
b Technische Universität Berlin, Department of Landscape Architecture and Environmental Planning, Geoinformation in Environmental Planning Lab, Berlin, Germany
c Department of Environmental Biology, University of Rome “La Sapienza”, 00185 Rome, Italy

A R T I C L E   I N F O

Article history:
Received 25 May 2016
Received in revised form 21 July 2016
Accepted 22 July 2016

Keywords:
Biodiversity
Heterogeneity
Landscape metrics
Rao’s Q
Remote sensing
Spatial ecology
Shannon’s entropy

A B S T R A C T

Measuring biodiversity is a key issue in ecology to guarantee effective indicators of ecosystem health at different spatial and time scales. However, estimating biodiversity from field observations might present difficulties related to costs and time needed. Moreover, a continuous data update for biodiversity monitoring purposes might be prohibitive. From this point of view, remote sensing represents a powerful tool since it allows to cover wide areas in a relatively low amount of time. One of the most common indicators of biodiversity is Shannon’s entropy $H$, which is strictly related to environmental heterogeneity, and thus to species diversity. However, Shannon’s entropy might show drawbacks once applied to remote sensing data, since it considers relative abundances but it does not explicitly account for distances among pixels’ numerical values. In this paper we propose the use of Rao’s Q applied to remotely sensed data, providing a straightforward R-package function to calculate it in 2D systems. We will introduce the theoretical rationale behind Rao’s index and then provide applied examples based on the proposed R function.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction


However, estimating biodiversity from field data presents a number of drawbacks mainly related to time and costs, together with intrinsic difficulties to build standardized procedures for reproducible data gathering (Palmer et al., 2002).

For this purpose, using maps in a GIS environment or heterogeneity-related maps derived from remotely sensed imagery (e.g. Carranza et al., 2007) might help in finding hotspots of diversity over space and track their variation over time (Boyd and Foody, 2011), from local (Feilhauer et al., 2013) to global (Rocchini et al., 2010) spatial scales. This is true under the light of the Spectral Variation Hypothesis (Palmer et al., 2002) which states that the higher the environmental heterogeneity, the higher will be the species diversity of a certain area. The rationale under the Spectral Variation Hypothesis is that a higher spatial variability (measured by spectral diversity from remotely sensed images) is related to a higher amount of ecological niches for species living therein. Hence, measuring the heterogeneity of a landscape is critical since it is directly related to its diversity (Gillespie et al., 2008; Skidmore et al., 2015). Moreover, landscape diversity is related to the diversity at other ecosystem levels such as species diversity.

It has been demonstrated that the measure being used can lead to very different (and sometimes misleading) results. As an example, one of the mostly used diversity measures of the landscape based on spectral remotely sensed data, i.e. the Shannon’s entropy (Shannon, 1948), has a number of implicit drawbacks like: (i) the difficulty to discriminate between differences in richness or relative abundance (Nagendra, 2002) or (ii) the impossibility to consider spectral values as numbers instead of classes (Rocchini and Neteler, 2012b). Concerning the second point, Shannon’s entropy accounts for richness and relative abundance of spectral values but it does not explicitly consider the numerical magnitude (values) of pixels.

The aim of this paper is to solve the aforementioned issue, by the application of Rao’s Q to remotely sensed data, providing a...
2. Theory under the use of the Rao’s Q index

Methods for measuring landscape diversity have mostly relied on the classification of remotely sensed image. However, image classification has several drawbacks which should be seriously taken into account, e.g.: (i) the accuracy assessment should be performed in a robust manner, thus requiring time and costs overall when field assessment is involved (Foody, 2002), (ii) it is difficult to build practically sound accuracy assessment protocols (Foody, 2008), (iii) the classification should be performed only by robust algorithms avoiding as much as possible manual digitization (Burnett and Blaschke, 2003), (iv) several issues have to be bypassed when choosing pure training samples in order to avoid mixing effects (Small, 2004). Besides these technical shortcomings, classification is a subjective task in its very nature and it inevitably leads to the degradation of continuous information (Palmer et al., 2002).

Rocchini et al. (2010) summarize several approaches to measure ecosystem diversity from remotely sensed images, mainly based on the continuous variability of pixel values (e.g. original digital numbers of a satellite image). Such approaches show their full power when relying on Free and Open Source algorithms. Open Source algorithms allow indeed robustness and reproducibility thanks to the public availability of the used code (Rocchini and Neteler, 2012a).

Among the most spread diversity indexes used in ecology there is the Shannon entropy index \(H\) (Shannon, 1948). This index can be easily applied to remote sensed data. Given a certain number of reflectance values in a remotely sensed image, also referred to as digital numbers, \(H\) can be calculated as \(H = - \sum p_i \times \log(p_i)\). In this particular application \(H\) takes into account the relative proportion \(p_i\) of each reflectance value \(i\). Generalizing, \(H\) considers the equitability of the system. Furthermore, when transformed to the Pielou evenness index \(J\) (Pielou, 1969), calculated as \(J = H / H_{\text{max}}\), it shows the maximum possible diversity within the same number of reflectance values. Quoting Ricotta and Avena (2003), who provided an elegant mathematical dissertation about the Pielou index applied to both species and landscape classes:

“The normalization of \(H\) with respect to maximum entropy \((J = H / H_{\text{max}})\) is termed ‘evenness’ because it measures deviation from an even distribution of individuals amongst the \(N\) species”.

Translating the sentence from species to spectral diversity, it turns out to be:

“The normalization of \(H\) with respect to maximum entropy \((J = H / H_{\text{max}})\) is termed ‘evenness’ because it measures deviation from an even distribution of individual pixels amongst the \(N\) reflectance values.”

However, Shannon and Pielou indices only rely on the relative abundance of reflectance, not considering the numerical value of reflectance per se. Facing the problem from a mathematical point of view, let \(M\) be an image of \(3 \times 3\) pixels (indicated by \(c\) to avoid confusion with \(p\) used in this manuscript to indicate the proportion of area of each category):

\[
M = \begin{pmatrix}
C_{1,1} & C_{1,2} & C_{1,3} \\
C_{2,1} & C_{2,2} & C_{2,3} \\
C_{3,1} & C_{3,2} & C_{3,3}
\end{pmatrix}
\]  

Let \(i\) and \(j\) be two different pixel values, e.g. two Digital Numbers (DNs) of a 8 bit image with \(i \neq j\), as:

\[
M = \begin{pmatrix}
i & i & j \\
i & j & j \\
j & j & j
\end{pmatrix}
\]  

\((2)\)

In this case \(H = - \sum p \times \log(p) = -(3/9 \times \log(3/9) + 6/9 \times \log(6/9)) = 0.637\). Shannon entropy does not take into account the value of \(i\) and \(j\) but just the proportion of \(i\) and \(j\) values. Therefore, it does not discriminate among different contexts such as, (a) \(i = 1\) and \(j = 200\) or (b) \(i = 201\) and \(j = 200\).

On the contrary, Rao’s Q index does take into account \(i\) and \(j\) value by considering their pairwise distance \(d_{ij}\):

\[
Q = \sum \sum d_{ij} \times p_i \times p_j
\]  

\((3)\)

As an example, in case of (a) \(Q = 88.444\), while in case of (b) \(Q = 0.444\).

As a consequence, deriving Rao’s Q involves calculating a distance matrix \(M_d\) for all the pixel values:

\[
\begin{pmatrix}
d_{1,1} & d_{1,2} & d_{1,3} & \cdots & d_{1,n} \\
d_{2,1} & d_{2,2} & d_{2,3} & \cdots & d_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
d_{n,1} & d_{n,2} & d_{n,3} & \cdots & d_{n,n}
\end{pmatrix}
\]  

\((4)\)

or more simply \(d_{ij}\), when \(N\) pixels are considered (see also Rocchini (2007) on distance matrices in a spectral space). Thus, Rao’s Q is related to the sum of all the pixel values pairwise distances, each of which is multiplied by the relative abundance of each pair of pixels in the analysed image \(d \times (1/N^2)\). In other words, Rao’s Q is the expected difference in reflectance values between two pixels drawn randomly with replacement from the considered evaluated pixels set. The distance matrix can be built in several dimensions (layers), thus allowing to consider more than one band at a time. As a consequence Rao’s Q can be calculated in a multidimensional (multi-layers) system.

In remote sensing applications the derivation of synthetic indexes of any sort (i.e., diversity) is often performed considering small chunks of the whole image per time, commonly defined as ‘windows’ or ‘moving windows’. From now on, we will use this terminology to indicate the local space of analysis.

3. Coding Rao’s Q in R

The function spectralrao() to derive Rao’s Q, written in the R statistical language (R Core Team, 2016), is reported in Appendix 1 and stored in the GitHub repository https://github.com/mattmar/spectralr. The function accepts matrix, RasterLayer or SpatialGridDataFrame object as input (or a list of them). It can be ran with two different settings, using (i) a single matrix (mode=“classic”) or (ii) more matrices(mode=“multidimensional”) as input. Distance can be calculated relying on Euclidean, Manhattan and Canberra distances by the distance.m parameter. Appendix 2 provides a complete description of such distances, with their advantages and disadvantages, together with proper reference to previous ecological papers using them. Further, a user-defined distance matrix can be also provided through the function argument distance.m.

In this manner, it is possible to obtain \(H\) as output, together with Rao’s Q, setting the option shannon=TRUE. On the other hand, if mode=“multidimensional”, a list of matrices must be provided as input. The overall distance matrix is thus calculated in a multi-
hyper-dimensional system by using the previously stated measures through the function argument `distance.m`. Each distance is then multiplied by the inverse of the squared number of pixels in the considered moving window (as in Eq. (3)), and the Rao’s Q is finally derived by applying the summation (see Eq. (3)).

In the following section, we describe a step-by-step workflow to derive Rao’s Q for both modes, using simulated or spectral-like real matrices as input data.

Firstly, the R function can be loaded by relying on the source file available in Appendix 1, as:

```r
source("yourpath/spectralrao.r")
```

Synthetic data, i.e., two matrices `r1` an `r2`, can be generated by:

```r
### Clustered simulated spectral matrices r1
xy1 <- matrix(rnorm(25, 0, .2), ncol=5, nrow=5)
xy2 <- matrix(rnorm(25, -5, .1), ncol=5, nrow=5)
xy3 <- matrix(rnorm(25, .5, 1), ncol=5, nrow=5)
xy4 <- matrix(rnorm(25, -8, .25), ncol=5, nrow=5)
r1 <- cbind(rbind(xy1, xy2), rbind(xy3, xy4))
### Clustered simulated spectral matrices r2
xy11 <- matrix(rnorm(20, 4.5, 0.1), ncol=5, nrow=5)
xy21 <- matrix(rnorm(30, 5.5, .1), ncol=5, nrow=5)
xy31 <- matrix(rnorm(40, 6.5, .1), ncol=5, nrow=5)
xy41 <- matrix(rnorm(10, 7.0, .65), ncol=5, nrow=2)
r2 <- t(cbind(rbind(xy11, xy21), rbind(xy31, xy41)))
```

Using (i) as input data, (ii) the Euclidean distance as the metric to calculate the distance matrix and (iii) an operational moving window of 3 × 3 pixels, Rao’s Q can be derived as:

```r
ramatrix <- spectralrao(r1, node="classic",
 distance.m="euclidean", window=3, shannon=T)
```

For this particular set of data (`r1`), H’ is of low applicability (Fig. 1), due to the high heterogeneity in the input data. On the contrary, Rao’s Q meaningfully highlighted the areas with higher heterogeneity, the intersection between the simulated submatrices.

In a slightly different way, using both `r1` and `r2` as input data, we can derive the multidimensional form of Rao’s Q:

```r
ramatrix <- spectralrao(list(r1,r2), node="multidimensional", distance.m="euclidean", window=3, shannon=F)
```

When visualizing this example in Fig. 2, it is evident how Rao’s Q, interpreted in its multidimensional meaning, sintezizes the contrasting signal from two different datasets in a single index.

Additional arguments implemented in the R code (Appendix 1) are related to the tolerance of retaining NAs at the border of the images by using the argument `na.tolerance` and the resizing and centering of input data, relying on the (true or false) argument `rescale` based on commonly used mean and standard deviation subtraction techniques (see Appendix 1).

The function accepts also R “spatial objects” as input data. To illustrate this application, in the following chunk of code a June 2015 MODIS Normalized Difference Vegetation Index (NDVI) image at 0.1 degrees resolution was downloaded from the Nasa Earth Observation dataset for Europe and Rao’s Q was calculated.

Download the raster file using we get:

```r
cd ~
wget -O example_mosaic_ndvi.2015.tif http://eo.esa.gwdc.nasa.gov/sensor/Leaf/RenderData?
s=1690249&cs=rgb&format=TIFF&width=3600&height=1800
```

Setting the no data pixels as NAs and calculating Rao’s Q:

```r
ndvi2015 <- raster("~/example_mosaic.ndvi.2015.tif")
ndvi2015[ndvi2015==255]<NA
ramatrix <- spectralrao(ndvi2015, distance.m="euclidean",
 window=9, shannon=T)
```

In Fig. 3, it is apparent that H’ tends to saturate in case of high diversity since in the local 9 × 9 pixels window of analysis all the pixel values, even though similar among them, are still different. As a consequence, since H’ does not take into account their distance but only their relative abundance, its value will always approximate saturation. On the contrary, Rao’s Q overcomes this limitation by the pairwise distance term.

The output of `spectralrao()` is a function of a list of objects. The output list has dimension 1 if `shannon=FALSE` or if `mode="multidimensional"` or dimension 2 if `shannon=TRUE` and `mode="classic"`. If RasterLayer or SpatialGridDataFrame R objects are provided as input, the function output will be a list of RasterLayer object(s).

4. Discussion

In this paper, we demonstrated the potential advantages of applying Rao’s Q to calculate diversity in digital imagery, thus avoiding the non-dimensionality of other more common indices like the Shannon’s index.

Dealing with digital images, an advantage of Rao’s Q over more conventional diversity measures, is that, while the calculation of H’ relies solely on the relative proportion of the digital numbers.
While small degrees Fig. 3. In this MODIS Normalized Difference Vegetation Index (NDVI) image at 0.1 degrees resolution of June 2015, the Shannon and the Rao indices are calculated. While Shannon tends to saturate towards higher values, Rao’s Q is not affected by small differences between pairs of pixel values.

(Rao’s Q) has been extensively used in functional diversity application (Botta-Dukát, 2005; Ricotta and Moretti, 2011; Marcantonio et al., 2014). Functional ecologists make use of a wide set of functional traits (plants functional characteristics) to assess the diversity of natural systems. Rao’s Q has been shown to be a valid candidate to summarize them in a single diversity value (Botta-Dukát, 2005). However, as previously stated, this is the first application of the Rao’s Q in a 2D space with remotely sensed data.

Note here that if, for a single band, Rao’s Q is calculated using half the squared Euclidean distance \( d_{ij} = \frac{1}{2} (i - j)^2 \) the resulting index \( \frac{1}{2} \sum \sum p_i \times p_j \times (i - j)^2 \) reduces to the well-known formula of variance (expressed as the average squared difference among DN values; see Pavoine (2012)), which is routinely used in remote sensing for summarizing the spatial complexity of digital images (Rocchini et al., 2010). Accordingly, Rao’s Q can be interpreted as a multivariate generalization of the variance of a quantitative variable such as the DNs of a spectral band, thus bridging the gap between remotely sensed measures of diversity and spatial complexity.

Note also that in principle, apart from the Euclidean distance, Rao’s Q can be calculated with a plethora of different multivariate measures of dissimilarity (see Podani, 2000 for review) that may be selected according to the specific users’ needs. In this framework, due to the additive property of Q for which if \( d_{ij} = d_{[1]} + d_{[2]} \) then \( Q = Q_{[1]} + Q_{[2]} \), it is possible to calculate a number of multivariate indices of Rao’s Q based on different combinations of single-band indices. For example, given two bands U and V, with the proposed R code we can calculate the Rao’s Q for both bands separately. In this one-dimensional case, the distance used is simply the absolute difference between the DNs of each band. The resulting index values \( Q_{[U]} \) and \( Q_{[V]} \) can be then additively recombined into one single index as \( Q = Q_{[U]} + Q_{[V]} \). This is tantamount saying that, first, the univariate distances associated to the single bands are recombined into the multivariate Manhattan or city-block distance (see Podani, 2000) such that \( d_{ij} = d_{[U]} + d_{[V]} \), and next the Rao index Q is calculated directly from the multivariate Manhattan distances \( d_{ij} \).

If the Manhattan distance is divided by the number of variables such that \( d_{ij} = \frac{1}{2} d_{[U]} + \frac{1}{2} d_{[V]} \) and hence \( Q = \frac{1}{2} Q_{[U]} + \frac{1}{2} Q_{[V]} \), the so-called mean character difference or Czekanowski dissimilarity is obtained. Finally, if the simple average is substituted by the weighted average \( d_{ij} = w_{[U]} d_{[U]} + w_{[V]} d_{[V]} \) (i.e. \( Q = w_{[U]} Q_{[U]} + w_{[V]} Q_{[V]} \) with \( 0 \leq w \leq 1 \) and \( \sum w = 1 \)) we obtain a highly flexible generalization of the Czekanowski index (Pavoine et al., 2009) in
which the weights $w$ of single bands can be determined according to the reflectance properties of single bands or to the specific user’s requirements. For example, the weights can be set proportional to the range of DN values in each band.

Due to its flexibility, Rao’s $Q$ based on the aforementioned multivariate distances may be helpful to optimize the relationship between biodiversity values recorded from remote sensors and species inventories recorded from field observations.

Acknowledgements

We are grateful to the handling Editor Prof. Giovanni Zurlini and two anonymous referees for comments on a previous version of this manuscript.

DR was partially funded by: (i) the EU BON (Building the European Biodiversity Observation Network) project, funded by the European Union under the 7th Framework Programme, Contract No. 308454, (ii) by the ERA-Net BiodiveRSA, with the national funders ANR, BelSPO and DFG, part of the 2012–2013 BiodiveRSA call for research proposals, (iii) the EU-LIFE project LIFE14ENV/IT/000514 FutureForCoppices.

The PhD Scholarship of MM is supported by FIRS > T (FEM International Research School Trentino).

Appendix A. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ecolind.2016.07.039.

References


Spectral Rao Code to calculate Rao's quadratic entropy on a
numeric matrix, RasterLayer object (or lists)
using a moving window. The function also calculates
Shannon diversity index.
Rao's Q Min = 0, if all pixel classes have
distance 0. If the chosen distance ranges between
0 and 1, Rao's Max = 1-1/S (Simpson Diversity,
where S is pixel classes).
Latest update: 21th July 2016
Link: https://github.com/mattmar/spectralrao/blob/master/spectralrao.r
This code is a Supplementary material of the paper:
Measuring Rao’s Q diversity index from remote sensing: an open source
solution.
Ecological Indicators.

Function
spectralrao<-function(matrix, distance_m="euclidean", p=NULL, window=9,
mode="classic", shannon=FALSE, rescale=FALSE, na.tolerance=0.0,
debugging=FALSE) {
# Load required packages
# Initial checks
if( !(is(matrix,"matrix") | is(matrix,"SpatialGridDataFrame") |
is(matrix,"RasterLayer") | is(matrix,"list")) ) {
  stop("Not a valid input object.")
}
# Change input matrix/ces names
if( is(matrix,"SpatialGridDataFrame") ) {
  matrix <- raster(matrix)
} else if( is(matrix,"matrix") | is(matrix,"RasterLayer") ) {
  rasterm<-matrix
} else if( is(matrix,"list") ) {
  rasterm<-matrix[[1]]
}
# Deal with matrix and RasterLayer in a different way
if( is(matrix[[1]],"RasterLayer") ) {
  if( mode="classic" ){
    rasterm<-round(as.matrix(rasterm),3)
    message("RasterLayer ok: Rao and Shannon output matrices
will be returned")
  } else if(mode="multidimension" & shannon==FALSE){
    message("RasterLayer ok: RaoQ will be returned")
  } else if(mode="multidimension" & shannon==TRUE){
    stop("Matrix check failed: RaoQ will not compatible, set shannon=FALSE")
  } else if( is(matrix,"matrix") | is(matrix,"list") ) {
    if( mode="classic" ){
      message("Matrix check ok: Rao and Shannon output matrices
will be returned")
    } else if( mode="multidimension" ){
      message("Multidimension not compatible, set mode=classic")
    }
  } else {
    stop("Invalid input object")
  }
}
if (mode == "multidimension" & shannon == FALSE) {
    message("Matrix check ok: \nA matrix with multidimension RaoQ will be returned")
} else if (mode == "multidimension" & shannon == TRUE) {
    stop("Matrix check failed: \nMultidimension and Shannon not compatible, set shannon=FALSE")
} else {
    stop("Matrix check failed: \nNot a valid input, please provide a matrix, list or RasterLayer object")
}

## Derive operational moving window

if (window %% 2 == 1) {
    w <- (window - 1) / 2
} else {
    stop("Moving window size must be odd")
}

## Output matrices preparation

raoqe <- matrix(rep(NA, dim(rasterm)[1] * dim(rasterm)[2]), nrow = dim(rasterm)[1], ncol = dim(rasterm)[2])
shannond <- matrix(rep(NA, dim(rasterm)[1] * dim(rasterm)[2]), nrow = dim(rasterm)[1], ncol = dim(rasterm)[2])

## If classic RaoQ

if (mode == "classic") {
    # Reshape values
    values <- as.numeric(as.factor(rasterm))
    rasterm_1 <- matrix(data = values, nrow = dim(rasterm)[1], ncol = dim(rasterm)[2])
    hor <- matrix(NA, ncol = dim(rasterm)[2], nrow = w)
    ver <- matrix(NA, ncol = w, nrow = dim(rasterm)[1] + w * 2)
    trasterm <- cbind(ver, rbind(hor, rasterm_1, hor), ver)
    classes <- levels(as.factor(rasterm))
    d1 <- dist(classes, method = distance_m)
    for (cl in (1 + w):(dim(rasterm)[2] + w)) {
        for (rw in (1 + w):(dim(rasterm)[1] + w)) {
            if (length(!which(!trasterm[c(rw - w):c(rw + w), c(cl - w):c(cl + w)] %in% NA)) < window^2 - ((window^2) * na.tolerance)) {
                raoqe[rw - w, cl - w] <- NA
            } else {
                tw <- summary(as.factor(trasterm[c(rw - w):c(rw + w), c(cl - w):c(cl + w)]), maxsum = 10000)
                if ("NA's" %in% names(tw)) {
                    tw <- tw[-length(tw)]
                }
                if (debugging) {
                    message("Working on co ords ", rw, ",", cl, ". classes length: ", length(tw), ", window size=", window)
                }
                tw_labels <- names(tw)
                tw_values <- as.vector(tw)
                if (length(tw_values) == 1) {
                    raoqe[rw - w, cl - w] <- NA
                }
            }
        }
    }
}

else {
  p <- tw_values/sum(tw_values)
  p1 <- diag(0,length(tw_values))
  p1[upper.tri(p1)] <- c(combn(p,m=2,FUN=prod))
  p1[lower.tri(p1)] <- c(combn(p,m=2,FUN=prod))
  d2 <- unname(as.matrix(d1)[as.numeric(tw_labels), as.numeric(tw_labels)])
  raoqe[rw-w,cl-w]<-sum(p1*d2)
}

} # End classic RaoQ
#----------------------------------------------------#
} else if(mode=="multidimension"){
  # If multidimensional RaoQ
  # Check if there are NAs in the matrices
  if ( is(rasterm,"RasterLayer") ){
    if(any(sapply(lapply(matrix, function(x) {as.matrix(x)}),
           is.na)==TRUE))
      message("\n Warning: One or more RasterLayers contain NA
which will be threated as 0")
  } else if ( is(rasterm,"matrix") ){
    if(any(sapply(matrix, is.na)==TRUE))
      message("\n Warning: One or more matrices contain NA which
will be threated as 0")
  }
  # #Check whether the distance is valid or not #
  if( distance_m="euclidean" | distance_m="manhattan" |
      distance_m="canberra" ) {
      #Define the distance functions
      # euclidean
      multieuclidean <- function(x) {
        tmp <- lapply(x, function(y) {
            (y[[1]]-y[[2]])^2
        })
        return(sqrt(Reduce(`+`,tmp)))
      }
      # manhattan
      multimanhattan <- function(x) {
        tmp <- lapply(x, function(y) {
            abs(y[[1]]-y[[2]])
        })
        return(Reduce(`+`,tmp))
      }
      # canberra
      multicanberra <- function(x) {
        tmp <- lapply(x, function(y) {
            abs(y[[1]] - y[[2]]) / (abs(y[[1]]) + abs(y[[2]]))
        })
        return(Reduce(`+`,tmp))
      }
      # Decide what function to use #
      if( distance_m="euclidean" ) {

distancef <- get("multieuclidean")
} else if(distance_m=="manhattan") {
  distancef <- get("multimanhattan")
} else if(distance_m=="canberra") {
  distancef <- get("multicanberra")
}
} else {
  stop("Distance function not defined for multidimensional Rao's Q; please chose among euclidean, manhattan or canberra")
}
#
##Reshape values
#
  vls<-lapply(matrix, function(x) {as.matrix(x)})
#
##Rescale and add fake columns and rows for moving w
#
  hor<-matrix(NA,ncol=dim(vls[[1]])[2],nrow=w)
  ver<-matrix(NA,ncol=w,nrow=dim(vls[[1]])[1]+w*2)
  if(rescale) {
    trastersm<-lapply(vls, function(x) {
      t1 <- raster::scale(raster(cbind(ver,rbind(hor,x,hor),ver)))
      t2 <- as.matrix(t1)
      return(t2)
    })
  } else {
    trastersm<-lapply(vls, function(x) {
      cbind(ver,rbind(hor,x,hor),ver)
    })
  }
#
##Loop over all the pixels in the matrices
#
  if( (ncol(vls[[1]])*nrow(vls[[1]]))> 10000) {
    message("\n Warning: ",ncol(vls[[1]])*nrow(vls[[1]])*length(vls), " cells to be processed, may take some time... \n")
  }
  for (cl in (1+w):(dim(vls[[1]])[2]+w)) {
    for(rw in (1+w):(dim(vls[[1]])[1]+w)) {
      if( length(!which(!trastersm[[1]][c(rw-w):c(rw+w),c(cl-w):c(cl+w)]%in%NA)) < window^2-((window^2)*na.tolerance) ) {
        raoqe[rw-w,cl-w] <- NA
      } else {
        tw<-lapply(trastersm, function(x) { x[(rw-w):(rw+w),(cl-w):(cl+w)]
      }
    }
  }
##Vectorize the matrices in the list and calculate
#the among matrix pairwise distances

  lv <- lapply(tw, function(x) {as.vector(t(x))})
  vcomb <- combn(length(lv[[1]]),2)
  vout <- c()
  for(p in 1:ncol(vcomb) ) {
    lpair <- lapply(lv, function(chi) { c(chi[vcomb[1,p]],chi[vcomb[2,p]])
    vout[p] <- distancef(lpair)
    }
}

raoqe[rw-w,cl-w] <- sum(rep(vout,2) * 
(1/(window)^4),na.rm=TRUE)
}
}
)# end multidimensional RaoQ
#----------------------------------------------------#
##ShannonD
#
if(shannon){
  #Reshape values
  values<-as.numeric(as.factor(rasterm))
  rasterm_1<-matrix(data=values,nrow=dim(rasterm)[1],ncol=dim(rasterm)[2])
  #Add fake columns and rows for moving window
  #
  hor<-matrix(NA,ncol=dim(rasterm)[2],nrow=w)
  ver<-matrix(NA,ncol=w,nrow=dim(rasterm)[1]+w*2)
  trasterm<-cbind(ver,rbind(hor,rasterm_1,hor),ver)
  #
  #Loop over all the pixels
  #
  for (cl in (1+w):(dim(rasterm)[2]+w)) {
    for(rw in (1+w):(dim(rasterm)[1]+w)) {
      if( length(!which(!trasterm[c(rw-w):c(rw+w),c(cl-w):c(cl+w)]%in%NA)) < window^2-((window^2)*na.tolerance) ) {
        shannond[rw-w,cl-w]<-NA
      } else {
        tw<-summary(as.factor(trasterm[c(rw-w):c(rw+w),c(cl-w):c(cl+w)]))
        if( "NA's"%in%names(tw) ) {
          tw<-tw[-length(tw)]
        }
        tw_values<-as.vector(tw)
        p<-tw_values/sum(tw_values)
        p_log<-log(p)
        shannond[rw-w,cl-w]<-(-(sum(p*p_log)))
      }
    }
  } # End ShannonD
}
#----------------------------------------------------#
##Return the output
#
if( is(rasterm,"RasterLayer") ) {
  if( shannon ) {
    #Rasterize the matrices if matrix==raster
    rastertemp <- stack(raster(raoqe,
    template=matrix),raster(shannond, template=raster))
  } else if(shannon==FALSE){
    rastertemp <- raster(raoqe, template=rasterm)
  }
}
##Return multiple outputs
#
if( is(rasterm,"RasterLayer") ) {
return(rastertemp)
} else if(!is(rasterm,"RasterLayer") & shannon) {
    return(list(raoq,shannon))
} else if(!is(rasterm,"RasterLayer") & shannon==FALSE) {
    return(list(raoq))
}
Appendix 2 - Description of the most appropriate multivariate distances for remotely sensed data contained in the R function spectralrao()

Duccio Rocchini$^{1,*}$, Matteo Marcantonio$^{1,*}$, and Carlo Ricotta$^2$

1 Fondazione Edmund Mach, Research and Innovation Centre, Department of Biodiversity and Molecular Ecology, Via E. Mach 1, 38010 S. Michele all’Adige (TN), Italy, corresponding author: ducciorocchini@gmail.com, duccio.rocchini@fmach.it
2 Department of Environmental Biology, University of Rome "La Sapienza", 00185 Rome, Italy
* authors contributed equally to the ms, corresponding author: ducciorocchini@gmail.com, duccio.rocchini@fmach.it

July 21, 2016

<table>
<thead>
<tr>
<th>Distance</th>
<th>Description</th>
<th>Ecological references</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>This function reflects our intuitive feeling about distances, thus representing the standard reference for any other dissimilarity function (Podani, 2000). For $N$ bands, the Euclidean distance can be formulated as: $D_E : (x, y) \mapsto \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$. For the calculation of $D$, all differences are squared. Therefore, the Euclidean distance is most sensitive to large differences. The minimum value of $D$ is 0, and there is no upper bound.</td>
<td>Gower and Legendre (1986); Legendre and Legendre (1998); Podani (2000).</td>
</tr>
</tbody>
</table>
**Manhattan**

The Manhattan distance is the sum of absolute differences:

\[ D_M : (x, y) \mapsto \sum_{i=1}^{N} |x_i - y_i|. \]

In the univariate case (i.e. for one single remotely sensed band) the Manhattan distance is identical to the Euclidean distance. Like for the Euclidean distance, the minimum value of D is 0, and there is no upper bound.

The Canberra distance is derived from the Manhattan distance by standardizing separately the absolute differences of each band with the sum of both values, such that:

\[ D_C : (x, y) \mapsto \sum_{i=1}^{N} \frac{|x_i - y_i|}{|x_i| + |y_i|}. \]

For N bands, the range of the Canberra distance is \([0, N]\). Hence, division by N gives a standardized coefficient in the range \([0,1]\).

**Canberra**

Legendre and Legendre (1998); Podani (2000).

Firstly introduced by Lance and Williams (1966), its use in ecology is discussed by Warton et al. (2012).

**References**


