ecostructure - Grade of Membership Model and Visualization for ecological species abundance data

September 21, 2017

1 Introduction

The **ecostructure** package is an R package that replicates the statistical analysis in this paper, but its toolbox of functions is generic enough in handling and analyzing other species abundance data. The package provides functions for fitting the Grade of Membership (GoM) model, along with the visualization of model fit using Block Structure plot **????**. The package comes with the raw taxonomic data saved as an ExpressionSet object and provides a pipeline for reading and processing counts data corresponding to different dimensions of diversity, e.g. - phylogenetic, regional and functional, which serve as readymade input for the GoM model. This package is an upgraded version of the CountClust package due to Dey et al **??** for fitting GoM models on RNA-seq data

2 Installation

The package is available on Github and can be installed as follows

```
library(devtools)
install_github("kkdey/ecostructure")
```

Load the package as

```
library(ecostructure)

## Warning:  replacing previous import ’ape::rotate’ by ’raster::rotate’ when loading
’ecostructure’

## Warning:  replacing previous import ’ape::zoom’ by ’raster::zoom’ when loading ’ecostruc

## Warning:  replacing previous import ’raster::density’ by ’stats::density’ when loading
’ecostructure’

library(Biobase)
```

to use the GoM model, the user needs to install the **maptpx** package

```
library(devtools)
install_github("kkdey/maptpx")
```

3 Data Preparation

One can load the taxonomic data, together with the grid metadata ans species metadata as an ExpressionSet object as follows

```
data <- get(load(system.file("extdata", "HimalayanBirdsData.rda",
                             package = "ecostructure")))
taxonomic_counts <- t(exprs(data))
taxonomic_counts[1:5,1:5]
```

```
##    Macropygia_unchall Streptopelia_chinensis Streptopelia_senegalensis
## A2                  0                      0                         0
## A3                  0                      0                         0
## A4                  0                      0                         0
## A6                  0                      0                         0
## A7                  0                      0                         0
##    Columba_pulchricollis Streptopelia_orientalis
## A2                     0                       0
## A3                     0                       0
## A4                     0                       0
## A6                     0                       0
## A7                     2                       0
```

The corresponding grid metadata can be read as

```
grid_metadata <- pData(phenoData(data))
head(grid_metadata)
```

```
##    Elevation North East WorE
## A2       198  27.0 92.9    E
## A3       734  27.0 92.4    E
## A4      1243  27.0 92.4    E
## A6      2629  27.1 92.5    E
## A7      2340  27.1 92.4    E
## A8       300  27.0 93.0    E
```

The species metadata can be read as follows

```
species_metadata <- pData(featureData(data))
head(species_metadata)
```

```
##                           bill_length bill_width bill_depth wing tarsus  mass
## Macropygia_unchall               11.08         4.26         4.97  198   26.3 168.0
## Streptopelia_chinensis           10.77         3.50         3.87  140   22.6 159.0
## Streptopelia_senegalensis         9.23         2.88         3.27  130   20.4  83.9
## Columba_pulchricollis            12.98         5.59         5.68  203   25.4 330.0
## Streptopelia_orientalis          10.88         4.09         3.90  192   25.5 233.0
## Chalcophaps_indica               11.77         3.61         4.42  151   26.3 121.0
```

Along with the taxonomic data and metadata, the package provides the phylogenetic tree data for the bird species as a **.tre** file that can be loaded as follows, using the package **ape**.

```
phylo_tree <- ape::read.tree(system.file("extdata", "AllHim_Mar_27_2015.tre",
                             package = "ecostructure"))

## Warning in ape::read.tree(system.file("extdata", "AllHim_Mar_27_2015.tre", :  empty
character string.

phylo_tree

## NULL
```

The shape files for the regional motif analysis can be loaded as follows

```
shp_file <- ape::read.tree(system.file("extdata", "AllHim_Mar_27_2015.tre",
                             package = "ecostructure"))

## Warning in ape::read.tree(system.file("extdata", "AllHim_Mar_27_2015.tre", :  empty
character string.

shp_file

## NULL
```

4 Grade of Membership Model and Visualization

Here we illustrate how one can fit the Grade of Membership model and perform the visualization of the model fit using the Block Structure plot. Here we present a case study with number of clusters chosen to be between 2 and 4.

```
elevation_metadata=grid_metadata$Elevation;
east_west_dir = grid_metadata$WorE;
gom_fit <- CountClust::FitGoM(taxonomic_counts, K=2:4, tol=0.1)

##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 17671.9, 31.7, 19.4, 103.3, 3.1, 1.5, 0.3, 0.1, done.
## log BF( 2 ) = 1949.26
##
```

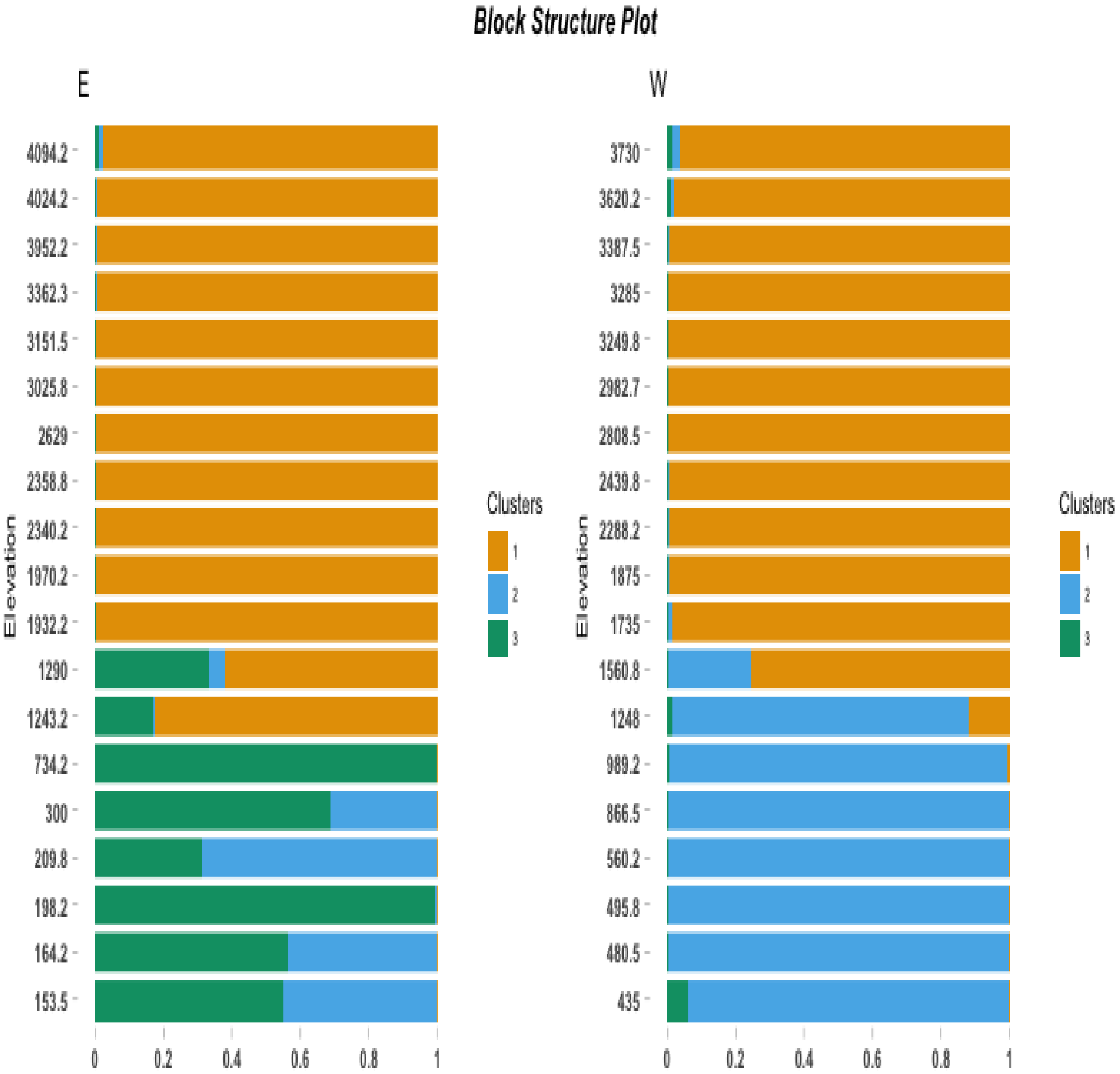
```
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 3
## log posterior increase: 4499, 13, 5.9, 1.6, done.
## log BF( 3 ) = 2108.91
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 4
## log posterior increase: 3991.8, 20.1, 121.7, 0.3, 0.1, done.
## log BF( 4 ) = 2273.93
```

`gom_fit` is a list of size 3, with each component representing the model fit for the cluster k, varying from 2 to 4. The two main components of the model fit are the membership proportion matrix ω, given by `gom_fit[[k]]$omega` and the motif matrix `gom_fit[[k]]$theta`. Examples for $K = 2$ are

```
omega <- gom_fit[[2]]$omega
head(omega)

##         topic
## document        1        2        3
##       A2 8.67e-05 9.98e-01 0.001983
##       A3 4.71e-04 3.89e-01 0.610313
##       A4 4.91e-04 2.94e-04 0.999216
##       A6 1.00e+00 1.46e-04 0.000228
##       A7 1.00e+00 5.68e-05 0.000163
##       A8 1.03e-04 9.96e-01 0.003641

rowSums(omega)

## A2 A3 A4 A6 A7 A8 B1 B2 B3 B4 B5 D1 D3 G1 J1 J2 J4 J5 J6 K1 K2 K4 K5 K6 L1 M1
##  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## M2 M3 M4 N1 N2 N3 S1 U3 U4 MA U1 U2
##  1  1  1  1  1  1  1  1  1  1  1  1

theta <- gom_fit[[2]]$theta
head(theta)

##                             topic
## phrase                             1        2        3
##   Macropygia_unchall        1.53e-03 1.02e-06 8.70e-07
##   Streptopelia_chinensis    4.82e-07 1.02e-06 1.10e-02
##   Streptopelia_senegalensis 4.20e-07 1.01e-06 4.73e-03
##   Columba_pulchricollis     7.65e-04 1.02e-06 8.67e-07
##   Streptopelia_orientalis   6.22e-03 1.03e-06 6.08e-03
##   Chalcophaps_indica        4.21e-07 3.73e-03 1.54e-03

colSums(theta)

## 1 2 3
## 1 1 1
```

Using the grid metadata, we provide a Block Structure Plot representation of the membership proportion matrix. In a block Structure plot representation, one metadata is used for forming blocks (here the East/West direction) and in each block, the the samples (along the rows of the Structure plot) are arranged by a second metadata (say Elevation).

```
BlockStructure(omega, blocker_metadata = east_west_dir,
               order_metadata = elevation_metadata,
               yaxis_label = "Elevation",
               levels_decreasing = FALSE)
```



ecostructure provides tools to compare the GoM model fit on the data with respect to null model using the `nullmodel_GoM` function.

```
nullmodel_GoM(taxonomic_counts, K=2,
              tol=500, null.model="frequency",
              iter_randomized=5, plot=FALSE)
```

```
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 3212.3, done.
## log BF( 2 ) = 356.32
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2739.9, done.
## log BF( 2 ) = 231.69
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2939, done.
## log BF( 2 ) = 334.75
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 3059.7, done.
## log BF( 2 ) = 287.24
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 3018.7, done.
## log BF( 2 ) = 419.97
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2665.4, done.
## log BF( 2 ) = 1227.52
## $GoMBF.obs
## [1] 16309
##
## $GoMBF.rand
## [1] 12658 13513 13719 13054 13470
##
## $pval
## [1] 0
```

The function returns for a fixed K, the observed Bayes factor for the GoM model fit on the actual counts data, as well as the Bayes factor from applying GoM model on `iter_randomized` many counts matrices generated under a specified null model. It also provdies a p-value of the observed Bayes factor with respect to the distribution of the Bayes factors from GoM on null model generated matrices.

In the above example **frequency** based null model was used. The other options are **richness**, **trialswap** and **independentswap**. Ideally if the observed Bayes factor should be higher than the Bayes factors from null model generated counts data.

5 Processing motif data

In this section, we demonstrate how **ecostructure** can be used to process grid level counts data corresponding to different axes of diversity, that may range from being functional to regional to phylogenetic.

5.1 phylogenetic motif

For building the counts data corresponding to phylogenetic diversity, we provide a function, `collapse_counts_by_p` to collapse the taxa in the taxonomic counts data based on the phylogenetic similarity profile of the species. The function reads in the taxonomic counts data and the phylogenetic tree data and a user defined cut off at which to slice the tree and collapse the taxa in each branch into a single phylogenetic unit.

```
tree <- ape::read.tree(system.file("extdata",
                                   "grids_tree_3_10_16.tre",
                                    package = "ecostructure"))
phylo_counts <- collapse_counts_by_phylo(taxonomic_counts,
                                         tree, collapse_at = 10)
dim(phylo_counts)
```

5.2 regional motif

For the regional profile, **ecostructure** allows the user to create Global assemblage dispersion fields and build maps data from those fields. To create the assemblage dispersion fields, the user would require to obatin the shapefiles (*.shp* files) for the all the species in the observed data.

One source of obtaining the *.shp* files for mapping to geographic boundaries is from the Natural Earth webpage (www.naturalearthdata.com/downloads). For our Himalyan birds data, we obatined the *.shp* files from BirdLife International (www.birdlife.org).

The user can put all the *.shp* files in the `all_bird_shapefiles()` and then using this folder of shapefiles and the local taxonomic data, the user can create the global assemblage dispersion fields as follows

```
disp <- CreateGlobalDispersionFields(taxonomic_counts,
              shapefiles_dir = "all_bird_sjapefiles/")
```

```
dispersion.field <- readRDS(system.file("extdata",
                    "dispersion_field_list.rds", package = "ecostructure"))
proj <- CRS(' +proj=longlat +ellps=WGS84')
global_shapefile <- readShapeLines(system.file("extdata",
         "ne_50m_admin_0_countries/ne_50m_admin_0_countries.shp",
          package = "ecostructure"), proj4string=proj)
par(mfrow)
maps <- CreateMapsFromDispersionFields(dispersion.field, global_shapefile)
```

The function returns a list of map plots with as many elements as the number of sites. We can see the maps as follows

```
maps[[1]]
```

Finally, the user can generate the counts data corressponding to regional diversity using the `DispersionFieldTocou` function

```
regional_counts <- DispersionFieldToCounts(dispersion.field)
```

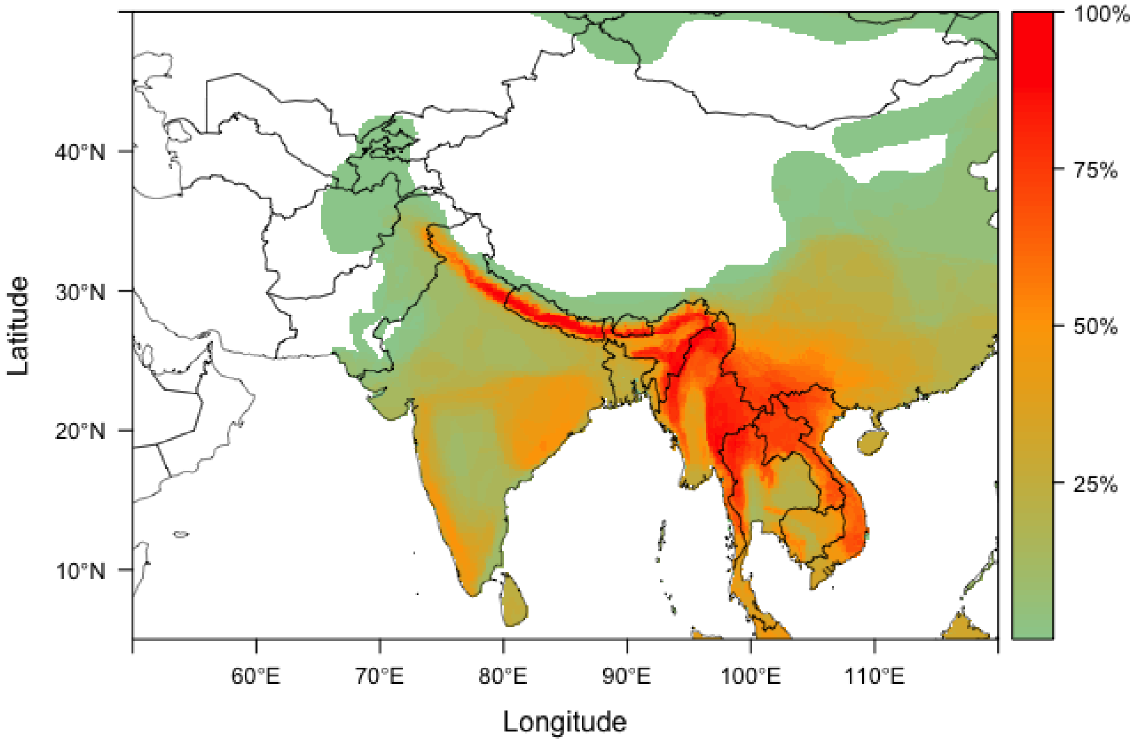
```
dim(regional_counts)

regional_counts[1:5,1:5]
```

5.3 functional motif

In order to build counts data corresponding to functional diversity, we order the species based on some ordering metadata (like bill shape, size of the bird etc in our example). But there will be many zeros in the matrix as the species abundance data is sparse. To effectively account for the functional diversity and take into account the relatedness among the bird species, the zeros are filled in by kriging based on the species with non-zero abundance. We use the function `krige_counts`.

```
func_counts <- krige_counts(taxonomic_counts,
        order = species_metadata$bill_length,
            krige.control = list(cov.mod = "whitmat", sill=0.5, smooth=.01))
```

```
dim(func_counts)
head(func_counts[,1:5])
```



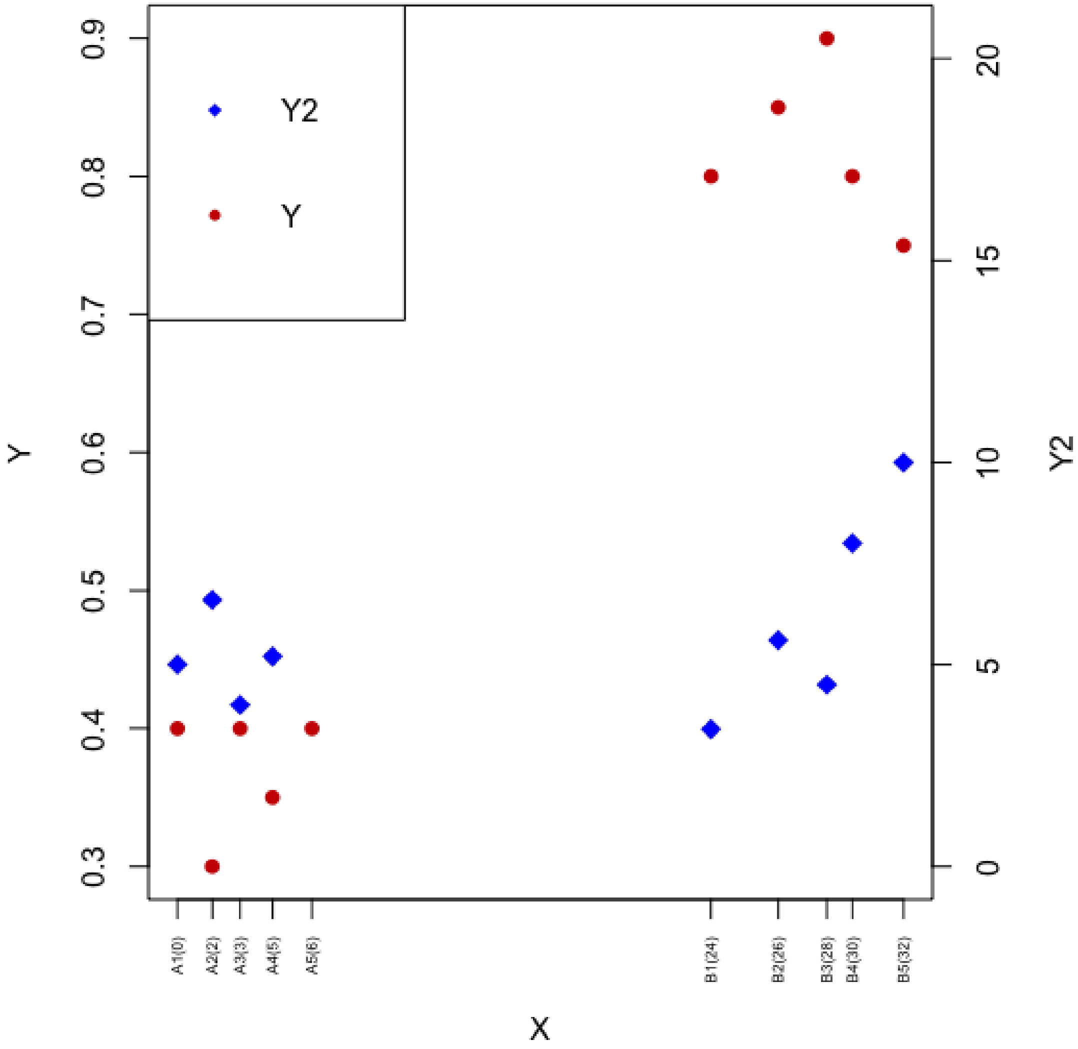
6 Extras

ecostructure provides additional functions for plotting a variable of interest (which could be the grades of membership or the motif pattern) against a metadata and a diversity measure or against two metadata in three way scatter plot functions.

An example of plotting a variable against two metadata are as follows.

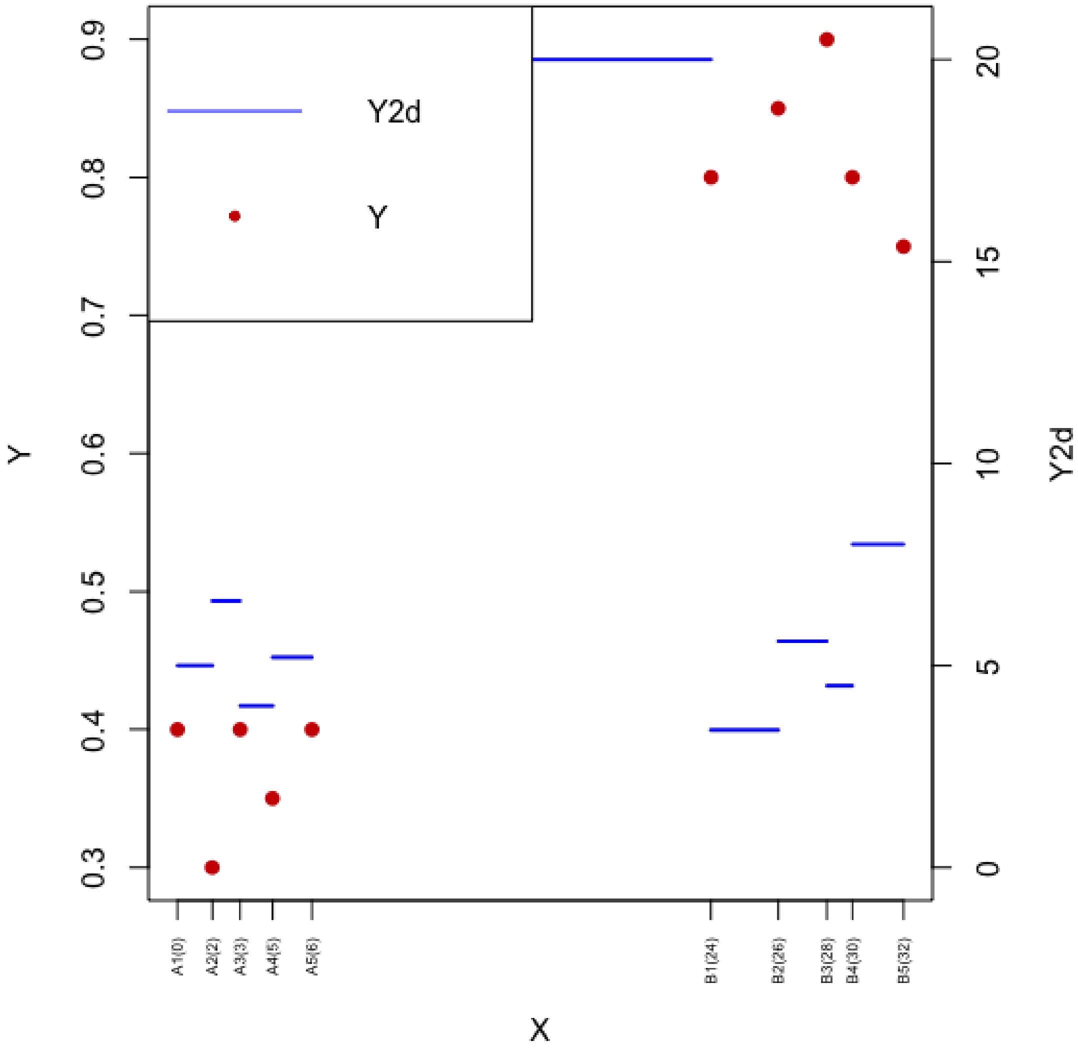
```
annotation = data.frame(x_names = c(paste0("A",1:5), paste0("B",1:5)),
x = c(0.5,2.0, 3.2, 4.6, 6.3,  23.5, 26.4, 28.5, 29.6, 31.8),
y1 = c(0.4, 0.3, 0.4, 0.35, 0.4, 0.8, 0.85, 0.9, 0.8, 0.75),
y2 =c(5, 6.6, 4, 5.2, 20, 3.4, 5.6, 4.5, 8, 10))

head(annotation)
```

```
##   x_names    x   y1   y2
## 1      A1  0.5 0.40  5.0
## 2      A2  2.0 0.30  6.6
## 3      A3  3.2 0.40  4.0
## 4      A4  4.6 0.35  5.2
## 5      A5  6.3 0.40 20.0
## 6      B1 23.5 0.80  3.4

topic_meta_meta(annotation)
```



An example of plotting a variable against two diversity measures as follows.

```
annotation = data.frame(x_names = c(paste0("A",1:5), paste0("B",1:5)),
    x = c(0.5,2.0, 3.2, 4.6, 6.3,  23.5, 26.4, 28.5, 29.6, 31.8),
    y1 = c(0.4, 0.3, 0.4, 0.35, 0.4, 0.8, 0.85, 0.9, 0.8, 0.75),
    y2d =c(5, 6.6, 4, 5.2, 20, 3.4, 5.6, 4.5, 8, 10))

head(annotation)

##   x_names    x   y1  y2d
## 1      A1  0.5 0.40  5.0
## 2      A2  2.0 0.30  6.6
## 3      A3  3.2 0.40  4.0
## 4      A4  4.6 0.35  5.2
## 5      A5  6.3 0.40 20.0
## 6      B1 23.5 0.80  3.4

topic_meta_diversity(annotation)
```


Y2d
Y
Y
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Y2d
0
5
10
15
20
A1(0)
A2(2)
A3(3)
A4(5)
A5(6)
B1(24)
B2(26)
B3(28)
B4(30)
B5(32)
X