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

September 21, 2017

1 Introduction

The **ecostructure** R package is primarily aimed at providing tools and functions to replicate the statistical analysis in this paper. However we note that its toolbox is generic enough in handling and analyzing other species abundance data. We provide the bird species abundance data along with the relevant sample metadata and species metadata as an ExpressionSet object in this package. We provide a pipeline for reading and processing the data and the metadata and also to extract or process the data corresponding to different axes of diversity - for example, phylogenetic, regional and trait-based. We discuss how the Grade of Membership (GoM) model can be fitted to the counts data and the results from the fit can be viewed using the Block Structure Plot representation, which is analogous to the STRUCTURE plot visualization in [**?**] [**?**], but can account for ordering and blocking metadata. This package is an upgraded version of the CountClust Bioconductor package due to Dey et al [**?**] that is better suited at processing and analyzing the structure of ecological abundance data.

2 Installation

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

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

Load the package as

```
library(ecostructure)
library(Biobase)
```

For fitting the Grade of Membership model, we recommend the user to install the latest version of the **maptpx** package as follows.

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

3 Data Processing

The bird taxonomic abundance data for 304 bird species across 38 Himalayan forest patches, together with the grid metadata and species metadata, are saved as an ExpressionSet object which the user can read into R 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, comprising of the elevation, latitude and longitude information of the forest patches or grids, can be read as follows

```
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
```

Finally, the species metadata, comprising of the bill traits, wing size, tarsus and mass of the birds, 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
```

Besides with the taxonomic data and metadata, we provide 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"))
```

We also provide the shapefiles for the regional motif analysis.

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

4 Grade of Membership Model and Visualization

Here we illustrate how one can fit the Grade of Membership model on the taxonomic data processed above and perform the visualization of the model fit using the Block Structure plot. Here we perform the fit for a number of clusters varying from 2 to 4.

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

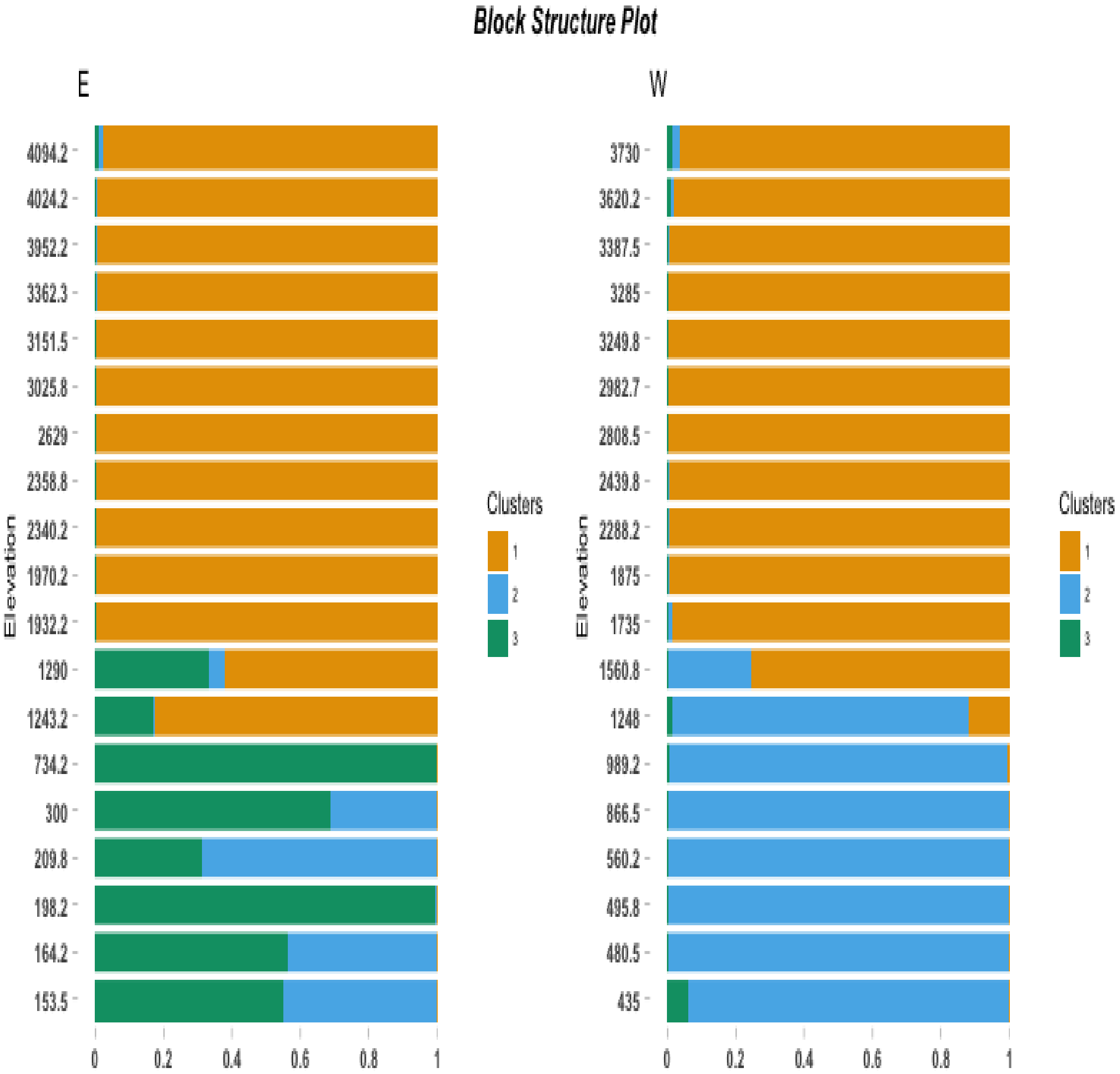
```
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2484.4, 40.8, 27.4, 0.4, 1.4, 0.6, 5.4, 14.7, done.
## log BF( 2 ) = 2181.05
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 3
## log posterior increase: 4236.5, 16.5, 10.9, 3.3, 3.7, 1.1, 9.7, 11.8, 17.9, 3.2, 0.5, do
## log BF( 3 ) = 2569.1
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 4
## log posterior increase: 3864.6, 18, 13.8, 1, done.
## log BF( 4 ) = 2124.78
```

`gom_fit` is a list of size 3, with each component representing the model fit for the cluster k, where k varies from 2 to 4. The two main components of the model fit are the cluster membership proportion matrix ω, given by `gom_fit[[k]]$omega` and the cluster motif matrix `gom_fit[[k]]$theta`, which is a description of the cluster using species compositions. We illustrate the results for $K = 2$.

```
############ membership proportion matrix (omega)  #################

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

##         topic
## document        1        2        3
##       A2 7.97e-05 1.00e+00 1.04e-04
##       A3 7.87e-05 9.99e-01 6.92e-04
##       A4 1.60e-04 9.98e-01 1.88e-03
##       A6 4.36e-04 1.24e-04 9.99e-01
##       A7 3.26e-04 9.73e-05 1.00e+00
##       A8 1.39e-04 1.00e+00 9.36e-05

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

########### cluster motif matrix (theta)  ###################

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

##                              topic
## phrase                               1        2        3
##   Macropygia_unchall        7.08e-07 6.05e-07 2.50e-03
##   Streptopelia_chinensis    2.95e-06 7.71e-03 6.89e-07
##   Streptopelia_senegalensis 7.09e-07 3.30e-03 6.87e-07
##   Columba_pulchricollis     7.08e-07 6.05e-07 1.25e-03
##   Streptopelia_orientalis   1.25e-02 2.50e-03 8.47e-07
##   Chalcophaps_indica        7.09e-07 3.30e-03 6.88e-07

colSums(theta)

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

We visulize the cluster membership proportions matrix `omega` using a Block Structure Plot representation. In this representation, we use one metadata for forming blocks (the East/West direction in the figure below) and in each block, the samples are arranged by a second metadata (Elevation in the figure below).

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



ecostructure also provides tools to compare the GoM model fit for a particular K (number of clusters) on the data, with respect to null model using the `nullmodel_GoM` function. The package lets the user choose different null model types - *frequency*, *richness*, *independent swap* and *trial swap*.

```
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: 2761, done.
## log BF( 2 ) = 334.73
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2889.8, done.
## log BF( 2 ) = 449.79
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2538.2, done.
## log BF( 2 ) = 298.7
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 3015.5, done.
## log BF( 2 ) = 402.95
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 2830.2, done.
## log BF( 2 ) = 350.39
##
## Estimating on a 38 document collection.
## Fit and Bayes Factor Estimation for K = 2
## log posterior increase: 3003.6, done.
## log BF( 2 ) = 787.51
## $GoMBF.obs
## [1] 15725
##
## $GoMBF.rand
## [1] 13621 13593 13229 12991 13387
##
## $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 the 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.

5 Processing data along different axes of diversity

In this section, we demonstrate how **ecostructure** can be used to process grid level counts data corresponding to different axes of diversity - regional, phylogenetic and trait-based. The idea is to obtain motifs or clusters defined by these different axes of diversity.

5.1 Phylogenetic analysis

For building the counts data based on phylogenetic diversity, we provide a function, `collapse_counts_by_phylo()` 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 under 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)

## [1]  38 196
```

We see that at the branching time `collapse_at` of 10, we get 196 phylogenetic clusters of species, and the abundance data are summed over all taxa in a particular cluster to generate the output `phylo_counts`.

We ue the data `phylo_counts` as input for the `CountClust::FitGoM` or `CountClust::FitGoMpool` funtions to determine how clustering patterns are influenced by phylogenetic diversity.

5.2 Regional analysis

For the regional profile, **ecostructure** allows the user to create Global assemblage dispersion fields and build maps data from those fields. To create these assemblage dispersion fields, the user is required to obtain 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 obtained 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 using the `CreateGlobalDispersionFields` function as demonstrated below.

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

We next show how the maps can be generated from the above dispersion field and the global shape file using the `CreateMapsFromDispersionFields` function.

```
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 = c(1,1))
maps <- CreateMapsFromDispersionFields(dispersion.field, global_shapefile)
```

The function returns a list of map plots with as many elements as the number of sites. We demonstrate an example visualization of the map for the first site.

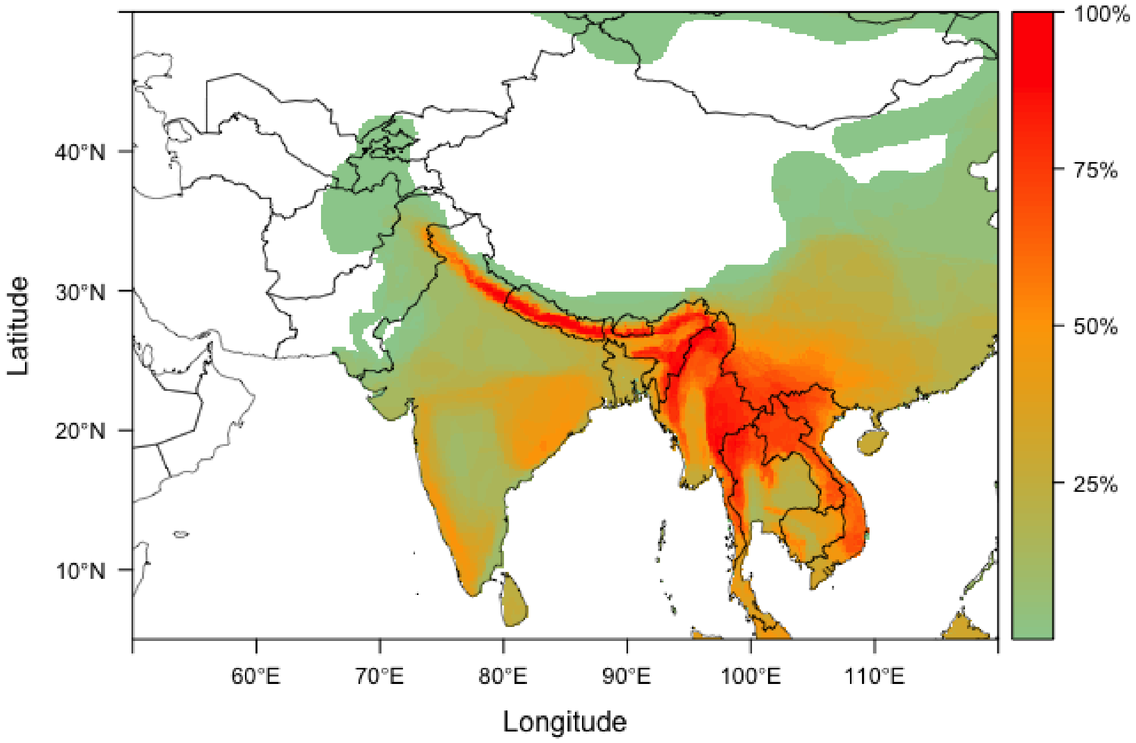
```
maps[[1]]
```

Finally, the user can generate the counts data coresssponding to regional diversity profile using the `DispersionFieldTocounts()` function

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

```
dim(regional_counts)

## [1]     38 201600

regional_counts[1:5,1:5]

##    [,1] [,2] [,3] [,4] [,5]
## U1    0    0    0    0    0
## U2    1    1    1    1    1
## MA    0    0    0    0    0
## A2    0    0    0    0    0
## A3    0    0    0    0    0
```

The columns in this data represent 1 degree by 1 degree cells on which the dispersion fields are assembled. The cells are serially stacked along latitudes to form columns in the above matrix. This data is then used as an input for `CountClust::FitGoM` or `CountClust::FitGoMpool` to determine how clustering patterns are influenced by regional diversity.



5.3 Trait based analysis

In order to build grid counts data based on functional diversity, we discuss two approaches.

In the first approach, 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` to perform this.

```
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])
```

The other approach is to use a trait or traits to collapse the bird species by performing hierarchical clustering of the bird species, cutting off a dendrogram at a particular level, thereby forming several clusters and then for each grid point, aggregating the counts data for all the species in the cluster. The cut level is chosen subjectively based on what proportion of variation in the actual abundance data is explained by the clusters at that level. We use the `trait_cluster()` function in **ecostructure** to perform this, an application of which is demonstrated below.

```
bill_traits <- as.matrix(dist(scale(species_metadata[,c(1:3)])))
bill_trait_clust <- trait_cluster(counts = taxonomic_counts,
                                  traits = bill_traits, prop_div=0.3)
```

We first generate a traits matrix, with the columns representing the traits and rows corresponding to the species. Then we use that to perform a hierarchical clustering of species and cut the dendrogram at a specified level, given by `prop_div`.

The clusters of species formed by cutting the dendrogram are then used to collapse the counts data and reduce the original counts matrix with species along the columns to one with the clusters along the columns. This matrix can then used as input for the `CountClust::FitGoM` or `CountClust::FitGoMpool` to determine how clustering patterns are influenced by species traits.

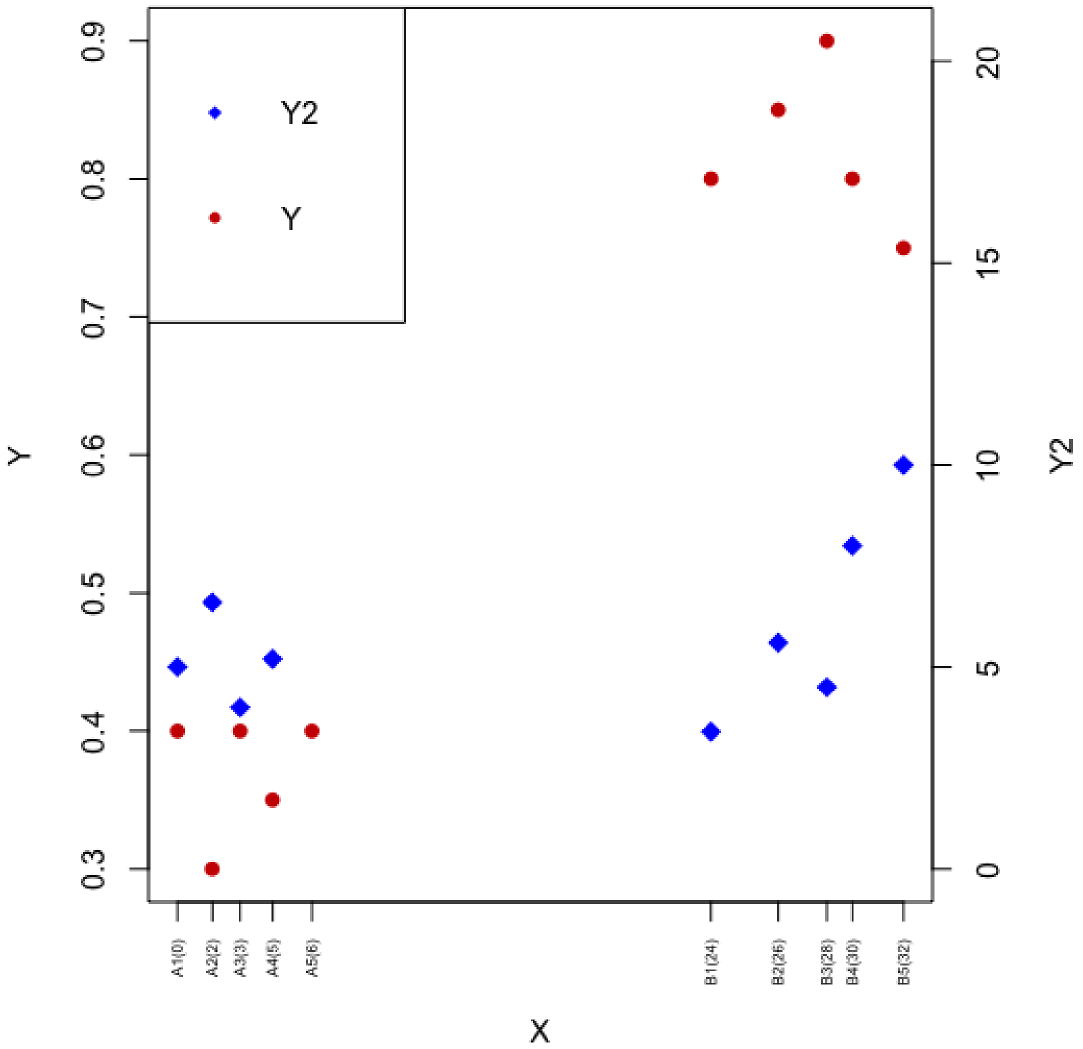
6 Extras

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.

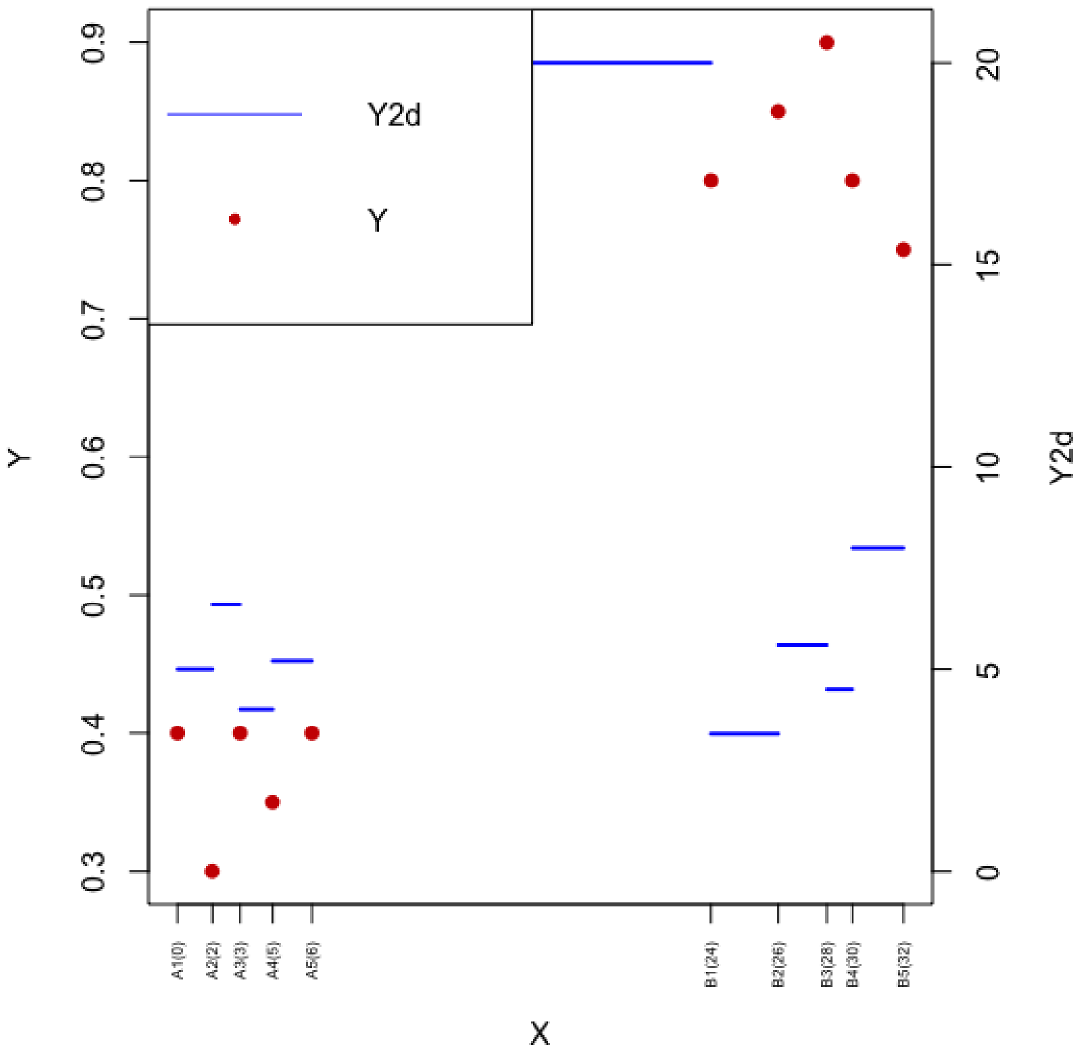
```
annotation = data.frame(x_names = c(paste0("A",1:5), paste0("B",1:5)),
x = c(0.5,2.0, 3.2, 4.6, 6.3,  23.5, 26.4, 28.5, 29.6, 31.8),
y1 = c(0.4, 0.3, 0.4, 0.35, 0.4, 0.8, 0.85, 0.9, 0.8, 0.75),
y2 =c(5, 6.6, 4, 5.2, 20, 3.4, 5.6, 4.5, 8, 10))

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)
```



7 Session Info

```
sessionInfo()

## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS Sierra 10.12.5
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ecostructure_0.99.1   Biobase_2.34.0        BiocGenerics_0.20.0
##  [4] maptools_0.9-2        SpatialExtremes_2.0-5 phytools_0.6-20
##  [7] maps_3.2.0            ape_4.1               gridExtra_2.3
## [10] rgdal_1.2-11          raster_2.5-8          sp_1.2-5
## [13] ordtpx_0.0.1          slam_0.1-40           maptpx_1.9-3
## [16] CountClust_0.1.2      ggplot2_2.2.1         knitr_1.16
##
## loaded via a namespace (and not attached):
##  [1] viridisLite_0.2.0       splines_3.3.3           gtools_3.5.0
##  [4] expm_0.999-2            highr_0.6               stats4_3.3.3
##  [7] latticeExtra_0.6-28     animation_2.5           numDeriv_2016.8-1
## [10] lattice_0.20-34         limma_3.30.13           quadprog_1.5-5
## [13] phangorn_2.2.0          digest_0.6.12           RColorBrewer_1.1-2
## [16] colorspace_1.3-2        picante_1.6-2           cowplot_0.8.0
## [19] Matrix_1.2-10           plyr_1.8.4              pkgconfig_2.0.1
## [22] mvtnorm_1.0-6           scales_0.4.1            tibble_1.3.4
## [25] combinat_0.0-8          mgcv_1.8-17             hexbin_1.27.1
## [28] nnet_7.3-12             lazyeval_0.2.0          rasterVis_0.41
## [31] mnormt_1.5-5            survival_2.40-1         magrittr_1.5
## [34] evaluate_0.10           msm_1.6.4               nlme_3.1-131
## [37] MASS_7.3-45             foreign_0.8-67          vegan_2.4-3
## [40] tools_3.3.3             BiocStyle_2.2.1         stringr_1.2.0
## [43] munsell_0.4.3           cluster_2.0.5           plotrix_3.6-6
## [46] clusterGeneration_1.3.4 rlang_0.1.1.9000        igraph_1.1.1
## [49] boot_1.3-18             gtable_0.2.0            flexmix_2.3-14
## [52] reshape2_1.4.2.9000     zoo_1.8-0               fastmatch_1.1-0
## [55] permute_0.9-4           modeltools_0.2-21       stringi_1.1.5
## [58] SQUAREM_2016.8-2        Rcpp_0.12.12            scatterplot3d_0.3-40
## [61] coda_0.19-1
```

References

Dey K, Hsiao J and Stephens M. CountClust: Clustering and Visualizing RNA-Seq Expression Data using Grade of Membership Models. 2016. *R package version 0.1.2.* https://github.com/kkdey/CountClust

Pritchard, Jonathan K., Matthew Stephens, and Peter Donnelly. 2000. Inference of population structure using multilocus genotype data. *Genetics*. 155.2, 945-959.

Rosenberg NA, Pritchard JK, Weber JL, Cann HM, Kidd KK, Zhivotovsky LA, Feldman MW. 2002. The genetic structure of human populations. *Science*. 298, 2381-2385.