Classifying vegetation stands in terms of structure and composition

Miquel De Cáceres[1]

[1]Centre Tecnològic Forestal de Catalunya, Solsona, Catalonia, Spain

April 11, 2018

Contents

1 Introduction

The aim of this tutorial is to show how to implement the method to calculate dissimilarities between communities presented in [De Cáceres et al., 2013]. This method allows incorporating both the structure and the composition of the community in the dissimilary measurement. The functions needed to carry out computations have been included in package `vegclust` so we start by loading the package:

```
> library(vegclust)
```

2 Post-fire vegetation regeneration data

In order to illustrate the method we will use a stratified vegetation data set containing data from 96 stands. The data was obtained to investigate patterns vegetation regeneration three years after the impact of a wildfire. Data

1

were collected in 2012 by Miquel De Caceres and Albert Petit in Horta de Sant Joan (Catalonia, Spain). The R object is of class `stratifiedvegdata` (actually a list).

```
> data(medreg)
> class(medreg)

[1] "stratifiedvegdata" "list"

> length(medreg)

[1] 96
```

The dataset contains 96 stands (list elements), each of them a data.frame where rows correspond to broad plant functional groups (Pine trees, Oak trees, Tall shrubs and small trees, Scrubs and small shrubs and Grass) and columns correspond to vegetation strata (1 to 7). The upper heights of the vegetation strata are the following (in cm.) vector:

```
> strataUp = c(20,50,100,300,600,1200,2400)
```

And the width (range of heights) of each stratum is:

```
> strataWidths = c(20,30,50,200,300,600,1200)
```

Species abundance values are percentage cover values estimated using cover classes:

```
> medreg[[1]]

                               1     2  3 4 5 6 7
Pine trees                   0.0   0.0  0 0 0 0 0
Quercus trees               12.5  25.0  0 0 0 0 0
Tall shrubs and small trees  0.0  62.5 25 0 0 0 0
Scrubs and small shrubs     12.5  37.5  0 0 0 0 0
Grass                       50.0   0.0  0 0 0 0 0
```

The data is read as follows. Shrubs reaching stratum 3 (50 - 100 cm) had a cumulative cover of 25%, while shrubs reaching only stratum 2 (20 - 50 cm) had a cumulative cover of 62.5%. Thus, the observers grouped plants according to their height and functional group, and estimated the cover for those groups of plants.

3 Cumulative abundance profiles

The *cumulative abundance profile* (CAP) is a function that takes a value of *size* as input (here the *size* is a vegetation stratum) and returns the cumulative abundance of organisms (here the cumulative cover value) whose size is *equal to* or *larger than* the input value. In our case, the CAP function is the cumulative cover of plants reaching the current stratum or higher strata. Calculations are made using function `CAP()` of vegclust:

```
> medreg.CAP <- CAP(medreg)
```

Note that a different CAP is calculated for each functional group and plot. The structure of the resulting R object is very similar to the stratified data:

```
> class(medreg.CAP)

[1] "CAP"  "list"

> length(medreg.CAP)

[1] 96
```

If we inspect the first element of the list, we can see the difference between the original data and the cumulative abundance profile.

```
> medreg.CAP[[1]]

                             1    2  3 4 5 6 7
Pine trees                 0.0  0.0  0 0 0 0 0
Quercus trees             37.5 25.0  0 0 0 0 0
Tall shrubs and small trees 87.5 87.5 25 0 0 0 0
Scrubs and small shrubs   50.0 37.5  0 0 0 0 0
Grass                     50.0  0.0  0 0 0 0 0
```

Moreover, it is possible to graphically display the CAP of a given stand (it may become difficult to interpret when the number of species is large). For example, we can display the CAP for each functional group of the first stand (`plots="1"`):

```
> plot(medreg.CAP, plots="1", sizes=strataUp, xlab="Height (cm)",
+      ylab="Cumulative percent cover")
> legend("topright", col=1:5, lty=1,
+        legend=c("Pines","Oaks","Tall shrubs","Scrubs","Grass"),
+        bty="n")
```
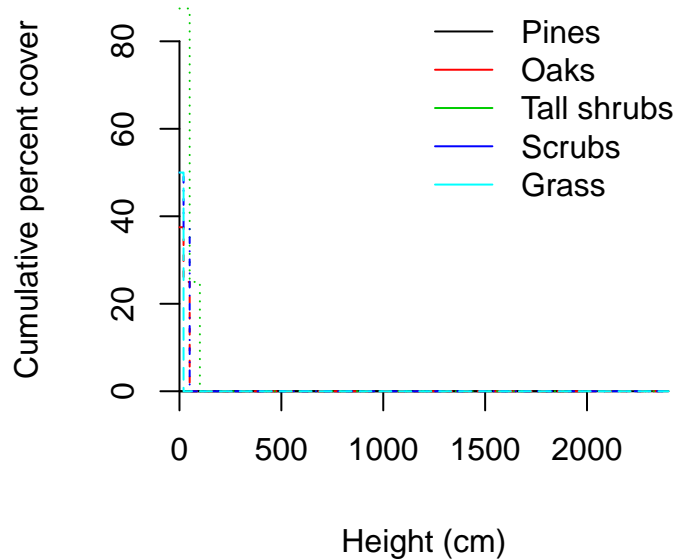
In this case the vegetation is a short but dense shrubland. Note that in the plot we used `strataUp` to set the x-axis, so that real heights are adequately represented.

4 Cumulative abundance surfaces

The concept of cumulative abundance profile can be extended to two structural variables, which leads to the concept of *cumulative abundance surface* (CAS). The CAS is a function that takes a the values of two structural variables (*size1* and *size2*) as input and returns the cumulative abundance of organisms whose size is *equal to* or *larger than* the input values in one of the structural variables or in both. In the case of forests, natural choices for structural variables are tree diameter and tree height. Since our post-fire regeneration dataset only includes one structural variable, we will illustrate the concept of CASs using a synthetic data set consisting in a single plot where the species identity, diameter and height of a hundred trees has been measured. We start by building a tree-based data set:

```
> pl = rep(1,100) # All trees in the same plot
> sp = ifelse(runif(100)>0.5,1,2) # Random species identity (species 1 or 2)
> h=rgamma(100,10,2) # Heights (m)
```

```
> d = rpois(100, lambda=h^2) # Diameters (cm)
> m = data.frame(plot=pl,species=sp, height=h,diameter=d)
```

In this example, we will use basal area (m2) as measure of abundance. We calculate the area (in square meters) of each tree:

```
> m$ba = pi*(m$diameter/200)^2
```

This specific data looks as follows:

```
> print(head(m))

  plot species    height diameter          ba
1    1          1 3.988839       15 0.017671459
2    1          2 4.040549       13 0.013273229
3    1          1 4.925023       26 0.053092916
4    1          2 3.549833       17 0.022698007
5    1          2 3.314271       15 0.017671459
6    1          2 2.427585        4 0.001256637
```

We start our analysis by defining two sets of size classes, one for height and the other for diameter:

```
> heights = seq(0,4, by=.25)^2 # Quadratic classes
> diams = seq(0,130, by=5) # Linear classes
```

We are ready to stratify the data set:

```
> tree.S<-stratifyvegdata(m, sizes1=heights, sizes2=diams,
+                 plotColumn = "plot", speciesColumn = "species",
+                 size1Column = "height", size2Column = "diameter",
+                 abundanceColumn = "ba")
```

Function `stratifyvegdata` is used to reshape data sets so that they are suitable for calculating CAPs or CASs. In the case of one structural variable, the function returns a list of matrices, one for each plot. The post-fire vegetation data presented in the previous section is an example of this. In the case of two structural variables, the `stratifyvegdata` returns a list of three-dimensional arrays, one for each plot. The cumulative abundance surface is then calculated using function `CAS`:

```
> tree.CAS <- CAS(tree.S)
```
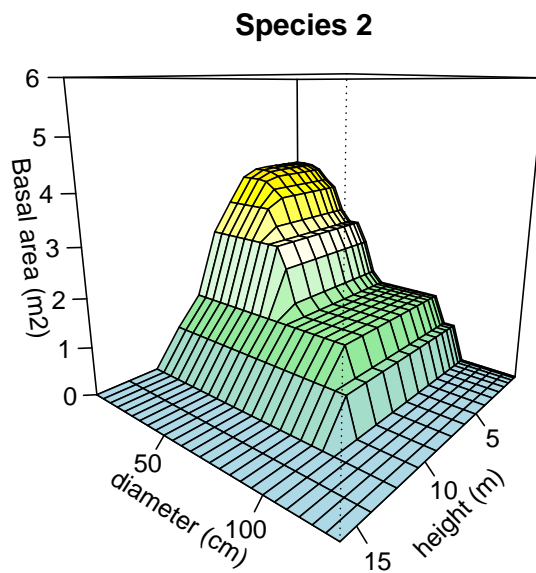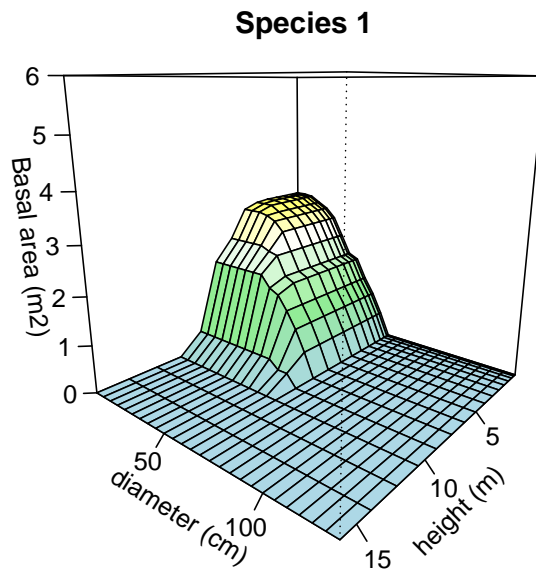
We can plot the surface corresponding to each species as follows:

```
> par(mfrow=c(2,1), mar=c(4,5,2,1))
> plot(tree.CAS, species=1, sizes1=heights[-1], xlab="height (m)",
+     ylab="diameter (cm)", sizes2=diams[-1], zlab="Basal area (m2)",
```

```
+          zlim = c(0,6), main="Species 1")
> plot(tree.CAS, species=2, sizes1=heights[-1], xlab="height (m)",
+          ylab="diameter (cm)", sizes2=diams[-1], zlab="Basal area (m2)",
+          zlim = c(0,6), main = "Species 2")
```

Species 1

Species 2

One nice property of CAS is that its marginal distributions are CAPs. This can be easily shown if we compare the marginal CAP for height:

```
> print(CASmargin(tree.CAS, margin=1))

$`1`
  (0,0.0625] (0.0625,0.25] (0.25,0.562] (0.562,1] (1,1.56]
1   3.134917      3.134917      3.134917 3.134917 3.134917
2   3.974743      3.974743      3.974743 3.974743 3.974743
  (1.56,2.25] (2.25,3.06] (3.06,4] (4,5.06] (5.06,6.25]
1    3.134917    3.134838 3.109234 3.004776    2.391145
2    3.974743    3.974429 3.954087 3.810673    3.291604
  (6.25,7.56]  (7.56,9]  (9,10.6] (10.6,12.2] (12.2,14.1]
1    1.937656 0.4124911 0.0000000           0           0
2    2.687004 1.3351769 0.7542964           0           0
  (14.1,16]
1         0
2         0


attr(,"class")
[1] "CAP"  "list"
```

with the CAP directly build using heights:

```
> tree.S2<-stratifyvegdata(m, sizes1=heights, plotColumn = "plot",
+                          speciesColumn = "species", size1Column = "height",
+                          abundanceColumn = "ba")
> print(CAP(tree.S2))

$`1`
  (0,0.0625] (0.0625,0.25] (0.25,0.562] (0.562,1] (1,1.56]
1   3.134917      3.134917      3.134917 3.134917 3.134917
2   3.974743      3.974743      3.974743 3.974743 3.974743
  (1.56,2.25] (2.25,3.06] (3.06,4] (4,5.06] (5.06,6.25]
1    3.134917    3.134838 3.109234 3.004776    2.391145
2    3.974743    3.974429 3.954087 3.810673    3.291604
  (6.25,7.56]  (7.56,9]  (9,10.6] (10.6,12.2] (12.2,14.1]
1    1.937656 0.4124911 0.0000000           0           0
2    2.687004 1.3351769 0.7542964           0           0
  (14.1,16]
1         0
2         0


attr(,"class")
[1] "CAP"  "list"
```

Finally, compare the previous three-dimensional figures, with the marginal
CAP plots for diameters and heights:

```
> par(mfrow=c(2,1), mar=c(4,5,2,1))
> plot(CASmargin(tree.CAS,margin=1), plots=1, sizes=heights[-1],
+      xlab="height (m)", ylab="Basal area (m2)", ylim = c(0,7))
> plot(CASmargin(tree.CAS,margin=2), plots=1, sizes=diams[-1],
+      xlab="diameter (cm)", ylab="Basal area (m2)", ylim = c(0,7))
```
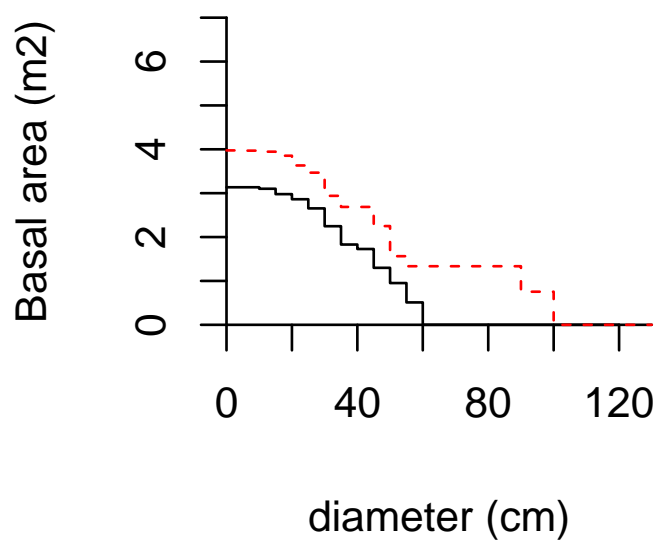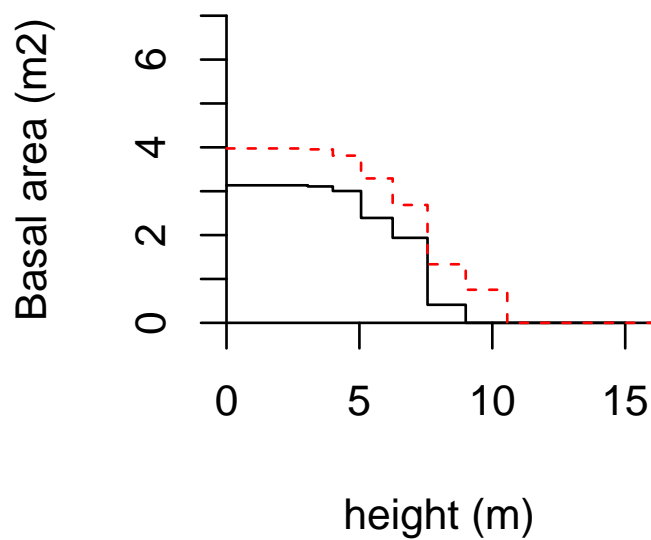
5 Dissimilarities in structure and composition

Although CAPs can be used to graphically display the structure and composition of vegetation stands, the whole point of defining the CAP function was to allow comparisons between stands. Returning to the post-fire vegetation regeneration data, ee can calculate dissimilarities for all pairs of stands, thus obtaining a square and symmetric matrix with dissimilarity values:

```
> medreg.D = vegdiststruct(medreg.CAP, method="bray",
+                          classWeights=strataWidths)
```

In the above sentence we forced strata to have different weight, according to the range of heights that each stratum occupies. There are different alternatives with respect to the dissimilarity index. In our case we chose the CAP generalization of Bray-Curtis [De Cáceres et al., 2013]. If we want to know, for example, the dissimilarity between stands '1' and '2' we simply write:

```
> as.matrix(medreg.D)[1,2]
```

```
[1] 0.2713178
```

When calculating dissimilarities it is possible to transform the CAP values in order to prevent large abundance values to have an undue influence in the analysis. In our case we choose to take the square root of cumulative cover values:

```
> medreg.Dsqrt = vegdiststruct(medreg.CAP, method="bray",
+                          classWeights=strataWidths, transform="sqrt")
```

We can use metric multidimensional scaling to represent the distances between stands obtained in both cases:

```
> par(mfrow=c(2,1), mar=c(4,5,2,1))
> X<-cmdscale(medreg.D, k=2)
> plot(X, xlab="MDS 1", ylab="MDS 2", asp=1,
+      main="Cover untransformed", cex=0.5)
> Xsqrt<-cmdscale(medreg.Dsqrt, k=2)
> plot(Xsqrt, xlab="MDS 1", ylab="MDS 2", asp=1,
+      main="Cover sqrt-transformed", cex=0.5)
```

Cover untransformed

Cover sqrt−transformed
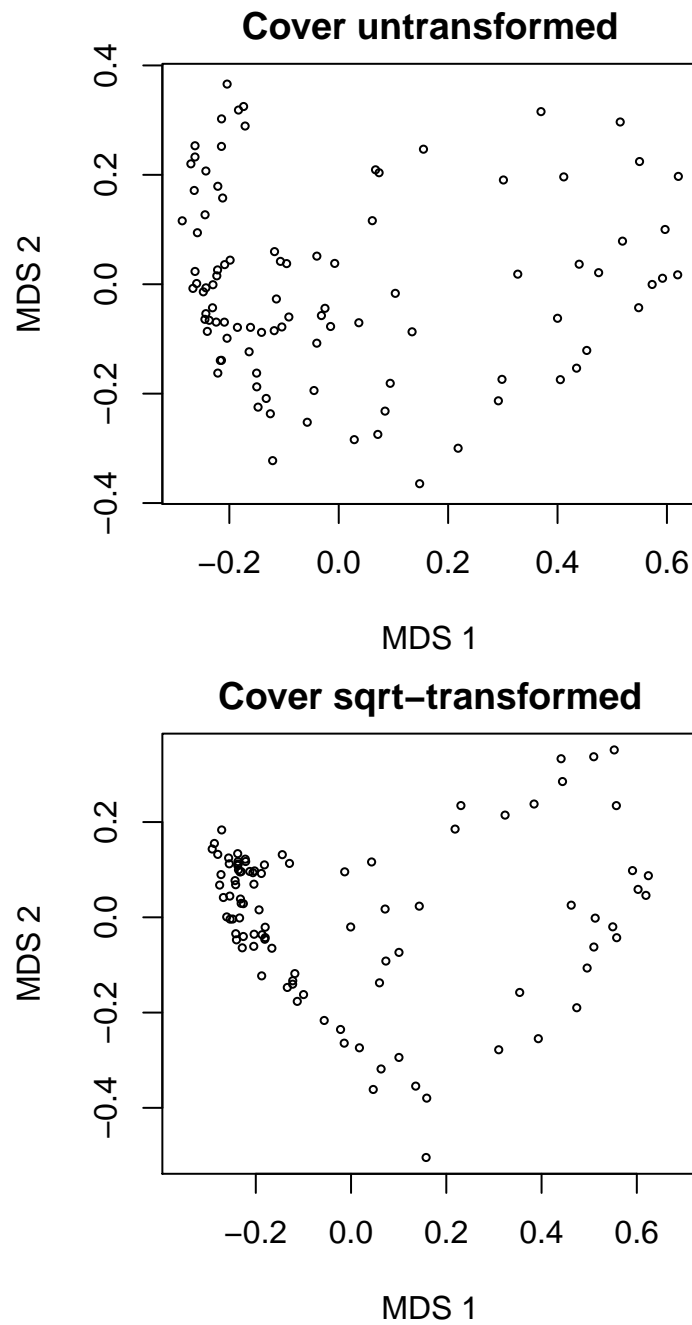
Note that the differences between the two ordination plots are remarkable.

6 Classification of vegetation stands

In this section we use the square-root transformed dissimilarities between vegetation stands to obtain a classification of the stands in terms of their

structure and composition. If you are not familiarized with non-hierarchical clustering, you can read the tutorial about `vegclust` package. We start by setting the number of clusters to be found (`nclusters`) and the size of clusters (`dnoise`, a parameter used to leave stands that are far from all group prototypes unclassified):

```
> nclusters = 6
> dnoise = 0.40
```

We call function `vegclust` using the clustering method *HNCdd*, which indicates (a) hard clustering, (b) medoids as prototypes, and (c) noise clustering (i.e. excluding outliers in a special class called *noise class*):

```
> vc<-vegclustdist(medreg.Dsqrt, mobileMemb = nclusters,
+                  method="HNCdd", dnoise=dnoise, nstart=100)
```

With `nstart=100` we indicate that the algorithm should be run 100 times starting from random seeds. This is advisable in order to maximize the chance of having suboptimal solutions. The prototypes identified by the algorithm are the following medoids (these are indices of stands in `medreg`):

```
> medoids<-vc$mobileCenters
> print(medoids)

[1] 20 24 71 92 60  3
```

The number of stands belonging to each cluster can be found using:

```
> cluster<-defuzzify(vc)$cluster
> table(cluster)

cluster
M1 M2 M3 M4 M5 M6  N
21 28 13  9 10  6  9
```

Note that, because of the model chosen (and with the parameter `dnoise`), there are a number of stands that are left unclassified (i.e. those assigned to class 'N'). A useful way to display the results of the cluster analysis is by showing the stand memberships to clusters in the ordination:

```
> clNum = as.numeric(as.factor(cluster))
> plot(Xsqrt, xlab="MDS 1", ylab="MDS 2",
+      pch=clNum, col=clNum)
> legend("topleft", col=1:(nclusters+1), pch=1:(nclusters+1),
+        legend=levels(as.factor(cluster)), bty="n")
```
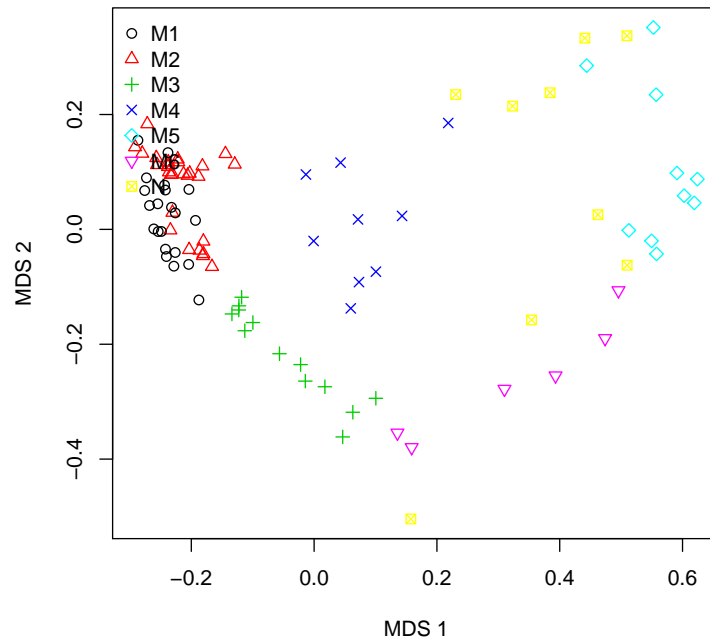
While the stands belonging to *true* clusters are more or less close, those that are assigned to the *noise* can be far appart, because the only fact that makes them be in the same class is their lack of membership for *true* clusters.

To facilitate the interpretation of the clusters we can extract the cumulative abundance profiles of the cluster medoids:

```
> CAPm = CAPcenters(medreg.CAP, vc)
> n = names(CAPm)
```

For example, we can inspect the structure and composition of the fourth group:

```
> round(CAPm[[n[4]]], dig=1)
```

	1	2	3	4	5	6	7
Pine trees	0.0	0.0	0.0	0.0	0	0	0
Quercus trees	25.0	25.0	0.0	0.0	0	0	0
Tall shrubs and small trees	117.5	112.5	87.5	12.5	0	0	0
Scrubs and small shrubs	25.0	25.0	0.0	0.0	0	0	0
Grass	37.5	0.0	0.0	0.0	0	0	0

The following displays graphically the CAPs of all six groups of vegetation stands (code not shown):
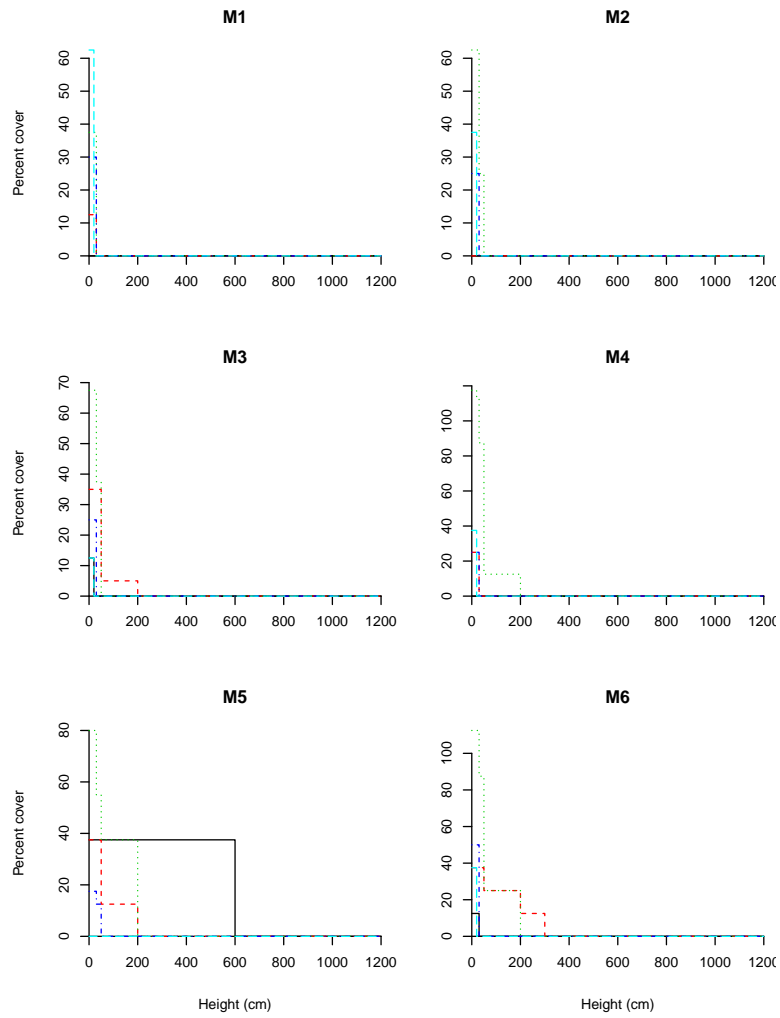
References

Miquel De Cáceres, Pierre Legendre, and Fangliang He. Dissimilarity measurements and the structure of plant communities. *Methods in Ecology and Evolution*, 2013.