Coordenadas Principales

Jose Luis Vicente Villardon

11/19/2019

Análisis de Coordenadas Principales

Dada una matriz $$ de disimilaridades/distancias entre un conjunto de n objetos o puntos, es posible convertirla en una matriz de productos escalares tomando

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}\Delta^2\mathbf{H}^T$$

donde $\mathbf{H}_{(n\times n)}$ es la matriz de centrado :

$$\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$$

Se dice que $$ es una matriz de distancias euclídeas si es posible obtener una configuración de puntos $\mathbf{Y}_{(n\times d)}$, de los n objetos o puntos en dimensión d, de forma que las interdiistancias euclídeas entre las filas de $\mathbf{Y}$ reproduzcan exactamente las distancias observadas en $$.

Si $\mathbf{B}$ es una matriz de productos escalares entre cualquier conjunto de n vectores (puntos) con respecto a su centro de gravedad, en cualquier espacio Euclídeo, entonces las proyecciones de los puntos en el subespacio de baja dimensión más próximo se obtienen de la estructura espectral de $\mathbf{B}$, como :

$$\mathbf{Y} = \mathbf{U}\mathbf{D}_\lambda^{1/2}$$

Donde $\mathbf{B} = \mathbf{U}\mathbf{D}_\lambda\mathbf{U}^T$ ($\mathbf{U}^T\mathbf{U} = \mathbf{I}$) es la descomposición de valores y vectores propios de $\mathbf{B}$.

La configuración obtenida por los puntos de $\mathbf{Y}$ reproduce aproximadamente, los productos escalares de la matriz original $\mathbf{B}$ y, por tanto, las distancias a partir de las que fueron calculados. Para la representación en dimensión reducida basta tomar las primeras columnas de $\mathbf{Y}$. Si la matriz $\mathbf{B}$ es semidefinida positiva, entonces puede obtenerse una representación exacta en $(n-1)$ dimensiones. La variabilidad explicada en dimensión reducida r vendrá dada, como es habitual, por

$$\frac{\sum_{i=1}^{r}\lambda_i^2}{\sum_{i=1}^{n-1}\lambda_i^2}\times 100$$

Si la distancia observada no es eucídea, es posible que tenh¡gamos algún valor propio negativo.

Ejemplo1: RAPD

Vamos a realizar un ejemplo que se incluye en el paquete MultBiplotR. En primer lugar instalaremos el pequete y los paquetes asociados de los que depende.

```
local({
  r <- getOption("repos")
  r["CRAN"] <- "http://cran.cnr.berkeley.edu/"
  options(repos = r)
})
# install.packages(c("scales", "geometry", "deldir", "rgl", "mirt", "GPARotation",
"MASS", "kde2d", "lattice", "splom", "dae"))
# install.packages("http://biplot.usal.es/multbiplot/multbiplot-in-r/multbiplotr_19
1119tar.gz", repos = NULL, type="source")
library(MultBiplotR)
```

```
## Loading required package: scales
```

```
## Loading required package: geometry
```

```
## Loading required package: deldir
```

```
## deldir 0.1-23
```

```
## Loading required package: mirt
```

```
## Loading required package: stats4
```

```
## Loading required package: lattice
```

```
## Loading required package: GPArotation
```

```
## Loading required package: rgl
```

```
## Loading required package: optimr
```

El ejemplo presentado en clase con datos de caña de azucar se incluye dentro del paquete con el nombre (RAPD). Cargamos los datos y los visualizamos.

```
data("RAPD")
```

Se trata de una matriz de datos binarios que solamente contiene valores 0 y 1 dependiendo de si cada marcador está presente o ausente.

En primer lugar vamos a añadir una variable nominal que contiene en origen de cada una de las vriedades de caña de azucar analizada.

```
Origin=c("B", "B",  "B", "B", "B",  "B", "B", "B",  "B", "B", "B",  "B", "B", "B",
"SP", "C", "C", "Co",  "Co",  "CP", "CP", "CP", "CP", "CL", "MEX", "POJ", "POJ", "R
agnar",
"PR", "PR", "PR", "PR", "PR",  "V",  "V", "V", "V", "V",  "V",  "V", "V", "V",
"V", "V", "V", "V", "V", "V", "V", "V" )

Origin=as.factor(Origin)
```

La hipótesis inicial es que las variedades que ¡tienen el mismo origen son más similares entre ellas que las varedades de distintos orígenes. Utilizaremos el Análisis de coordenadas Principales para tratar de corrborar esta hipótesis.

En primer lugar calculamos una medida de la distancia a partir de un coefieciente de disimilaridad. Utilizamos el coeficiente 4, que se corresponde con el coeficiente de concordancia simple.

```
Dis=BinaryProximities(RAPD, coefficient = 4)
names(Dis)
```

```
## [1] "TypeData"       "Type"           "Coefficient"    "Transformation"
## [5] "Data"           "Proximities"
```

El resultado es una lista que contiene los datos originales, el tipo de medida, el coeficiente y las proximidades ppropiamente dichas.

A continuaciób usamos este objeto para calcular las Coordenadas Principales a partir de las medidas de proximidad. Seleccionamos 4 dimesiones, aunque probablemente es suficiente con menos.

```
pco=PrincipalCoordinates(Dis, dimension = 4)
```

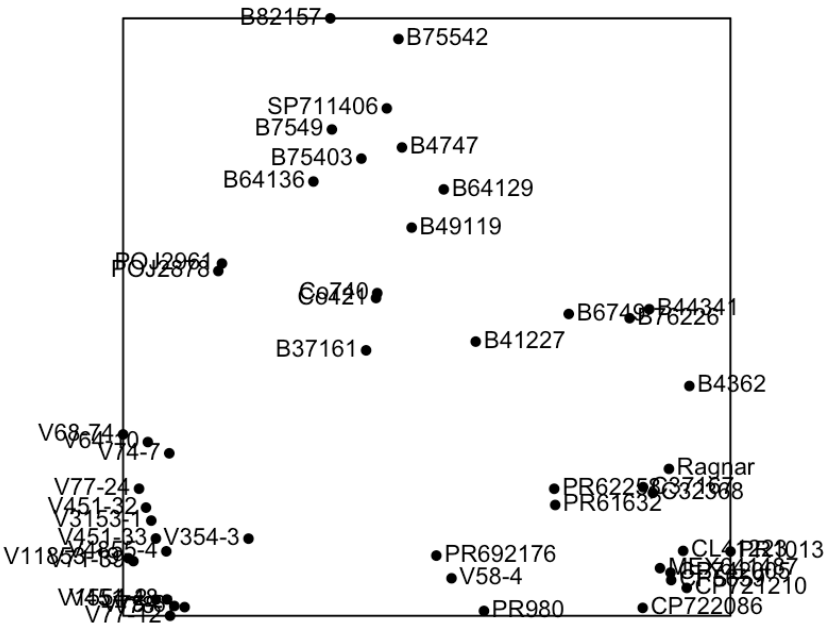
```
## [1] 50
```

```
summary(pco)
```

```
## [1] "PRINCIPAL COORDINATES ANALYSIS"
## [1] "Type of Data :  Binary"
## [1] "Type of Proximity :  dissimilarity"
## [1] "Coefficient :  Simple_Matching"
## [1] "Transformation :  sqrt(1-S)"
## [1] "-----------"
## [1] "-----------"
## [1] "Eigenvalues and explained Variance"
##       Eigenvalues Variance Explained Cummulative
## Dim 1 0.017767322          17.153079    17.15308
## Dim 2 0.012572618          12.137964    29.29104
## Dim 3 0.008370845           8.081452    37.37249
## Dim 4 0.005661987           5.466243    42.83874
## [1] "--------------------------------------"
##  RawStress    stress1    stress2   sstress1   sstress2        rsq
## 43.1251213  0.6235006  1.7092014  1.1755478  2.2150525  0.6754531
##   Spearman    Kendall
##  0.8155587  0.6358607
```

Las dos primeras dimensiones recogen el 29.29% de la variabilidad. Teniendo en cuenta que la dimensionalidad del proble es 49 (tenemos 50 filas), el pocentaje recogido por las primeras dimensiones es aceptable. Para tratar de corroborar la hipótesis de partida, relizamos el gráfico correspondiente. Como los nombres de las variedades son largos hemos reducido un poco el tamaño de las etiquetas. (RowCex =0.7)

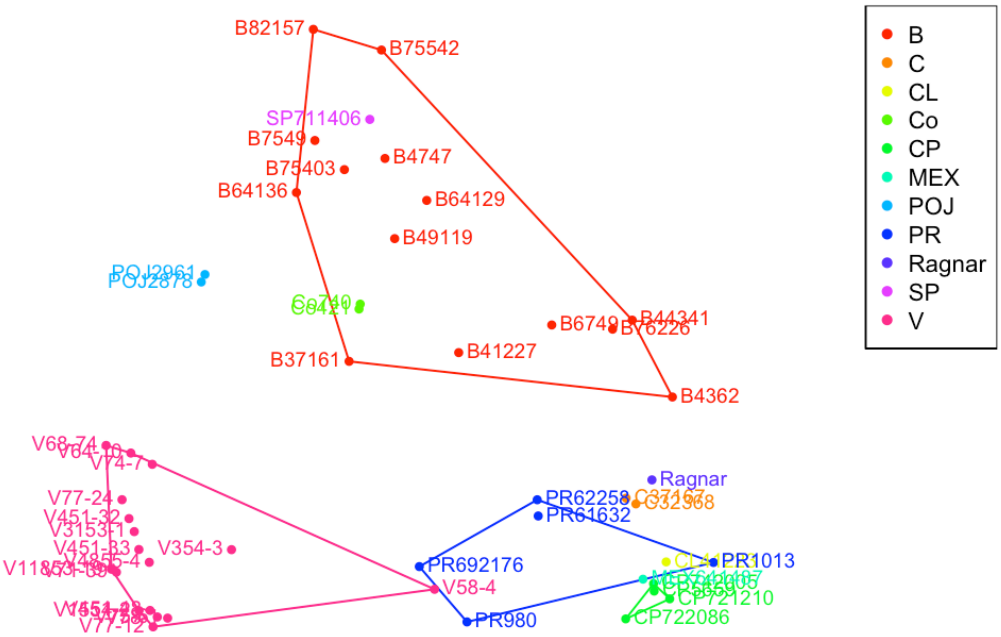
```
plot(pco, RowCex=0.7, ShowBox=TRUE)
```



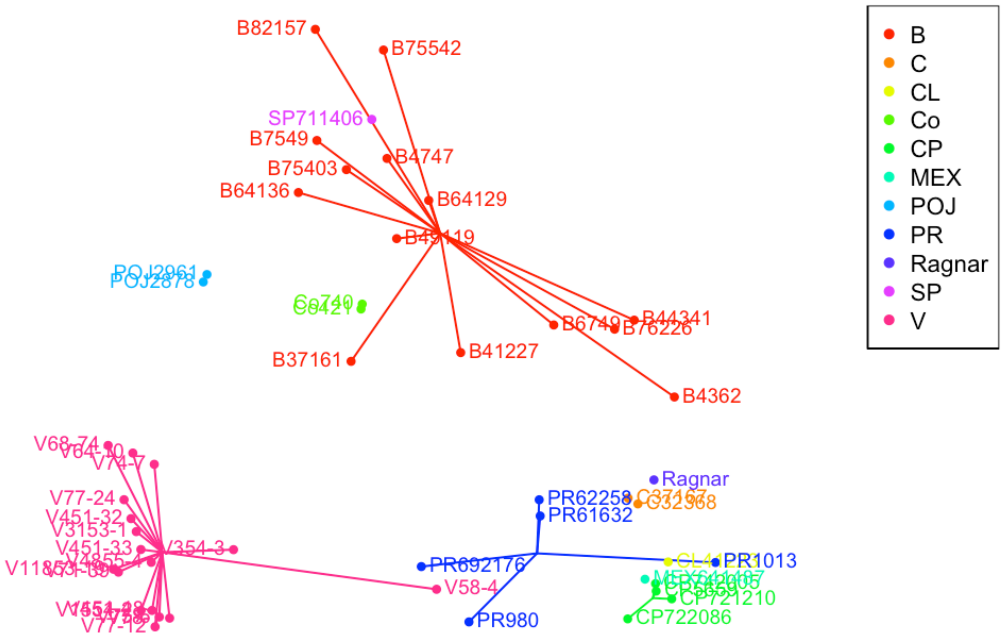
El gráfico resultante no contiene escalas en los ejes yaque, en realidad, son irrelevantes y no contienen información sobre los datos. Con las etiquetas seleccionadas es difícil seguir la pista a los puntos. Vamos a utilizar el color para añadir interpretabilidad al gráfico colocando los diferentes orígenes en distintos colores. La función *AddCluster2Biplot*, permite añadir clusters a cualquier representación eucídea de puntos, en particular, permite añadir clusters definidos por una variable nominal externa definida por el usiuario como es el caso del origen de las variedades.

Los clusters pueden representarse mediante la envoltura convexa del conjunto de puntos que lo forman, mediante una estrella que une el centro del cluster con todos sus puntos o mediante una elipse de confianza no paramétrica. Los clusters con dos o menos puntos no pueden representarse de ninguna de estas formas y se identifican solamente con el color.

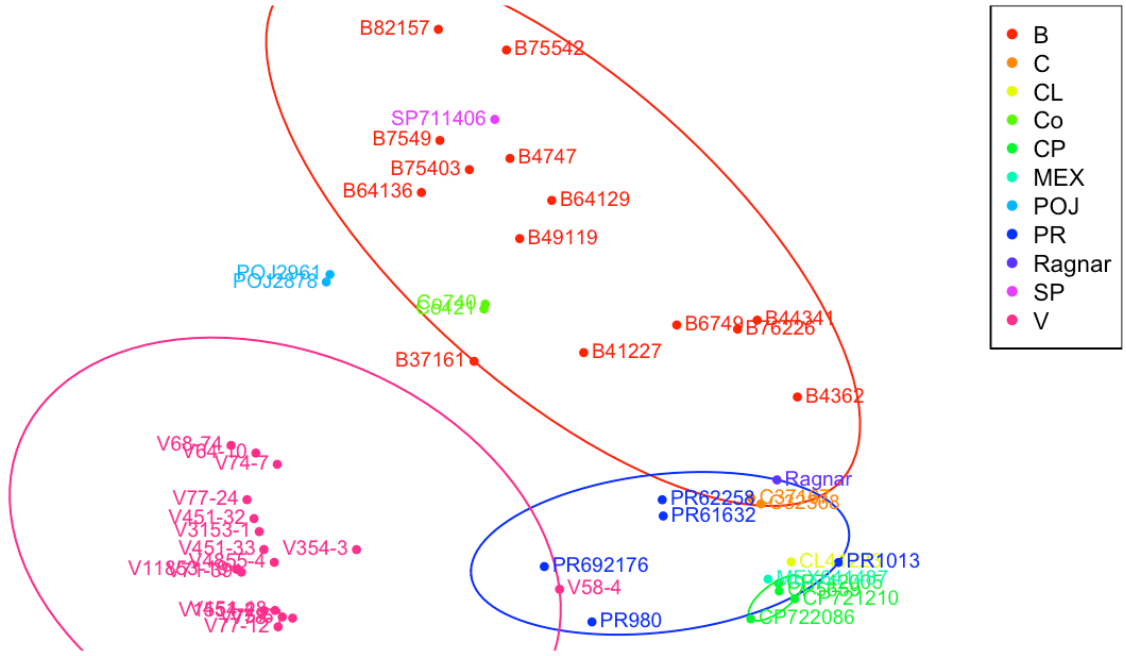
```
pco=AddCluster2Biplot(pco, ClusterType="us", Groups=Origin)
plot(pco, PlotClus=T, RowCex=0.6)
```




```
plot(pco, PlotClus=T, TypeClus="st", RowCex=0.6)
```

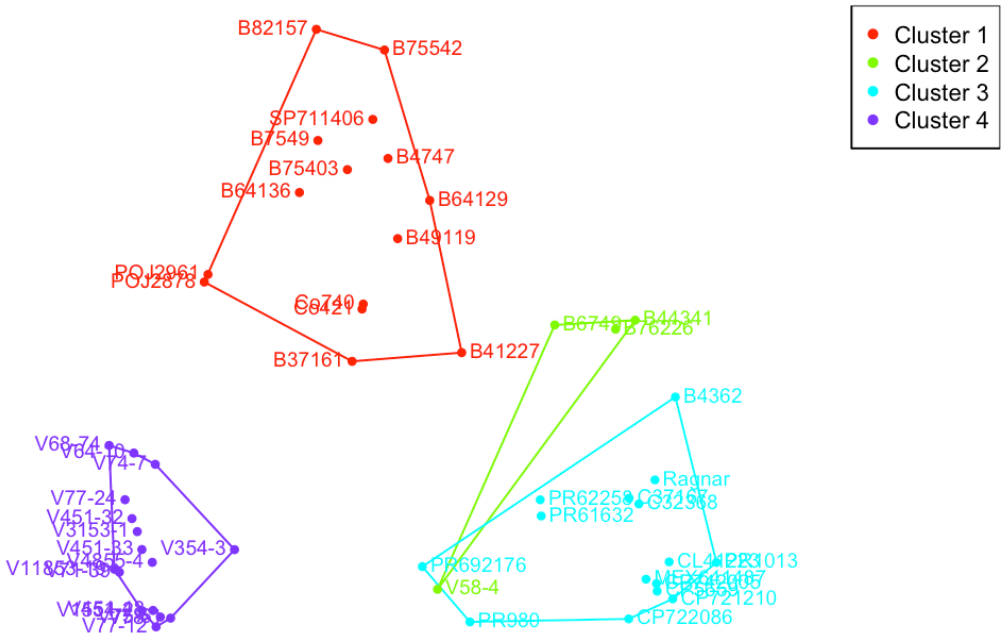



```
plot(pco, PlotClus=T, TypeClus="el", RowCex=0.6)
```

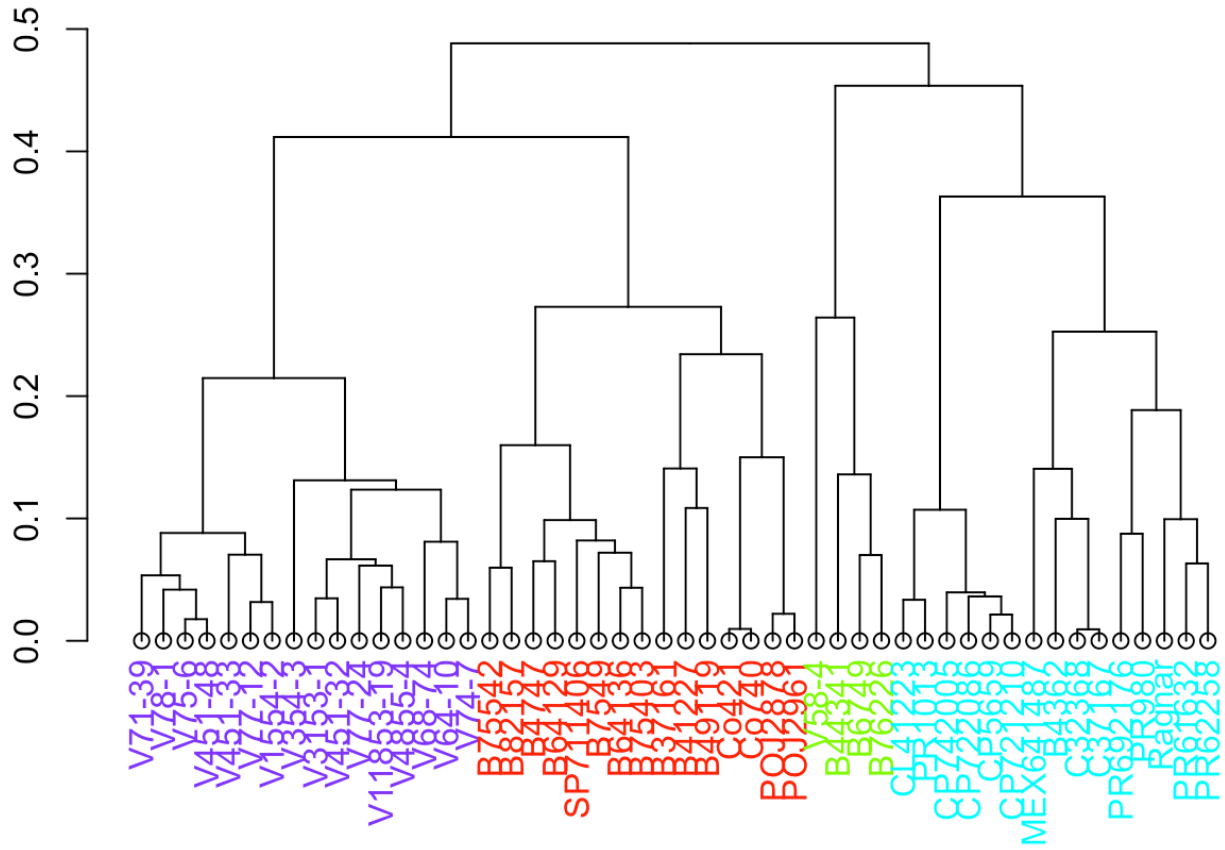


En lugar de grupos definidos por el usuario es posible usar grupos obtenidos a partir de un análisis de cluster jerárquico (la función usa *hclust* presente en la base de *R* y sus argumentos por defecto). Dentro del objeto se guarda también el dendograma resultante que puede dibujarse

```
pco=AddCluster2Biplot(pco, ClusterType="hi", NGroups=4)
plot(pco, PlotClus=T, RowCex=0.6)
```

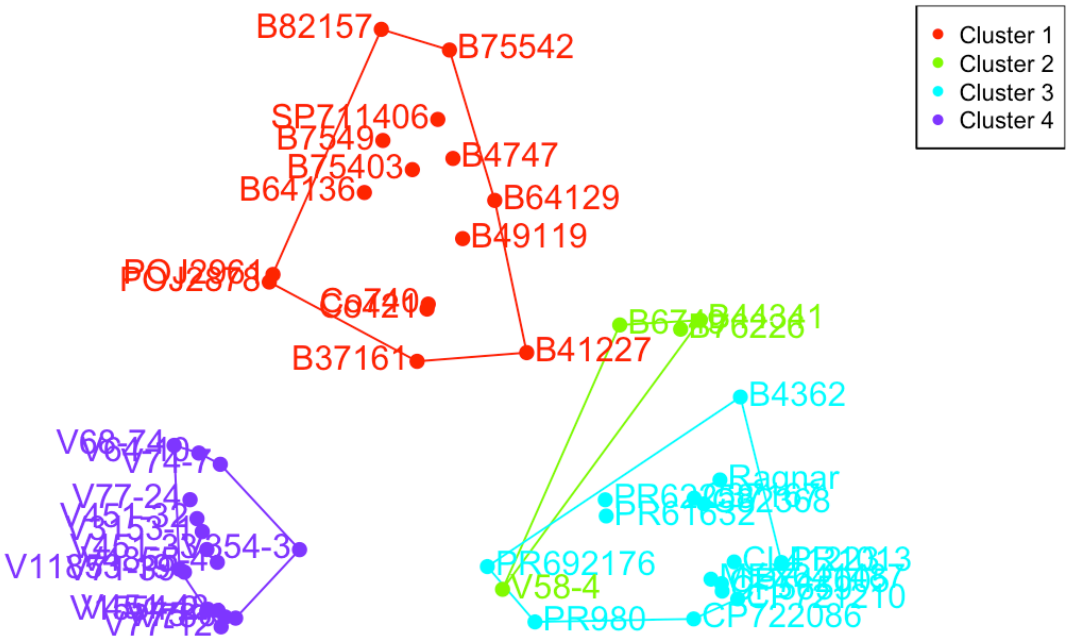



```
plot(pco$Dendrogram)
```



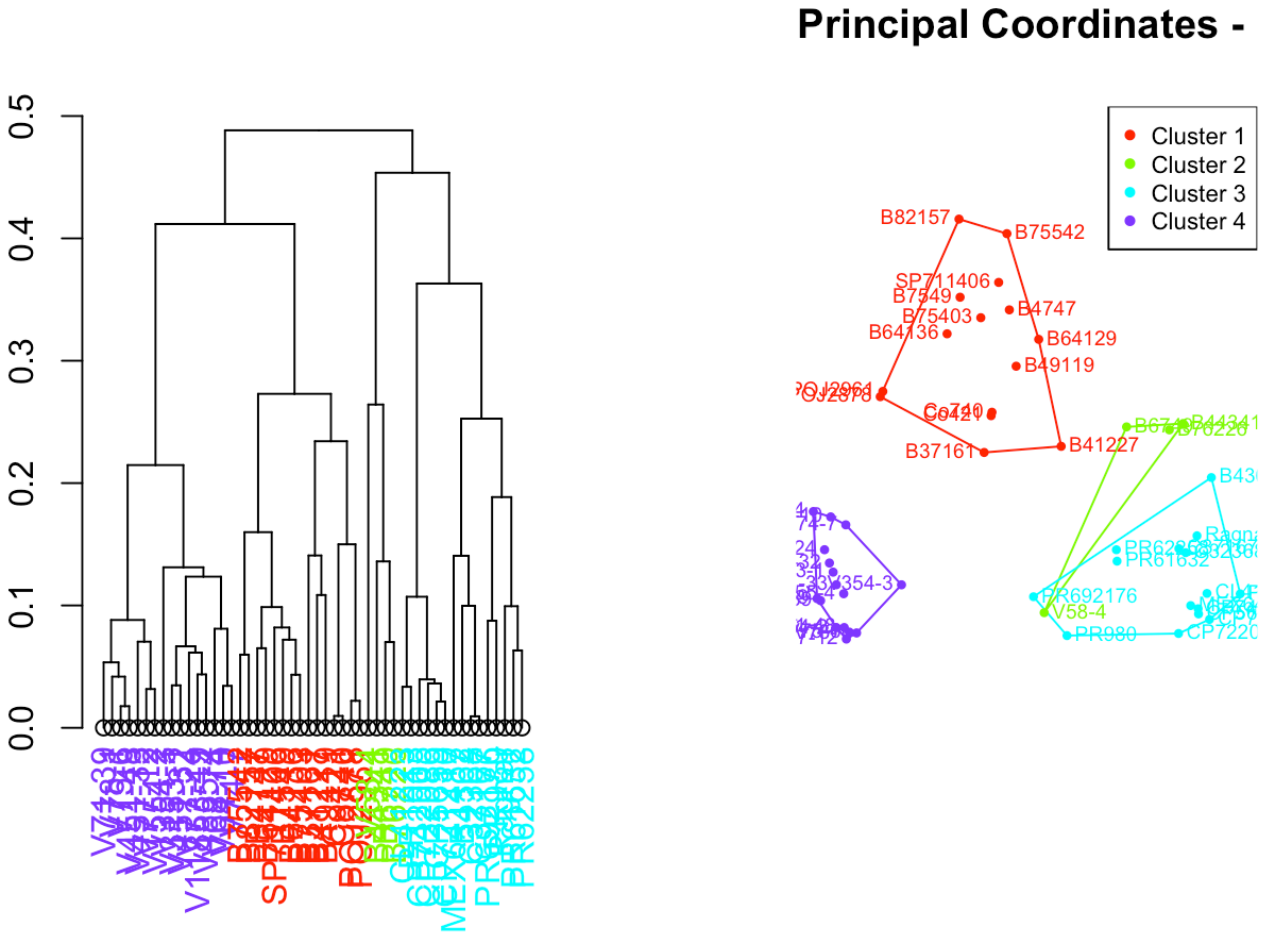
Los parámetros por defecto del método jerarquico se pueden cambiar, los argumentos son los de *hclust*.

```
pco=AddCluster2Biplot(pco, ClusterType="hi", NGroups=4, method="complete")
plot(pco, PlotClus=T)
```



Podemos incluso representar el mapa euclídeo y el dendrograma uno al lado del otro.

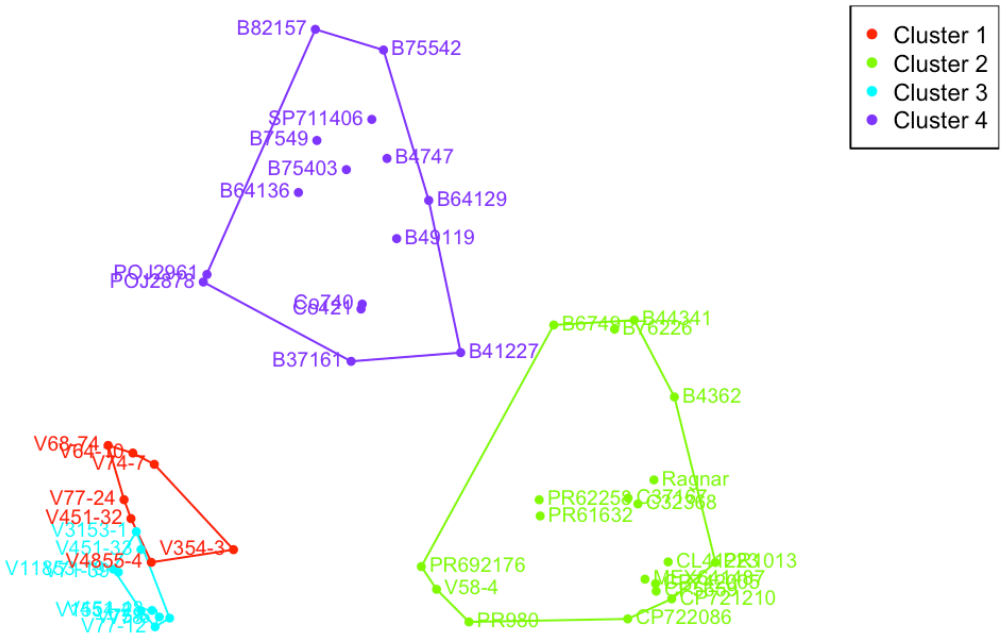
```
op <- par(mfrow = 1:2)
plot(pco$Dendrogram)
plot(pco, PlotClus=T, RowCex=0.6)
```




```
par(op)
```

De la misma manera, es posible introducir clusters derivados del método *k-medias*.

```
pco=AddCluster2Biplot(pco, ClusterType="km", NGroups=4)
plot(pco, PlotClus=T, RowCex=0.6)
```



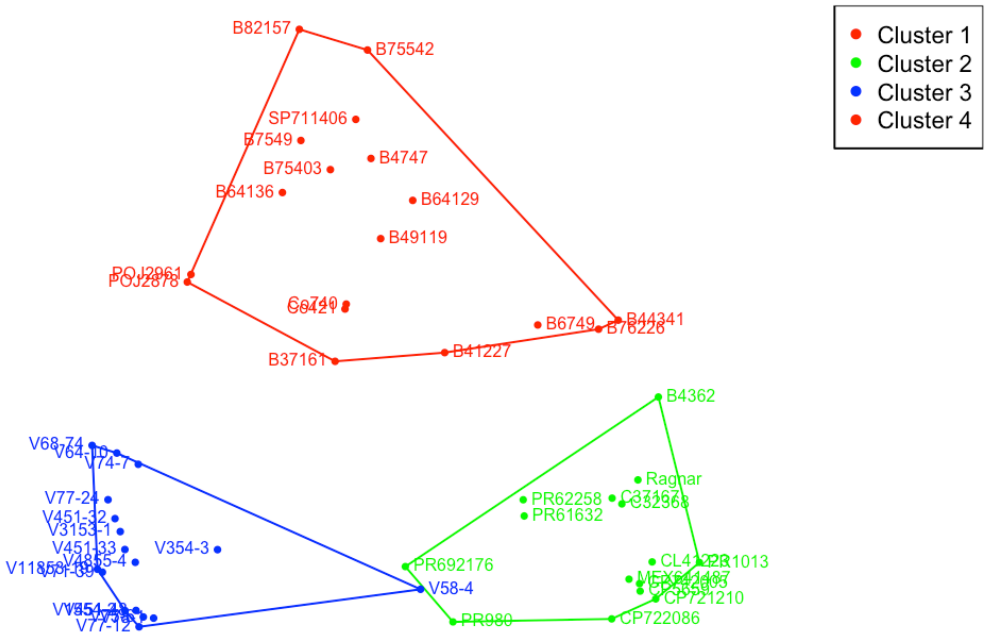
La representación obtenida es ligeramente diferente de las anteriores.

Podríamos también introducir clusters basados en mixturas de gausianas.

```
pco=AddCluster2Biplot(pco, ClusterType="gm", NGroups=3)
```

```
## [1] "1 0.573198 275.785576344"
## [1] "2 1.2993e-05 275.789159706"
## [1] "3 4.44e-07 275.789282259"
## [1] "4 1.9e-08 275.789287477"
```

```
plot(pco, PlotClus=T, RowCex=0.5)
```

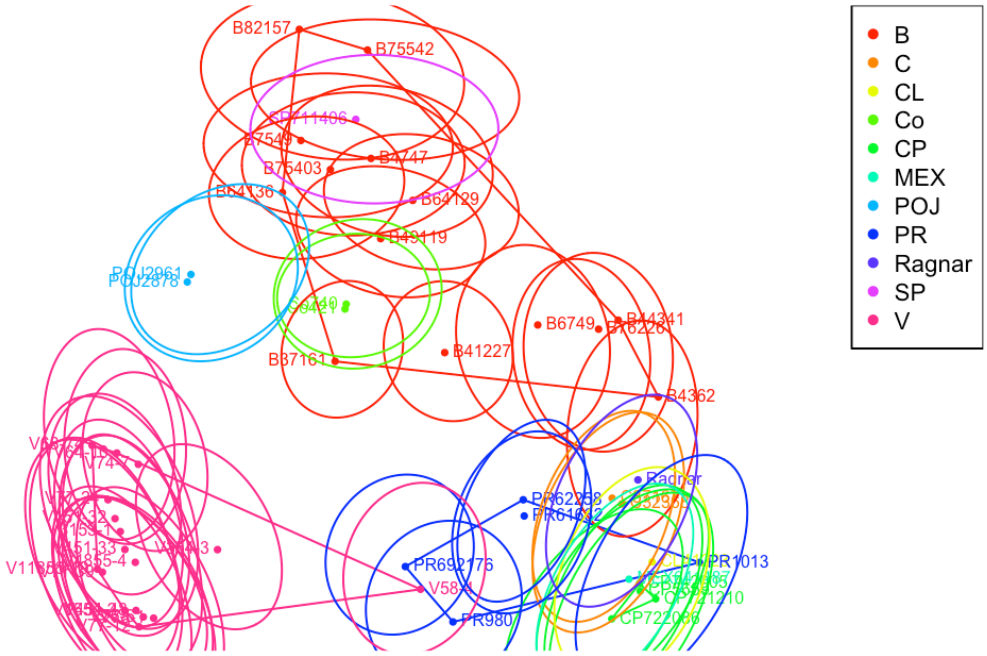


Podríamos también añadir medidas de incertidumbre sobre la posición de los puntos basadas en Bootstrap.

```
pco=PrincipalCoordinates(Dis, dimension = 2, Bootstrap = T)
```

```
## [1] 50
```

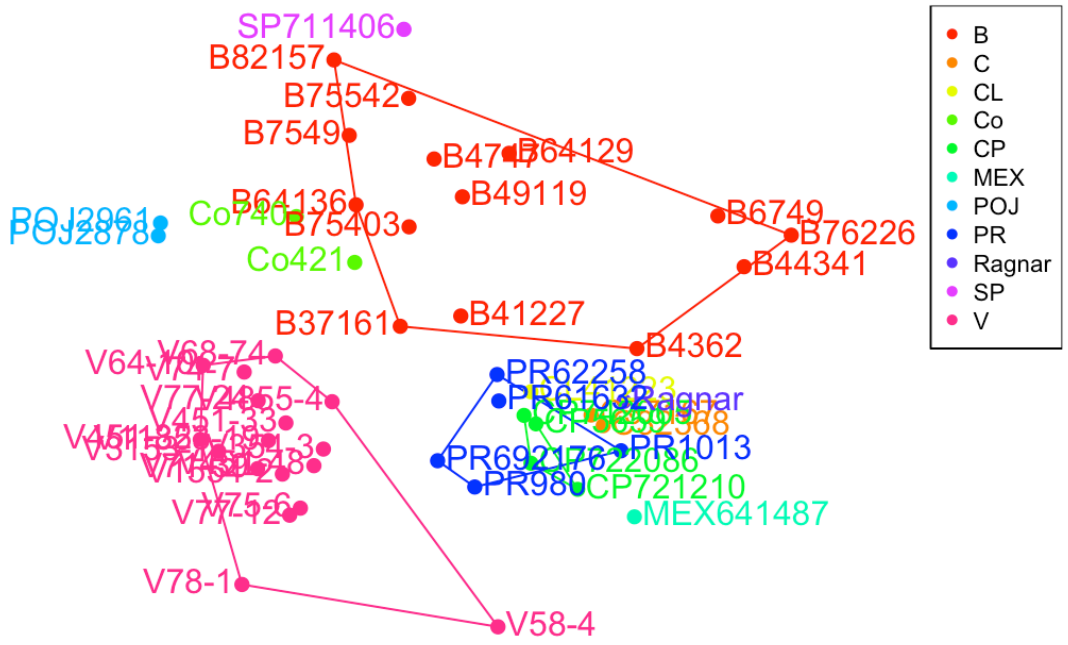
```
pco=AddCluster2Biplot(pco, ClusterType="us", Groups=Origin)
plot(pco, PlotClus=T, Bootstrap=T, RowCex=0.5)
```



Vemos que los distintos orígenes paracen estar bastante bien separados, corroborando la hipótesis de partida.

El paquete permite también la utilización de otras técnicas de Escalado Multidimensional (MDS), por ejemplo uno ordinal

```
sm=MDS(Dis, Model="Ordinal")
sm=AddCluster2Biplot(sm, ClusterType="us", Groups=Origin)
plot(sm, PlotClus=T)
```




```
summary(sm)
```

```
## [1] "PRINCIPAL COORDINATES ANALYSIS"
## [1] "Type of Data :  Binary"
## [1] "Type of Proximity :  dissimilarity"
## [1] "Coefficient :  Simple_Matching"
## [1] "Transformation :  Ordinal"
## [1] "-----------"
## [1] "-----------"
## [1] "COORDINATES"
##                   Dim 1         Dim 2
## B37161    -0.0009724544  1.665510e-03
## B4747      0.0009659857  1.124111e-02
## B44341     0.0187275998  5.074926e-03
## B41227     0.0024945362  2.254738e-03
## B4362      0.0125861288  3.883579e-04
## B49119     0.0025879173  9.077062e-03
## B64129     0.0052598948  1.156642e-02
## B6749      0.0172252200  7.981651e-03
## B64136    -0.0035070022  8.617704e-03
## B75542    -0.0004926013  1.471158e-02
## B7549     -0.0038962319  1.259407e-02
```

```
## B75403    -0.0004679201  7.356926e-03
## B76226     0.0214288851  6.880920e-03
## B82157    -0.0047716354  1.691132e-02
## SP711406  -0.0007666498  1.866223e-02
## C32368     0.0106553581 -4.022897e-03
## C37167     0.0099593701 -3.415143e-03
## Co421     -0.0035838767  5.320997e-03
## Co740     -0.0069245572  7.943341e-03
## CP5659     0.0068002714 -3.920116e-03
## CP721210   0.0091909842 -7.667434e-03
## CP742005   0.0061166107 -3.435434e-03
## CP722086   0.0065151151 -6.174767e-03
## CL41223    0.0065152267 -2.083463e-03
## MEX641487  0.0124427876 -9.239616e-03
## POJ2878   -0.0148354496  6.843929e-03
## POJ2961   -0.0147116659  7.577573e-03
## Ragnar     0.0118385883 -2.629217e-03
## PR1013     0.0116787338 -5.460352e-03
## PR61632    0.0046741709 -2.610057e-03
## PR62258    0.0045740791 -1.097380e-03
## PR692176   0.0011670497 -6.030378e-03
## PR980      0.0032943614 -7.533932e-03
## V58-4      0.0046245135 -1.553690e-02
## V64-10    -0.0122660322 -5.702782e-04
## V68-74    -0.0081323254 -3.388565e-05
## V71-39    -0.0091493265 -6.482921e-03
## V74-7     -0.0099154902 -9.531182e-04
## V75-6     -0.0066978170 -8.744414e-03
## V77-12    -0.0073086674 -9.146856e-03
## V77-24    -0.0091134735 -2.595933e-03
## V78-1     -0.0100430499 -1.311424e-02
## V451-48   -0.0059217970 -6.313159e-03
## V451-33   -0.0075281284 -3.868673e-03
## V3153-1   -0.0113772439 -5.488607e-03
## V451-32   -0.0124318414 -4.817532e-03
## V354-3    -0.0053775633 -5.362223e-03
## V1554-2   -0.0077267378 -6.794219e-03
## V11853-19 -0.0085243011 -4.874692e-03
## V4855-4   -0.0048795486 -2.652539e-03
## [1] "--------------------------------------"
##   RawStress     stress1     stress2    sstress1    sstress2         rsq
## 0.009019342 0.159294993 0.359462362 0.246601091 0.375862131 0.870786810
##    Spearman     Kendall
## 0.925558750 0.788295533
```

Los resultados son muy similares a los del PCoA ya que ambos son versiones distintas de la misma técnica.