Ordinación

Paula Vargas Pellicer

03/05/2022

En esta clase, aprenderemos a usar la ordinación para explorar patrones en conjuntos de datos ecológicos multivariados. Utilizaremos principalmente el paquete vegan para ver tres técnicas de ordinación (sin restricciones): Análisis de componentes principales (PCA), Análisis de coordenadas principales (PCoA) y Escalamiento multidimensional no métrico (NMDS).

Usaremos datos que están integrados dentro de los paquetes que estamos usando, por lo que no es necesario descargar archivos adicionales.

```
# paquetes (instala si no los tienes)
library(vegan)

library(ape)
library(dplyr)
```

¿Qué es la ordinación?

Objetivos de la ordinación

La ordinación es un término colectivo para las técnicas multivariadas que resumen un conjunto de datos multidimensionales de tal manera que cuando se proyecta en un espacio de baja dimensión (por ejemplo, en una grafica dos ejes), cualquier patrón intrínseco que puedan poseer los datos se hace evidente en la inspección visual.

En términos ecológicos: la ordinación resume los datos de la comunidad (como los datos de abundancia de especies: muestras por especie) mediante la producción de un espacio de ordenación de baja dimensión en el que las especies y muestras similares se trazan juntas, y las especies y muestras diferentes se colocan separadas. Ideal y típicamente, las dimensiones de este espacio dimensional representan gradientes ambientales que pueden ser interpretables.

En general, las técnicas de ordinación se utilizan en ecología para describir las relaciones entre los patrones de composición de especies y los gradientes ambientales subyacentes (por ejemplo, ¿qué variables ambientales estructuran la comunidad?). Dos ventajas muy importantes de la ordinación son que 1) podemos determinar la importancia relativa de diferentes gradientes y 2) los resultados gráficos de la mayoría de las técnicas a menudo conducen a interpretaciones fáciles e intuitivas de las relaciones entre especies y medio ambiente.

```r
# La base de datos que vamos a usar
data(varespec) # Cobertura vegetal de 44 especies
head(varespec)

# vamos a hacer un NMDS y graficar
varespec %>%
  metaMDS(trace = F) %>%
  ordiplot(type = "none") %>%
  text("sites")
```

De entrada, ya es más fácil encontrar patrones en la similitud de las muestras que en una base de datos.

Ordinación vs clasificación

La ordinación y la clasificación (o agrupación) son las dos clases principales de métodos multivariados que se emplean en la ecología de comunidades. Hasta cierto punto, estos dos enfoques son complementarios. La clasificación, o poner las muestras en clases (quizás jerárquicas), suele ser útil cuando se desea asignar nombres o mapear comunidades ecológicas. Sin embargo, dada la naturaleza continua de las comunidades, la ordinación puede considerarse un enfoque más natural. La ordinación tiene como objetivo organizar muestras o especies de forma continua a lo largo de gradientes.

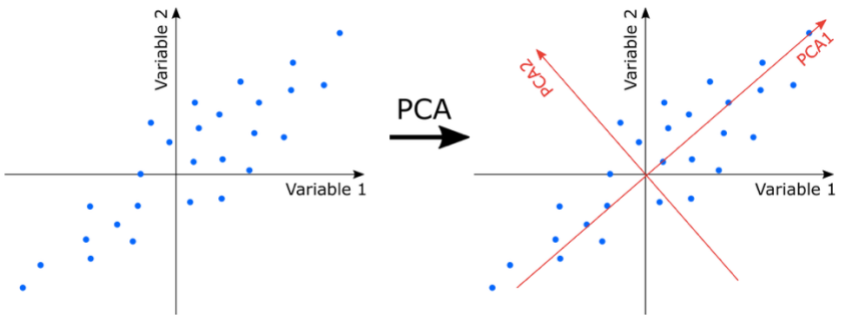
Diferentes técnicas de ordinación

Cómo y cuándo usar las tres técnicas principales (sin restricciones) de ordenación:

```
Análisis de componentes principales (PCA)
Análisis de coordenadas principales (PCoA)
Escalamiento multidimensional no métrico (NMDS)
```

Análisis de componentes principales (PCA)

PCA utiliza una rotación de los ejes originales para derivar nuevos ejes, que maximizan la varianza en el conjunto de datos. En 2D, esto se ve de la siguiente manera:



Computacionalmente, PCA es un análisis propio (eigenanalysis). Las consecuencias más importantes de esto son:

- Hay una solución única para el análisis propio (eigenvalue).
- Los ejes (también llamados componentes principales o PC) son ortogonales entre sí (y por lo tanto independientes).
- Cada PC está asociado con un valor propio (eigenvalue). La suma de los valores propios será igual a la suma de la varianza de todas las variables en el conjunto de datos.
- Los valores propios representan la varianza extraída por cada PC y, a menudo, se expresan como un porcentaje de la suma de todos los valores propios (es decir, la varianza total).
- Los valores propios relativos indican cuánta variación puede "explicar" una PC.
- Los ejes se clasifican por sus valores propios. Por lo tanto, el primer eje tiene el valor propio más alto y, por lo tanto, explica la mayor parte de la varianza, el segundo eje tiene el segundo valor propio más alto, etc.
- Existe una cantidad potencialmente grande de ejes (generalmente, la cantidad de muestras menos uno o la cantidad de especies menos uno, lo que sea menor). Sin embargo, el número de dimensiones que vale la pena interpretar suele ser muy bajo (por la existencia de algo llamado ruido).
- Las especies y las muestras se ordenan simultáneamente y, por lo tanto, ambas se pueden representar en el mismo diagrama de ordenación (si se hace esto, se denomina “biplot” o gráfica doble).
- Los datos de las variables en los PCA pueden entenderse como cuánto "contribuyó" cada variable a construir una PC. Se debe considerar el valor absoluto de las cargas ya que los signos son arbitrarios.

En la mayoría de las aplicaciones de PCA, las variables a menudo se miden en diferentes unidades. Por ejemplo, PCA de datos ambientales puede incluir pH, contenido de humedad del suelo, nitrógeno del suelo, temperatura, etc. Para ello, los datos deben estar estandarizados a media cero y varianza unitaria. Sin embargo, para la ordenación de comunidades ecológicas, todas las especies se miden en las mismas unidades y no es necesario estandarizar los datos.

Veamos cómo hacer un PCA en R.

```
PCA <- rda(varespec, scale = FALSE)
# Usa scale = TRUE si tus variables están en distintas escalas (ej. variables abióticas).
# Aquí, todas las especies se miden en la misma escala
```

```r
# haz una gráfica de barras de valores propios relativos. Esto va a dar el porcentaje de varianza explicada por cada eje
barplot(as.vector(PCA$CA$eig)/sum(PCA$CA$eig))
```

👉 ¿Cuánto de la varianza es explicada por el primer componente principal?

```r
# Calcula el porcentaje de varianza explicado por los primeros dos ejes
sum((as.vector(PCA$CA$eig)/sum(PCA$CA$eig))[1:2]) # 79%
```

👉 Ahora para los primeros tres ejes

```r
# grafica
plot(PCA)

plot(PCA, display = "sites", type = "points")

plot(PCA, display = "species", type = "text")


# Se pueden extraer los valores de especies y sitios del nuevo PC para otros análisis:
sitePCA <- PCA$CA$u # Sitios
speciesPCA <- PCA$CA$v # Especies

# En un grafica doble de PCA, las especies se dibujan como flechas que apuntan hacia la dirección del valor que incrementa para esa variable
biplot(PCA, choices = c(1,2), type = c("text", "points"), xlim = c(-5,10))
# gráfica doble de los ejes 1 vs 2

biplot(PCA, choices = c(1,3), type = c("text","points"))

# gráfica doble de los ejes 1 vs 3
```

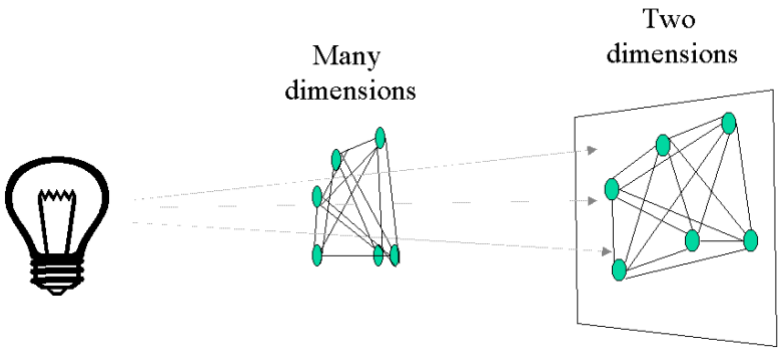
En contraste con algunas de las otras técnicas de ordinación, las especies están representadas por flechas. Esto implica que la abundancia de la especie aumenta continuamente en la dirección de la flecha y disminuye en la dirección opuesta. Por lo tanto, PCA es un método lineal. PCA es útil cuando esperamos que las especies estén linealmente (o incluso monotónicamente) relacionadas entre sí. Desafortunadamente, rara vez nos encontramos con una situación así en la naturaleza. Es mucho más probable que las especies tengan una curva de respuesta de especie unimodal:

Esta suposición lineal hace que PCA sufra un problema grave, el efecto de herradura o arco, que lo hace inadecuado para la mayoría de los conjuntos de datos ecológicos. La solución PCA a menudo se distorsiona en forma de herradura/arco (con la punta hacia arriba o hacia abajo) si la diversidad beta es de moderada a alta. La herradura puede aparecer aunque exista un importante gradiente secundario.

👉 ¿Puedes detectar una forma de herradura en la gráfica?

Análisis de coordenadas principales (PCoA)

El análisis de coordenadas principales (PCoA, también conocido como escalado multidimensional métrico) intenta representar las distancias entre muestras en un espacio euclidiano de baja dimensión. En particular, maximiza la correlación lineal entre las distancias en la matriz de distancias y las distancias en un espacio de baja dimensión (típicamente, se seleccionan 2 o 3 ejes).



El primer paso de un PCoA es la construcción de una matriz de (des)similitud. Si bien PCA se basa en distancias euclidianas, PCoA puede manejar matrices de (des)similitud calculadas a partir de variables cuantitativas, semicuantitativas, cualitativas y mixtas. Como siempre, la elección de la medida de (des)similitud es crítica y debe ser adecuada a los datos en cuestión. Para datos de abundancia, a menudo se recomienda la distancia de Bray-Curtis. Puedes utilizar el índice Jaccard para datos de presencia/ausencia. Cuando la métrica de distancia es euclidiana, PCoA es equivalente al análisis de componentes principales. Aunque PCoA se basa en una matriz de (des)similitud, la solución se puede encontrar mediante análisis propio. La interpretación de los resultados es la misma que con PCA.

```
# Calcular la matriz de distancias
# Bray-Curtis
dist <- vegdist(varespec,  method = "bray")

# PCoA no está incluido en vegan.
library(ape)
PCOA <- pcoa(dist)

# graficar
barplot(PCOA$values$Relative_eig[1:10])

# Algunas distancias pueden terminar en valores propios negativos. Se pueden
corregir:
PCOA <- pcoa(dist, correction = "cailliez")

# gráfica
biplot.pcoa(PCOA)
```

```r
# Aquí no se grafican las especies pues la matriz de distancia lo hace sitio por sitio
#pero podemos hacer los siguiente:
biplot.pcoa(PCOA, varespec)

# Se pueden extraer los primeros dos ejes, si lo necesitas para análisis estadísticos
PCOAaxes <- PCOA$vectors[,c(1,2)]

# comparar los resultados con PCA
biplot.pcoa(PCOA)

plot(PCA)
```

PCoA tiene varias fallas, en particular el efecto arco. Estos defectos se derivan, en parte, del hecho de que PCoA maximiza una correlación lineal. El escalamiento multidimensional no métrico (NMDS) rectifica esto maximizando la correlación del orden de clasificación.

Escalamiento multidimensional no métrico (NMDS)

NMDS intenta representar la disimilitud por pares entre objetos en un espacio de baja dimensión. Se puede utilizar cualquier coeficiente de disimilitud o medida de distancia para construir la matriz de distancia utilizada como entrada. NMDS es un enfoque basado en rangos; esto significa que los datos de distancia originales se sustituyen por rangos. Por lo tanto, en lugar de que el objeto A esté a 2,1 unidades de distancia del objeto B y a 4,4 unidades del objeto C, el objeto C es el "primero" más distante del objeto A, mientras que el objeto C es el "segundo" más distante. Si bien se pierde información sobre la magnitud de las distancias, los métodos basados en rangos son generalmente más sólidos para los datos que no tienen una distribución identificable.

NMDS es un algoritmo iterativo. Las rutinas de NMDS a menudo comienzan con la colocación aleatoria de objetos de datos en el espacio de ordenación. Luego, el algoritmo comienza a refinar esta ubicación mediante un proceso iterativo, intentando encontrar una ordenación en la que las distancias de los objetos ordenados coincidan estrechamente con el orden de las diferencias de objetos en la matriz de distancias original. El valor de tensión refleja qué tan bien la ordenación resume las distancias observadas entre las muestras.

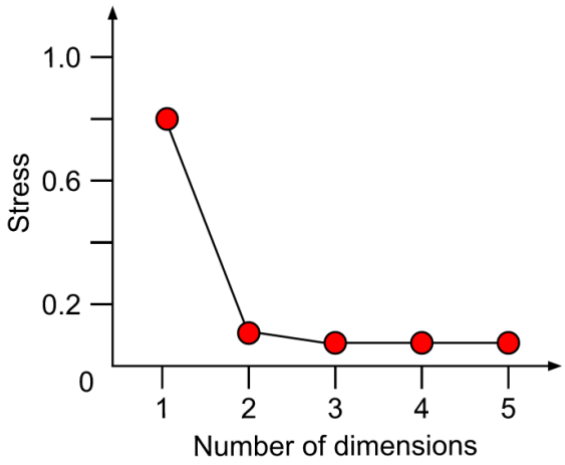
NMDS no es un análisis propio. Esto tiene tres consecuencias importantes:

- No hay un resultado de ordenación único.
- Los ejes de la ordenación no están ordenados según la varianza que explican
- El número de dimensiones del espacio de baja dimensión debe especificarse antes de ejecutar el análisis.

No hay una solución única. La solución final depende de la colocación aleatoria de los objetos en el primer paso. Es necesario ejecutar el algoritmo NMDS varias veces para garantizar que la ordenación sea estable, ya que cualquier ejecución puede quedar "atrapada" en los valores óptimos locales que no son representativos de las distancias reales. Nota: esto se hace automáticamente con metaMDS() en vegan.

Los ejes no se ordenan en NMDS. metaMDS() en vegan rota automáticamente el resultado final del NMDS usando PCA para hacer que el eje 1 corresponda a la mayor variación entre los puntos de muestra del NMDS. Esto no cambia la interpretación, no se puede modificar y funciona muy bien, pero debes siempre tenerlo en cuenta.

Se puede utilizar un gráfico de tensión (una medida de bondad de ajuste) frente a dimensionalidad para evaluar la elección adecuada de las dimensiones. Los propios valores de tensión pueden utilizarse como indicador. Los valores de estrés >0.2 son generalmente malos y potencialmente imposibles de interpretar, mientras que los valores <0.1 son buenos y <0.05 son excelentes, dejando poco peligro de mala interpretación. Se pueden utilizar valores de tensión entre 0.1 y 0.2, pero algunas de las distancias serán engañosas. Encontrar el punto de inflexión puede indicar la selección de un número mínimo de dimensiones.



Metodología de NMDS:

```
Paso 1: Realiza NMDS con 1 a 10 dimensiones
Paso 2: Verifica el gráfico de tensión vs dimensión
Paso 3: Elije el número óptimo de dimensiones
Paso 4: Realiza el NMDS final con ese número de dimensiones
Paso 5: Comprueba la solución convergente y la tensión final


# Calcular la matriz de distancias
dist <- vegdist(varespec,  method = "bray")
```

```
# NMDS.scree() automáticamente hace un NMDS para 1-10 dimensiones y grafica
NMDS.scree <- function(x) { # x es el nombre de la variable
  plot(rep(1, 10), replicate(10, metaMDS(x, autotransform = F, k = 1)$stress)
, xlim = c(1, 10),ylim = c(0, 0.30), xlab = "# de Dimensiones", ylab = "Tensi
on", main = "NMDS")
  for (i in 1:10) {
    points(rep(i + 1,10),replicate(10, metaMDS(x, autotransform = F, k = i +
1)$stress))
  }
}

# Usa la función que acabamos de crear para elegir el número óptimo de dimens
iones

NMDS.scree(dist)
```

Vemos en este gráfico que la tensión disminuye con el número de dimensiones. Este es un comportamiento normal de un gráfico de tensión.

```
set.seed(2)

# Vamos a hacer los resultados de dos formas, usando la matriz de distancias
y usando los datos "crudos":


NMDS1 <- metaMDS(dist, k = 2, trymax = 100, trace = F)
NMDS1


NMDS2 <- metaMDS(varespec, k = 2, trymax = 100, trace = F)

# Si no le das una matriz de similitud, metaMDS automáticamente aplica
Bray-Curtis.
NMDS2
```

¿Ves que los resultados no son los mismos? metaMDS() calculó las distancias de Bray-Curtis, pero primero aplicó una transformación de raíz cuadrada en la matriz de la comunidad.


Verifica el archivo de ayuda para metaNMDS() e intente adaptar la función para NMDS2, de modo que se desactive la transformación automática.

```
NMDS3 <-
plot(NMDS3)
```

No hay puntajes de especies (el mismo problema que encontramos con PCoA). Podemos solucionar este problema dando a metaMDS la matriz comunitaria original como entrada y especificando la medida de la distancia.

```r
plot(NMDS3, display = "sites", type = "n")
points(NMDS3, display = "sites", col = "red", cex = 1.25)
text(NMDS3, display ="species")

# Usando ordiplot y orditorp
ordiplot(NMDS3, type = "n")
orditorp(NMDS3, display = "species", col = "red", air = 0.01)
orditorp(NMDS3, display = "sites", cex = 1.1, air = 0.01)
```

Interpretación de los resultados

Ahora tenemos una buena grafica de ordinación y sabemos qué graficas tienen una composición de especies similar. También sabemos que el primer eje de ordinación corresponde al gradiente más grande de nuestra base de datos (el gradiente que explica la mayor variación en nuestros datos), el segundo eje al segundo gradiente más grande y así sucesivamente. La siguiente pregunta es: ¿Qué variable ambiental está impulsando las diferencias observadas en la composición de especies? Podemos hacerlo correlacionando las variables ambientales con nuestros ejes de ordenación. Por lo tanto, utilizaremos un segundo conjunto de datos con variables ambientales (muestra por variables ambientales). Seguimos utilizando los resultados del NMDS.

```r
# Cargamos la otra base de datos (ambientales)
data(varechem)

# envfit va a agregar datos de variables ambientales como vectores de
ordinación
ef <- envfit(NMDS3, varechem, permu = 999)
ef

# Las últimas dos columnas nos interesan: el coeficiente de correlación (cuad
rada) y el valor-p asociado


# grafica los vectores con correlaciones significativas e interpreta la gráfi
ca

plot(NMDS3, type = "t", display = "sites")
plot(ef, p.max = 0.05)
```

A continuación, digamos que tenemos dos grupos de muestras. Esto podría ser el resultado de una clasificación o simplemente de dos grupos predefinidos (por ejemplo, bosques viejos versus bosques jóvenes o dos tratamientos). Ahora, queremos ver los dos grupos en el diagrama de ordenación. Así es como lo haces:

```r
# Define la variable que agrupa (las primeras 12 muestras son de grupo 1, las
ultimas 12, de grupo 2
group = c(rep("Group1", 12), rep("Group2", 12))

# Un vector de colores con la misma dimensión que los grupos
```

```r
colors = c(rep("red", 12), rep("blue", 12))

# Polígonos de colores que corresponden a los grupos
ordiplot(NMDS3, type = "n")
for(i in unique(group)) {
  ordihull(NMDS3$point[grep(i, group),], draw="polygon",
   groups = group[group == i],col = colors[grep(i,group)],label=F) }

orditorp(NMDS3, display = "species", col = "red", air = 0.01)
orditorp(NMDS3, display = "sites", col = c(rep("red",12),
  rep("blue", 12)), air = 0.01, cex = 1.25)
```

Ejercicio

Realiza un análisis de ordinación con la base de datos de dunas (dune) proporcionado por el paquete vegan. Interpreta tus resultados usando las variables ambientales de dune.env.