Practico 5

En este práctico vamos a realizar los primeros análisis exploratorios sobre la estructura de las comunidades y su diversidad. Más concretamente, estaremos comparando las comunidades de microorganismos en los dos sistemas muestreados: laguna y arroyo. Para esto, vamos utilizar las abundancias de ASVs y su anotación taxonómica previamente computadas.

Las secciones de este tutorial son las siguientes:

1. Carga de librerías y datos
2. Formateo y limpieza de datos
3. Visualización de abundancias relativas
4. Análisis de coordenadas principales

---

1. Carga de librerías y datos

Empezamos con las librerías.

```
library(tidyverse)
library(vegan)
library(indicspecies)
```

Y luego los datos: la tabla de abundancia de ASVs, la cual incluímos además la anotación taxonómica de cada ASV (como recordarás, creada en el práctico 3) y la metadata asociada a las muestras (esta última cortesía de Griffero et al.).

```
ASV2TAX <- read_csv("data/asv_abund_annot.csv", col_names = T)
URL <- "https://raw.githubusercontent.com/pereiramemo/Curso-Metabarcoding-2023/main/data/metadata.tsv"
METADATA <- read_tsv(URL, col_names = T)
```

Puedes visualizar las tablas con la función View.

---

2. Formateo y limpieza de datos.

En esta parte vamos a formatear y limpiar los datos para facilitar los análisis posteriores. Como habrás notado, los nombres de las muestras no son muy informativos. Vamos a cambiar los nombres por un código de muestras que tenemos en la metadata, donde se incluyen los siguientes campos [Sistema]_[Mes]_[Número de muestra]. En estos códigos, las lagunas aparecen como LA y los arroyos como ST.

```
col_n <- dim(ASV2TAX)[2]

sample_names <- sub(x = colnames(ASV2TAX[,15:col_n]),
```

```
                    pattern = "_R1_filt.fastq.gz",
                    replacement = "") %>%
                as.data.frame() %>%
                left_join(x = .,
                          y = METADATA %>% select(Site, amplicon_code),
                          by = c("." = "amplicon_code"))


colnames(ASV2TAX)[15:col_n] <- sample_names$Site
```

Otra tarea importante en esta primera etapa, es eliminar los ASVs anotados como mitocondrias o cloroplastos.

```
i_c <- grep(x = ASV2TAX$tax.Class, pattern = "Chloroplast", value = F)
i_o <- grep(x = ASV2TAX$tax.Order, pattern = "Chloroplast", value = F)
i_f <- grep(x = ASV2TAX$tax.Family, pattern = "Chloroplast", value = F)
j_c <- grep(x = ASV2TAX$tax.Class, pattern = "Mitochondria", value = F)
j_o <- grep(x = ASV2TAX$tax.Order, pattern = "Mitochondria", value = F)
j_f <- grep(x = ASV2TAX$tax.Family, pattern = "Mitochondria", value = F)

i <- unique(c(i_c, i_o, i_f, j_c, j_o, j_f))
length(i)
ASV2TAX_clean <- ASV2TAX[-i,]
```

---

3. Visualización de abundancias relativas.

Como primera exploración, vamos a crear un plot de barras representando la abundancia de todos los fila que tenemos en nuestros datos. Para esto, primero vamos a seleccionar todos los ASVs que tuvieron una anotación taxonómica a nivel de fila con un *bootstrap* a mayor a 80%.

```
ASV2TAX_redu <- ASV2TAX_clean %>%
  filter(is.na(tax.Phylum) == F & boot.Phylum >= 80)
```

Luego, vamos a seleccionar las columnas que precisamos de la tabla `ASV2TAX_redu` y convertir esta subtabla a formato *tidy* (en relación a las muestras).

```
samples <- colnames(ASV2TAX_redu)[15:col_n]
ASV2TAX_redu_long <- ASV2TAX_redu %>%
  select(asv, tax.Phylum, samples) %>%
  pivot_longer(cols = samples, names_to = "Site", values_to = "abund")
```

Un pasito más: vamos a computar las abundancias relativas de cada filo y crearnos una nueva columna para los nombres de los fila. La finalidad de esta columna es tener representados únicamente los fila con una abundancia mayor a 2%. Todo lo demás lo agrupas en la categoría *other phyla*.

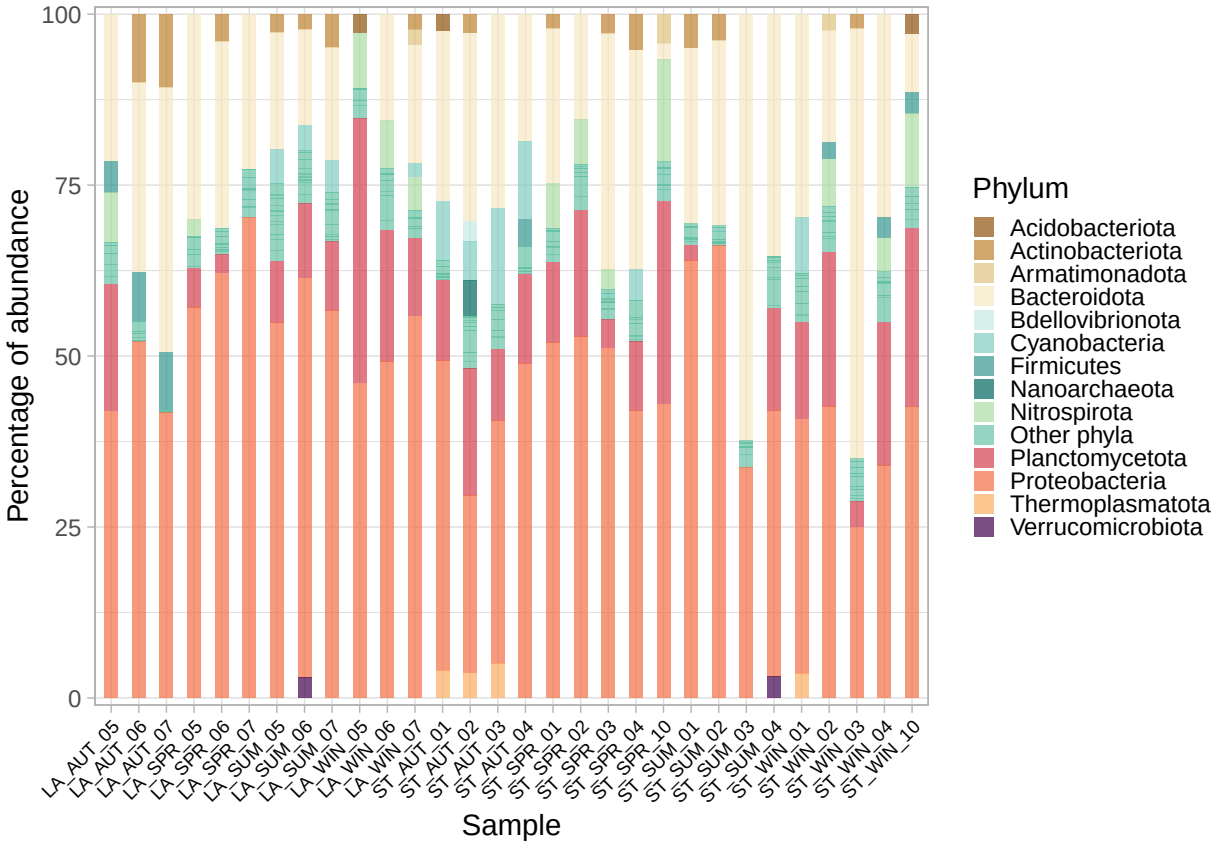
```
ASV2TAX_redu_long_ready2plot <- ASV2TAX_redu_long %>%
  group_by(Site, tax.Phylum) %>%
  summarize(abund_phylum = sum(abund)) %>%
  ungroup() %>%
  group_by(Site) %>%
  mutate(abund_phylum_rel = (abund_phylum/sum(abund_phylum))*100 ) %>%
  mutate(phylum2 = if_else(abund_phylum_rel > 2,
                           tax.Phylum, "Other phyla"))
```

Por último, vamos a visualizar la abundancia relativa de cada fila en cada muestra.

```
phyla_colors <- c('#8c510a','#bf812d','#dfc27d','#f6e8c3','#c7eae5','#80cdc1',
                  '#35978f','#01665e','#abdda4','#66c2a5', '#d53e4f','#f46d43',
                  '#fdae61','#40004b', '#762a83', '#9970ab',"gray40","gray80")
text_size <- 10

pp <- ggplot(ASV2TAX_redu_long_ready2plot, aes(x = Site,
                                               y = abund_phylum_rel,
                                               fill = phylum2)) +
            geom_bar(stat = "identity", width = 0.5, alpha = 0.7) +
            xlab("Sample") +
            ylab("Percentage of abundance") +
            scale_fill_manual(name="Phylum", values = phyla_colors) +
            scale_y_continuous(expand=c(0,0.1), limits = c(-1,101)) +
            theme_light() +
            theme(
                 axis.text.x = element_text(size = text_size -3, angle = 45,
                                            hjust = 1, color = "black"),
                 strip.background = element_blank(),
                 strip.text = element_text(color = "black", size = text_size)) +
                 guides(fill = guide_legend(keywidth = 0.6, keyheight = 0.6)) # +

pp
```



¿Ves alguna diferencia en la abundancia relativa entre lagunas y arroyos? ¿Puede hacer este mismo plot para otros niveles taxonómicos?

4. Análisis de coordenadas principales.

En esta sección vamos a realizar un análisis de coordenadas principales (PCoA) sobre las disimilitudes de Bray-Curtis entre muestras, rarefaccionadas y transformadas con la transformación de Hellinger.

Para realizar este análisis, tenemos que crearnos una tabla donde sólo tengamos las abundancias de ASVs. Es decir, a nuestra tabla `ASV2TAX_clean`, vamos a quitarle la columnas de anotación taxonómica (incluyendo los bootstraps).

```
col_n <- dim(ASV2TAX_clean)[2]

ASV <- ASV2TAX_clean[,c(2,15:col_n)] %>%
       column_to_rownames("asv") %>%
       t() %>%
       as.data.frame()
```

Nuevamante, puedes (y es recomendable) inspeccionar la tabla que generamos con la función View.

Otro control mínimo es ver sus dimensiones.

```
dim(ASV)
```

```
## [1]  30 964
```

30 muestras con 964 ASVs. Tiene sentido.

Como podemos ver, esta tabla de abundancias tiene un formato de muestras por ASVs (no es en formato *tidy*, el cual es muy útil, pero no es siempre aceptado para trabajar con otros paquetes fuera de `tidyverse`).

En el siguiente paso vamos a rarificar la tabla `ASV`. Para esto es necesario determinar la menor abundancia por muestra en nuestro set de datos.

```
sample_min <- rowSums(ASV) %>% min()
sample_min
```

```
## [1] 477
```

Vamos a rarificar todas las muestras a 477 *reads* por muestra (es bastante bajo, pero recuerda que es un set de datos de juguete). El paquete `vegan` incluye una función para esa tarea: rrarefy.

```
set.seed(123)
ASV_rare <- rrarefy(x = ASV, sample = sample_min)
```

¿Qué sentido tiene utilizar la función `set.seed` antes de correr la función `rrarefy`?
Una vez que rarificamos nuestra tabla, vamos a aplicar la transformación de Hellinger con la función decostand, también del paquete `vegan`.

```
ASV_rare_trans <- decostand(ASV_rare, method = "hellinger")
```

¿Conoces la transformación de Hellinger? Acá puedes leer sobre ésta y otras transformaciones relevantes. Respuesta rápida: con esta transformación estamos reduciendo el peso que tienen los ASVs muy abundantes al computar las disimilitudes entre muestras.

Ahora que tenemos nuestra tabla de abundancia pronta vamos a computar la disimilaridad de Bray-Curtis entre muestras, donde estaremos aplicando la función vegdist del paquete `vegan`.

```
ASV_rare_trans_diss <- vegdist(ASV_rare_trans, method = "bray")
```

Ya tenemos todo listo para correr el PCoA. Nuevamente, el paquete `vegan` tiene una función para esto: cmdscale.

```
ASV_pcoa <- cmdscale(ASV_rare_trans_diss, k = 4, eig = T)
```

¿Puedes determinar qué significa el argumento `k` en la función `cmdscale`? El objeto `AVS_pcoa` tiene en el atributo `points` las coordenadas de nuestras 30 muestras en los ejes principales.
Lo que vamos a hacer en el siguiente comando es crearnos una tabla donde las coordenadas principales y la metadata asociada que cargamos al comienzo del práctico.

```
ASV_pcoa_ext <- ASV_pcoa$points %>%
  as.data.frame() %>%
  rownames_to_column("Site") %>%
  left_join(x = ., y = METADATA, by = "Site")
```

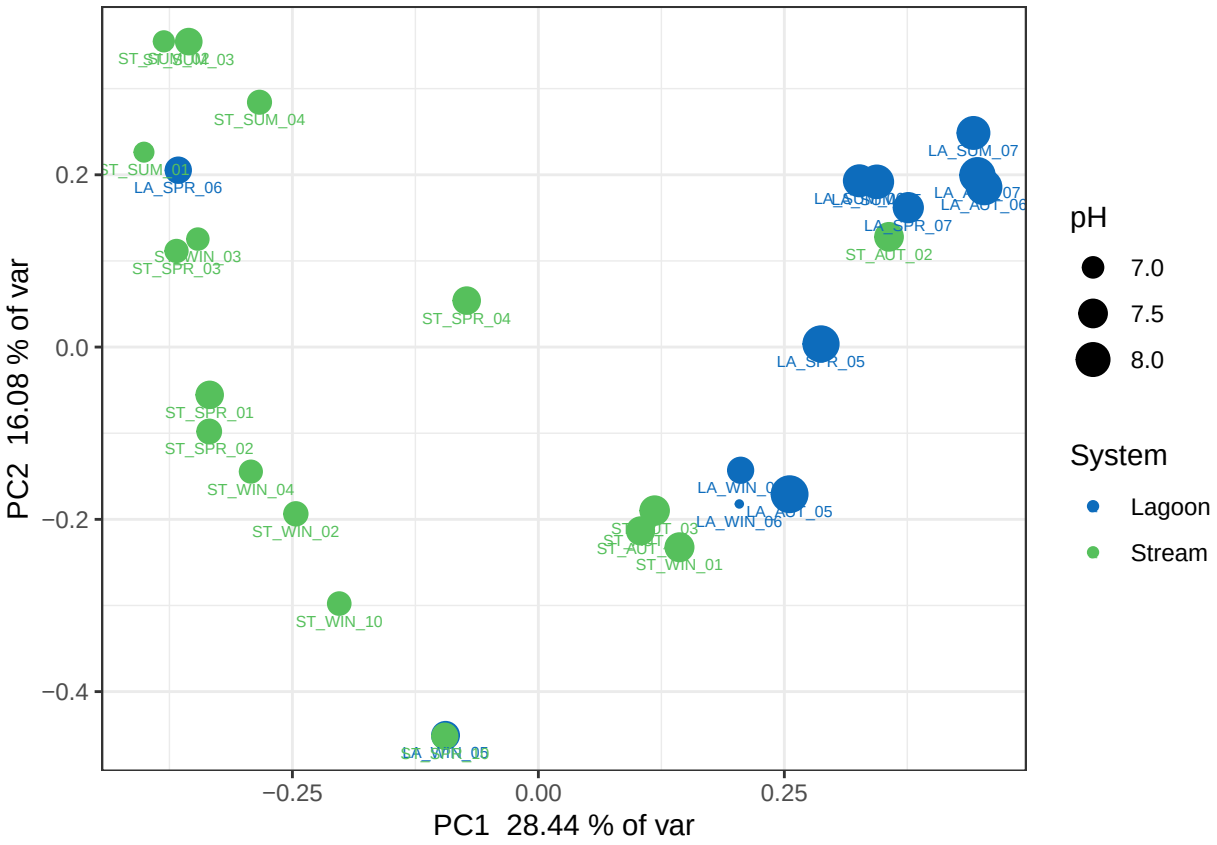
Como siguiente paso, vamos a determinar la variabilidad que es explicada por cada eje.

```
x_title <- paste("PC1 ", (ASV_pcoa$eig/sum(ASV_pcoa$eig))[1] %>% round(4) * 100, "% of var")
y_title <- paste("PC2 ", (ASV_pcoa$eig/sum(ASV_pcoa$eig))[2] %>% round(4) * 100, "% of var")
```

Ya tenemos todo para crear el plot.

```
system_colors <- c("#0d6cbc","#56C05C")
pcoa_plot <- ggplot(ASV_pcoa_ext,
                    aes(x = V1, y = V2, color =
                          System, size = pH )) +
             geom_point() +
             geom_text(aes(label = Site), vjust = 2, size = 2) +
             scale_color_manual(values = system_colors) +
             xlab(x_title) +
             ylab(y_title) +
             theme_bw()

pcoa_plot
```



¿Puedes verificar si existe una correlación entre el pH y el PCo1?