Análisis de datos ómicos (M0-157) Primera prueba de evaluación continua

Presentación y objetivos

Esta PEC completa la introducción a las ómicas mediante un ejercicio de repaso y ampliación que nos permite trabajar con algunas de las herramientas que hemos introducido durante el curso: Bioconductor y la exploración multivariante de datos.

Para llevarla a cabo, previamente debéis estar familiarizados con:

- Las tecnologías ómicas.
- Bioconductor y las clases que se utilizan para almacenar datos ómicos, como los expressionSets.
- Git como herramienta de control de versiones, así como GitHub.
- Las herramientas estadísticas de exploración de datos introducidas al final del primer reto.

El objetivo de esta PEC es que planifiquéis y ejecutéis una versión simplificada del proceso de análisis de datos ómicos, a la vez que practicáis con algunas de las herramientas y métodos que hemos trabajado durante el primer reto.

Descripción de la PEC

A continuación, se detallan las tareas que debéis llevar a cabo:

1. Seleccionad y descargad un dataset de metabolómica, que podéis obtener de *metabolomicsWorkbench* o de este repositorio de GitHub.

2. Cread un objeto de clase SummarizedExperiment que contenga los datos y los metadatos (información acerca del dataset, sus filas y columnas). La clase SummarizedExperiment es una extensión de ExpressionSet, utilizada por muchas aplicaciones y bases de datos (como es el caso de metabolomicsWorkbench). ¿Cuáles son sus principales diferencias con la clase ExpressionSet?

3. Llevad a cabo un análisis exploratorio que os proporcione una visión general del dataset en la línea de lo que hemos visto en las actividades de este reto.

4. Elaborad un informe que describa el proceso que habéis realizado, incluyendo la justificación de la selección del dataset, su incorporación al summarizedExperiment, el análisis exploratorio de los datos y la interpretación de los resultados desde el punto de vista biológico. La extensión máxima de este informe (sin tener en cuenta los Anexos) debe ser de 10 páginas, en formato PDF.

5. Cread un repositorio de GitHub que contenga:

- el informe,
- el objeto de clase SummarizedExperiment que contenga los datos y los metadatos en formato binario (.Rda),
- el código R para la exploración de los datos debidamente comentado (el control de versiones del mismo debe realizarse con Git)
- los datos en formato texto y
- los metadatos acompañados de una breve descripción en un archivo markdown.

El nombre del repositorio debe ser 'Apellido1-Apellido2-Nombre-PEC1'. La dirección (URL) del repositorio deberá estar incluida en el informe de manera clara. Tened en cuenta que a través de CANVAS debéis entregar únicamente el informe. Los recursos para llevar a cabo la PEC son los que se han proporcionado en las distintas actividades del primer reto. Además, deberéis familiarizaros con la clase SummarizedExperiment. Para ello podéis utilizar este tutorial de Bioconductor.

Informe

Las principales diferencias entre SummarizedExperiment y ExpressionSet es que este segundo esta más enfocado al análisis de microarrays, por lo que está más limitado a datos genómicos. Mientras que SummarizedExperiment permite el análisis de otros datos ómicos, además de poder analizar simultáneamente varios datasets omicos a la vez (transcriptómica, metabolómica, etc).

Para llevar a cabo la PEC1 se ha elegido el dataset de cachexia, en el que podemos encontrar dos grupos, el grupo control y el grupo con cachexia. En primer lugar, se cargaron los paquetes necesarios para el manejo de datos ómicos (SummarizedExperiment) y para el análisis de bioestadística y visualización (psych, ggplot2, pheatmap). A continuación, se subió el dataset con la función read_csv para poder construir un dataframe y así poder empezar con la creación del objeto de clase SummarizedExperiment:

```
# cargar paquetes necesarios
library(SummarizedExperiment)
library(readr)
library(tibble)
library(psych)
library(dplyr)
library(ggplot2)
library(knitr)
library(pheatmap)

# cargar dataset metabolomica
df <- read_csv("C:/Users/soliz-j/Downloads/Master/human_cachexia.csv")

# implementar una seed para crear el objeto
set.seed(1234)

# obtener información del dataset
assay_matrix <- as.matrix(df[, -(1:2)])  # eliminar columnas ID y grupo
rownames(assay_matrix) <- df$`Patient ID`  # cada fila = muestra
assay_matrix <- t(assay_matrix)

# crear colData del dataset
col_data <- DataFrame(
  PatientID = df$`Patient ID`,
  Group = factor(df$`Muscle loss`)  # cachexic / control
)
rownames(col_data) <- df$`Patient ID`

# crear rowData para los metabolitos
```

```
row_data <- DataFrame(
  Metabolite = rownames(assay_matrix)
)
rownames(row_data) <- row_data$Metabolite

# cargar descripción del dataset desde description.md
description_lines <- readLines("C:/Users/soliz-j/Downloads/Master/description.md")
description_text <- paste(description_lines, collapse = "\n")

# crear objeto de clase SummarizedExperiment
se <- SummarizedExperiment(
  assays = list(metabolites = assay_matrix),
  colData = col_data,
  rowData = row_data,
  metadata = list(description = description_text)
)

# visualizar resumen del objeto
se
save(se, file = "C:/Users/soliz-j/Downloads/Master/mi_objeto.Rda")
```

Para la creación del objeto se encapsuló la siguiente información:

- assay: matriz de expresión de metabolitos
- colData: anotaciones de las muestras (grupo experimental, ID)
- rowData: nombres e IDs de los metabolitos
- metadata: descripción textual del experimento extraída de un description.md

A continuación, se detalla las funciones para poder hacer un análisis descriptivo del dataset y de los datos que alberga. Los resultados se pueden consultar en el documento Markdown adjunto en el repositorio de la prueba:

```
# visualizar metadata de las muestras
colData(se)

# visualizar metabolitos
rowData(se)

# visualizar descripciones generales del experimento
metadata(se)

# preparacion de datos para análisis exploratorio
matriz <- assay(se)
grupos <- colData(se)$Group
names(grupos) <- rownames(colData(se))
```

```
df_long <- as.data.frame(t(matriz))
df_long$Sample <- rownames(df_long)
df_long$Group <- grupos[rownames(df_long)]

df_long <- df_long |>
  pivot_longer(cols = -c(Sample, Group),
               names_to = "Metabolite",
               values_to = "Value")


# crear tabla resumen de los datos por grupo y metabolito
resultados <- lapply(unique(df_long$Metabolite), function(met) {
  datos_met <- subset(df_long, Metabolite == met)
  res <- describeBy(datos_met$Value, group = datos_met$Group, mat = TRUE)
  res$Metabolite <- met
  res$Group <- res$group1
  res$ID <- paste0(res$Metabolite, "-", res$Group)
  res
})
resultados_df <- bind_rows(resultados)
head(resultados_df)
```

Se puede observar que cada grupo esta conformado por 47 muestras para la cachexia, mientras que el grupo control está formado por 33.

```
# análisis evaluativo de los datos

# lista de metabolitos
metabs <- unique(df_long$Metabolite)

# generar histograma para cada metabolito
for (m in metabs) {
  print(
    ggplot(
      dplyr::filter(df_long, Metabolite == m),
      aes(x = Value, fill = Group)
    ) +
      geom_histogram(bins = 30, alpha = 0.6, position = "identity") +
      theme_minimal() +
      labs(title = paste("Histograma de", m)) +
      theme(legend.position = "bottom")
  )
}

# generar bloxplot por cada metabolito
```

```
for (m in unique(df_long$Metabolite)) {
  print(
    ggplot(filter(df_long, Metabolite == m),
           aes(x = Group, y = Value, fill = Group)) +
      geom_boxplot() +
      theme_minimal() +
      labs(title = paste("Boxplot de", m)) +
      theme(legend.position = "none")
  )
}


# PCA
# escalar y transponer matriz para generar el PCA
scaled_expr <- scale(t(matriz))
pca <- prcomp(scaled_expr)

# agrupar datos
pca_df <- as.data.frame(pca$x)
pca_df$Group <- colData(se)$Group

# generar PCA
ggplot(pca_df, aes(x = PC1, y = PC2, color = Group)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "PCA de metabolomica de control vs cachexia")

# HEATMAP
# calcular coeficiente de variación
cv <- apply(assay(se), 1, function(x) sd(x) / mean(x))
top_metabs <- names(sort(cv, decreasing = TRUE))[1:25]

# generar heatmap
pheatmap(assay(se)[top_metabs, ],
         annotation_col = as.data.frame(colData(se)),
         scale = "row",
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         main = "Top 25 metabolitos más variables",
         fontsize_row = 6,   # Tamaño de letra de las filas
         fontsize_col = 2     # Tamaño de letra de las columnas
)
```

En cuanto a los resultados obtenidos en el análisis evaluativo de los datos podemos observar que los metabolitos no muestran una distribución normal en ninguno de los dos grupos ni en el total, tal y como se puede ver en los histogramas. Además, se puede observar una mayor variabilidad y concentración en los metabolitos del grupo de cachexia. Sin embargo, estos cambios no se reflejan en el PCA, que no se ve un cambio destacado en el perfil de ambos grupos.

Finalmente, se llevó a cabo un heatmap para analizar la agrupación de las muestras. Se puede observar que un gran numero de muestras de cachexia si muestran una agrupación clara. Sin embargo, los datos necesitarían una transformación para poder llevar a cabo un análisis mas profundo de los posibles cambios entre ambos grupos.

Todos los documentos de la PEC se encuentran en el repositorio de github: https://github.com/jorgesr92/Soliz_Rueda_Jorge_PEC1