MÁSTER UNIVERSITARIO EN BIOINFORMÁTICA Y BIOESTADÍSTICA

SOFTWARE PARA EL ANÁLISIS DE DATOS

Prueba de Evaluación Continua 3 (PEC3)

CONTENIDOS Y ORIENTACIONES

La PEC3 consta de varios ejercicios organizados en secciones que corresponden a cada uno de los contenidos de los distintos laboratorios. Para realizar la PAC3, es necesario haber trabajado los siguientes laboratorios y recursos asociados:

- LAB5. Paquetes de R para la Bioinformática
- LAB6. Introducción al Machine Learning con R

Nota: Se pueden incluir conceptos o funciones trabajadas en los contenidos de los Laboratorios anteriores.

FORMATO DE ENTREGA

Es obligatorio que la solución de la PAC3 se desarrolle con RMarkdown.

Se entregará el documento generado por Rmarkdown en formato HTML o PDF.

FECHA LÍMITE DE ENTREGA

La fecha límite de entrega de la PAC3 es el **martes 23 de diciembre**.

Sección 1. Modelos y machine learning (7 puntos)

Ejercicio 1 (3 puntos)

A partir del conjunto de datos **Breast Cancer Wisconsin** (del paquete 'mlbench') sobre diagnóstico de cáncer de mama basado en características de células tumorales, se destacan las siguientes variables:

- Cl.thickness: grosor del grupo celular (1-10)
- Cell.size: uniformidad del tamaño celular (1-10)
- Cell.shape: uniformidad de la forma celular (1-10)
- Bare.nuclei: núcleos desnudos (1-10)
- Class: diagnóstico (benign=benigno, malignant=maligno)

Las variables en este dataset están codificadas en escala ordinal de 1 a 10, donde valores más altos generalmente indican características más anormales. Información adicional: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

Puedes cargar el dataset con:

```
# install.packages("mlbench")
library(mlbench)
data("BreastCancer")
```

Se pide:

a) (0.5 puntos) Carga y explora el conjunto de datos. Muestra un resumen estadístico de las variables numéricas principales, verifica la presencia de valores perdidos y describe cómo tratarás estos valores faltantes.

b) (0.5 puntos) Realiza un gráfico de dispersión entre `Cell.size` (uniformidad del tamaño celular) y `Cell.shape` (uniformidad de la forma celular), diferenciando por colores el tipo de diagnóstico (Class: maligno vs benigno). Interpreta visualmente si existe una separación clara entre ambos grupos.

c) (0.5 puntos) Ajusta un modelo de regresión lieanl con `Cell.size` como variable dependiente y `Cell.shape` como variable independiente. Interpreta los coeficientes del modelo y su significación estadística.

d) (0.5 puntos) Realiza un análisis de los residuos del modelo anterior. Identifica posibles valores atípicos o influyentes utilizando gráficos diagnósticos apropiados. ¿Qué implicaciones tienen estos hallazgos para el modelo?

e) (0.5 puntos) Ajusta un modelo de regresión logística con `Class` como variable dependiente y `Cell.size` y `Cell.shape` como variables independientes. Interpreta los coeficientes del modelo y su significación estadística.

AYUDA: Puedes usar la función `glm()` con modelo `family = binomial(link = "logit")`.

f) (0.5 puntos) Calcula la matriz de confusión del modelo y evalúa su desempeño mediante la sensibilidad (sensitivity) y especificidad (specificity). Interpreta los resultados en el contexto clínico: ¿qué tipo de error (falsos positivos vs falsos negativos) sería más preocupante en el diagnóstico de cáncer?

AYUDA: Una matriz de confusión es una tabla de contingencia que permite evaluar el rendimiento de modelos de clasificación comparando las predicciones del modelo con los valores reales. En sus filas se representan las clases predichas y en sus columnas las clases reales (o viceversa), mostrando así cuántas observaciones se clasificaron correctamente y cuántas se confundieron entre distintas categorías. Por ejemplo, con 100 muestras evaluados, la matriz de confusión podría verse así:

	Real: Negativo	**Real: Positivo**
Pred: Negativo	50	10
Pred: Positivo	5	35

En R, puedes crear una matriz de confusión fácilmente con la función `table()` proporcionando como argumentos los valores reales y las predicciones.

AYUDA: Sensibilidad y especificidad son dos métricas fundamentales derivadas de la matriz de confusión que evalúan diferentes aspectos del rendimiento de un modelo de clasificación.

$$\text{Sensibilidad} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

$$\text{Especificidad} = \frac{\text{Verdaderos Negativos}}{\text{Verdaderos Negativos} + \text{Falsos Positivos}}$$

Usando el ejemplo anterior:

Sensibilidad = 35/(35+10) = 0.78

Especificidad = 50/(50+5) = 0.91

Ejercicio 2 (4 puntos)

El proyecto GEUVADIS (Genetic European Variation in Health and Disease) es un estudio que integra datos de variación genética y expresión génica en poblaciones europeas. Para este ejercicio, trabajaremos con datos de expresión génica (RNA-seq) de 465 individuos de cinco poblaciones diferentes: Utah (residentes de Utah con ascendencia Europea), Finland, British, Tuscan (Italia) y Yoruba (Nigeria). Este dataset nos permitirá estudiar cómo la estructura poblacional se refleja en los patrones de expresión génica.

Puedes descargar y leer los datos usando el siguiente código:

```
# Descargar el fichero, solo si no existe
if (!file.exists("geuvadis_fpkmss.tsv")) {
  download.file(url="https://www.ebi.ac.uk/gxa/experiments-content/E-GEUV-1/resources/
  ExperimentDownloadSupplier.RnaSeqBaseline/fpkmss.tsv",
              destfile="geuvadis_fpkmss.tsv")
}

# Los campos están separados por tabulador, con el encabezado y identificador (Ensamble)
# de los genes en la primera columna
expr_data <- read.delim("geuvadis_fpkmss.tsv",
                        skip=4,
                        header = TRUE,
                        row.names = 1,
                        sep = "\t",
                        check.names = FALSE)


# Hay genes con nombres repetidos que debemos diferenciar
row.names(expr_data) <- make.unique(expr_data[,"Gene Name"])
expr_data <- expr_data[,-1] # eliminamos los Ensamble ID

expr_data <- as.matrix(expr_data)
```

Se pide:

a) (0.5 puntos) Exploración de datos:

- Mostrad las dimensiones de la matriz de expresión (número de genes y muestras)
- Cread un vector con la población de cada muestra y calculad el número de individuos por población.
- Eliminad valores nulos (por filas). ¿Cómo afecta a las dimensiones de la matriz?

b) (0.5 puntos) Preparad los datos para el análisis de estructura poblacional:

- Seleccionad los genes más variables (por ejemplo, los 1000 genes con mayor varianza entre muestras). Justificad por qué este filtrado es importante para el análisis de estructura poblacional
- Aplicad una transformación logarítmica (log2) a los datos de expresión, añadiendo una constante pequeña para evitar problemas con valores cero
- Calculad y mostrad un resumen estadístico de los datos transformados

c) (2 puntos) Realizad un Análisis de Componentes Principales (PCA) sobre los datos de expresión génica:

- Aplicad PCA a la matriz de expresión filtrada y transformada
- Mostrad la proporción de varianza explicada por los primeros 10 componentes principales mediante un gráfico
- Cread un gráfico de dispersión de PC1 vs PC2, coloreando los puntos según la población de origen y utilizando diferentes formas para distinguir mejor las poblaciones
- Repetid el gráfico para PC2 vs PC3
- Interpretad los resultados: ¿Qué poblaciones se diferencian más claramente? ¿La población africana (Yoruba) muestra una separación clara de las poblaciones europeas? ¿Por qué creéis que ocurre esto desde un punto de vista biológico?

d) (1 punto) Realizad un análisis de agrupamiento sobre los patrones de expresión génica:

- Calculad una matriz de distancias entre muestras usando la distancia euclidiana sobre los primeros 10 componentes principales (esto reduce el ruido y mejora la interpretabilidad)
- Realizad un agrupamiento jerárquico con el método "ward.D2"
- Representad el dendrograma. Utilizad colores para identificar las diferentes poblaciones
- Calculad la altura óptima de corte del dendrograma para obtener 5 grupos y comparad estos grupos con las poblaciones reales. ¿Coinciden los clusters identificados con las poblaciones biológicas?

AYUDA: os puede ser útil utilizar la librería `dendextend` para añadir colores en el dendograma

Sección 2: Aplicaciones con R y Rstudio (3 puntos)

Ejercicio 3 (3 puntos)

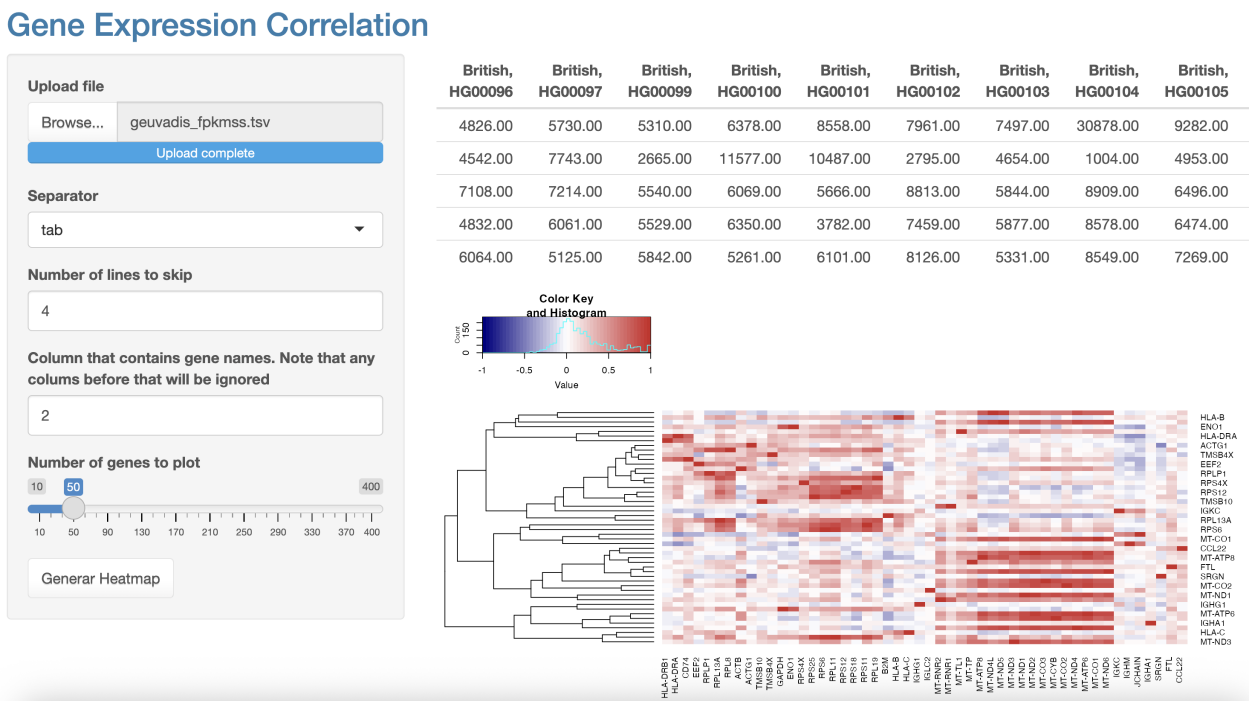
El análisis de correlación entre niveles de expresión génica permite identificar genes que funcionan de manera coordinada, lo cual es fundamental para comprender redes regulatorias y vías metabólicas. Desarrollad una aplicación web para la visualización de esta correlación. La aplicación debe ser capaz de visualizar la correlación en el dataset GEUVADIS del ejercicio anterior a partir del archivo original descargado.

El objetivo es crear una aplicación en R con Shiny que permita al usuario:

1. Cargar un archivo de datos de expresión génica
2. Ajustar opciones del formato del archivo como: (1) que separador usar, (2) el número de líneas que hay que saltar, (3) en que columna hay el nombre de los genes y empiezan los datos (e ignorar las columnas previas).
3. Filtrar los datos, por ejemplo eliminando NAs, y ordenar los genes según su varianza (como hemos hecho en el ejercicio anterior).
4. Generar un heatmap de la correlación entre los niveles de expresión génica limitando el número de genes a representar.

La aplicación debe permitir al usuario cargar un archivo en formato CSV, TSV o TXT, con los datos de expresión génica, y mostrar una tabla con una vista previa de los primeros datos cargados (primeras filas) para que el usuario pueda verificar que el archivo es correcto.

Un ejemplo del interfaz de la aplicación es el siguiente:



Es imprescindible añadir una captura de pantalla donde se vea la app, incluyendo el heatmap.

AYUDA: Los paquetes requeridos son 'shiny' y 'shinythemes' (opcional, para mejorar la estética de la aplicación). Algunas opciones populares pare representar un heatmap son las implementaciones en `gpltos` o `ComplexHeatmap`

AYUDA: después de cargar la libraría `shiny` añade la opción `options(shiny.maxRequestSize=128*1024^2)` para evitar tener problemas con el tamaño máximo del fichero.