RESPUESTAS ACTIVIDAD 1.

1. **Ambiente y librerías de trabajo (0,5 puntos):**

He utilizado Rstudio (versión 2024.09.1 Build 394) para trabajar con R (versión 4.4.2).

Todas las librerías de trabajo que he utilizado son:

* ggplot2 (para realizar las gráficas)
* readr (para la lectura de datos)
* stats (para el algoritmo de “Análisis de componentes principales” (PCA) y “Multidimensional scaling" (MDS), este último no incluido en el script principal)
* RDRToolbox (para el algoritmo “Isometric Mapping” (IMAP) y “Lineal L Embedding” (LLE), este último no incluido en el script principal).
* Rtsne (para el algoritmo “T-distributed stochastic Neighbor Embedding” (t-SNE))
* Rdimtools (para el algoritmo “Laplacian Eigenmaps” (LE) y “Maximum Variance Unfolding” (MVU), este último no incluido en el script principal)
* lle (para el algoritmo “Lineal L Embedding” (LLE), no incluido en el script principal)
* uwot (para el algoritmo “Uniform Manifold Approximation and Projection” (UMAP).

**¿Qué librerías han sido necesarias para llevar a cabo la implementación de cada uno de los métodos de aprendizaje no supervisado? Explica el rol de los argumentos de las funciones que has utilizado en cada método de reducción de dimensionalidad.**

\*Librerías: Contestado en el apartado anterior.

Paso común a todas las técnicas:

sapply(…[], as.numeric): sapply aplica una función a cada columna de la matriz o data frame (argumento: …); el argumento [] representa las columnas seleccionadas; el argumento as.numeric convierte los valores de cada columna a tipo numérico.

A continuación, detallo la información de los 4 algoritmos utilizados en el **archivo “script principal”.**

**PCA:** Para este algoritmo utilicé la librería stats, para poder ejecutar el algoritmo, y la librería ggplot2, para graficar los resultados (esta última la usé en todos los algoritmos).

Algoritmo:

prcomp(data, center=TRUE, scale=FALSE):

En la función prcomp, para el cálculo de componentes principales, se utilizan como argumentos:

data: los datos (muestra) sobre los cuales hará el análisis de componentes principales

enter =TRUE, para centrar las variables antes de realizar el PCA

scale= FALSE, para que no escale los datos ya que están todos en las mismas unidades.

Además, otras funciones utilizadas para la representación de este algoritmo:

data.frame(pca.results$x): la función data.frame toma como argumento la tabla de componentes principales generada por prcomp (pca.results) y se le dice mediante $x que busque los valores de componentes principales(x) en esta tabla)

Tras el cálculo de varianzas, min(which(varianza.acumulada > 0.9)) min es una función que coge el valor más pequeño dentro de los que cumplen la condición. La condición en este caso viene determinada por la función which, que selecciona como TRUE los valores que están por encima del 90% de la varianza acumulada, es decir, que expliquen más de un 90% de la varianza.

paste0(paste('PC1', round(varianza.explicada[1] \* 100, 2)), '%'): La función paste0 concatena el resultado anterior (…) con el símbolo de porcentaje (%); paste('PC1', round(…, 2)); la función paste concatena el texto ('PC1') con el valor redondeado (round), separándolos por un espacio; round(… ,2): esta función redondea el primer argumento (…) a los decimales concretados en el segundo argumento (2).

Elegí este algoritmo como primera opción asumiendo una relación lineal entre los datos de la muestra. Mediante la representación en 2 dimensiones pude encontrar una relación lineal de diferencias entre el grupo AGH (en rojo) y el resto de la muestra. También observé una separación entre los grupos CHC (en naranja) y HPB en morado) observada en la dimensión 2, pero si no conociera previamente las etiquetas de los grupos de datos esta deducción no podría hacerla, solo que hay 2 conjuntos de datos, izquierda (correspondiente a AGH) y derecha (resto de la muestra), observada en la dimensión 1.

**IMAP:** Elegí este algoritmo debido a que aunque altera mucho las distancias permite separar grupos de datos que no tienen relaciones lineales entre sí. Utilicé una K= 15, una K más baja apenas separaba los conjuntos de datos por sus varianzas y una K mayor dispersaba mucho los datos. Con K= 15, conseguí separar los 5 grupos de la leyenda, que también se intuyen sin las etiquetas, aunque los datos están dispersos. Se hacía necesario confirmar con más algoritmos.

Algoritmo:

Isomap(data=data2, dims=1:4, k=15, plotResiduals=TRUE):

Isomap: función para la reducción de dimensionalidad.

Argumentos de la función:

data : datos (matriz) sobre los que se realiza la reduccion de dimensionalidad.

Dims: dimensiones de las columnas del espacio reducido

K: número de vecinos cercanos a cada punto.

potResiduals: devuelve la varianza explicada por las diferentes dimensiones

Además, otra función, utilizada para la representación de este algoritmo:

isomap.df <- data.frame(isomap.results$dim2): dataframe con los puntos que queremos dibujar en el plano 2D de los datos transformados por el Isomap. En este caso los de la columna correspondiente a la dimensión 2.

**t-SNE:** Este algoritmo realizó una agrupación de los distintos grupos de datos más compacta y gracias a esto pude confirmar la presencia de 5 agrupaciones en los datos, que corresponden a las 5 etiquetas de tipos de cáncer. Además en el grupo CFB (azul) parece haber un subconjunto de datos que se alejan del grupo mayor y se mantienen muy próximos. Esto podría indicar un subtipo de cáncer dentro de este grupo principal (CFB).

Algoritmo:

tsne <- Rtsne(X=data3)

Rtsne: función para la reducción de dimensiones

Argumentos de la función:

X: datos sobre los que reduciremos la dimensionalidad

Además, otra función, utilizada para la representación de este algoritmo:

tsne\_result <- data.frame(tsne$Y)

$Y: seleccionamos la columna de la variable Y, esta contiene la matriz del t-SNE.

**UMAP:** Por último, utilice UMAP, ya que aunque distorsiona el espacio, las proyecciones de los puntos en el espacio de menor dimensión son localmente isométricas. Los resultados para una distancia mínima entre puntos igual a 0.1 separó bien los grupos, y permitió seguir observando un subgrupo en CFB (azul), a pesar de que los subconjuntos de datos están muy comprimidos. Curiosamente algunos subconjuntos de muestra del grupo AGH (en rojo) se mezclaban con CFB y algunos datos de la muestra perteneciente a la etiqueta HPB (morado) con CGC (verde), lo que me hace deducir que podría tratase de outliers.

Algoritmo:

umap(data5, n\_neighbors=0.2 \* nrow(data5),

n\_components = 2, min\_dist = 0.1, local\_connectivity=1, ret\_model = TRUE, verbose=TRUE)

umap(): Función para la reducción de la dimensionalidad.

Argumentos de la función:

* x: reducción de la dimensionalidad
* n\_neighbours: número entero que indica el número de vecinos cercanos, se trata de una proporción.
* nrow(data5): número total de filas en data5.
* n\_componentes: número entero que determina el tamaño del espacio de salida
* min\_dist: distancia minima permitida entre puntos
* ret\_model = TRUE: Indica que se debe devolver el modelo UMAP completo, no solo la proyección.
* verbose: tiempo hasta que se complete el calculo

Además, otra función, utilizada para la representación de este algoritmo:

data.frame(umap.results$embedding)

$embedding: Esta expresión accede a la matriz de embebido (embedding) dentro del objeto (umap.results). embedding: matriz que contiene las coordenadas de los datos proyectados en el espacio de menor dimensión.

1. **Procesamiento de los datos (1,5 puntos):**

**¿Qué problemas has detectado en el conjunto de datos que te han sido proporcionados? ¿Cómo los has solucionado?**

El principal problema en el conjunto de datos es que las variables de la base de datos (gene\_expression) venía sin etiquetar. Para etiquetar las variables tuve que utilizar la función readr para leer el archivo CSV con los datos de expresión génica, y asignar los nombres a las columnas basándome en los nombres de las etiquetas leídos en el archivo de texto separado (column\_names). Además, eliminé tres variables (genes: "MIER3" "ZCCHC12" "RPL22L1") que no tenían información, dado que la suma de sus columnas era igual a 0.

Dado que el archivo original “classes”, que contenía los tipos de cáncer asociados a cada muestra, presentaba 801 filas y el archivo de expresión de los genes (gene\_expression) contenía 800, eliminé la fila 1 (“sample 0”) al crear el objeto “clases” para evitar el error producido a la hora de representar el gráfico con la función ggplot, por la diferencia en el número de filas de las bases de datos originales (classes y gene\_expression).

1. **Métodos no supervisados (2,5 puntos):**

**¿Cuál es el motivo por el cual has seleccionado estas técnicas de reducción de dimensionalidad? (1 punto).**

**PCA:** utilicé este algoritmo de reducción de la dimensionalidad para una primera aproximación debido a que busca la combinación lineal de las variables originales que maximice la varianza de los datos y minimice la correlación entre ellas.

Dado que la mayor parte de las variables no seguían una relación lineal pasé al siguiente algoritmo:

**IMAP:** Lo utilicé debido a la relación no lineal de las variables y este algortimo recoge distancias no lineales, y a que busca conservar la distancia original entre los puntos del espacio de mayor dimensión al espacio de menor dimensión cuando es proyectado. De esta forma pensé que podía obtener unos resultados de agrupación de subtipos de datos que no alterara las relaciones entre los objetos.

**t-SNE**: utilicé este algoritmo dado que reduce la dimensionalidad mendiante técnicas no lineales y es muy bueno para representación de datos. En este caso me interesaba porque lo que busca es la probabilidad de que dos puntos sean cercanos en un espacio de menor dimensión y quería comprobar si generaba los subgrupos (tipos de cáncer) observados en IMAP con una menor dispersión de los datos dentro de los subgrupos.

**UMAP:** Lo utilicé porque este algoritmo garantiza que las proyecciones de los puntos en el espacio de menor dimensión sean localmente isométricas y, por tanto, esto indica que las distancias entre puntos cercanos en el espacio original se preservan en la proyección. Por tanto, aunque puede distorsionar las distancias entre puntos alejados, como mantiene muy bien la relación entre puntos cercanos, y lo que yo quería era terminar de confirmar la existencia de los 5 subgrupos de datos, este algoritmo me parecio muy apropiado.

Probé también otros algoritmos para reducción de la dimensionalidad:

**MDS:** Lo probé dado que no se necesitan hacer suposiciones previas y mantiene la estructura de similitud entre los objetos, pero los resultados fueron similares a los de PCA con una K=3.

**LLE:** Intenté utilizarlo, pero mi versión de R (4.4.2) no me permitía instalar el paquete.

LE: Lo utilicé por curiosidad, pero no me resultó útil para los conjuntos de datos de la actividad. Proporciones de vecinos pequeñas generaban mucha distorsión del espacio y no separaba bien subgrupos, y con Ks altas los datos aparecían muy dispersos.

**MVU:** Probé a ejecutarlo a pesar de saber que computacionalmente es muy exigente. Tenía interés en conocer el resultado de este algoritmo ya que separa los conjuntos de datos sin eliminar la estructura dentro de los grupos de vecinos. No me fue posible ejecutarlo sin que la memoria se saturase o me saliera este error disminuyendo mucho el número de filas: Error: no se puede ubicar un vector de tamaño 4.9 Mb.

**¿Qué aspectos positivos y negativos tienen cada una de las técnicas escogidas? (0,5 puntos).**

PCA:

**Ventajas:** El PCA elimina correlaciones entre las variables, mejorando la independencia de los datos y es muy efectivo en conjunto de datos que poseen variables con altas

correlaciones.

**Inconvenientes:** Está limitado a conjuntos de datos en donde las variables estén linealmente correlacionadas. Se necesitan procesos de estandarización previos para una correcta generación de los componentes principales y la selección de componentes debe ser cuidadosa seleccionando el mínimo de estas que expliquen la mayor variabilidad posible.

IMAP:

**Ventajas:** Se trata de un método muy eficaz para conjuntos de datos en cuyas

variables se mantienen relaciones no lineales.

**Inconvenientes:** si no existe suficiente densidad de puntos en el espacio de alta

Dimensión, el paso de determinación de distancias entre los puntos puede no funcionar con precisión pudiendo ser estas muy diferentes a las distancias originales. Encontrar el parámetro k correcto no es fácil.

t-SNE:

**Ventajas:** supera a otras técnicas en el manejo de datos no lineales ya que captura

muy bien las relaciones complejas entre las variables preservando muy bien la localidad de los datos. Por tanto, es muy útil para identificar y estructurar datos.

**Inconvenientes:** Tiene comportamiento estocástico debido a la influencia de elementos aleatorios, por lo que no es determinista. Tiene complejidad cuadrática en tiempo y espacio por lo que con muestras muy grandes puede ser muy lento. Además, está limitado a 3 dimensiones, por lo general.

UMAP:

**Ventajas:** supera ciertas limitaciones de otras técnicas de reducción de la dimensionalidad en estructuras de datos donde no existe linealidad. Además, las restricciones del MVU garantizan que las proyecciones de los puntos en el

espacio de menor dimensión sean localmente isométricas.

**Inconvenientes:** Tiene un gran coste computacional. Por tanto, se ve limitado por el número de muestras (número de pacientes en la actividad 1).

**Comenta para cada caso el resultado de representar gráficamente las primeras dos variables de cada uno de los métodos. ¿Qué se observa? ¿Tiene sentido biológico? Razona tu respuesta (1 punto).**

**PCA:** Aparecen los datos (pacientes) dispersos formando una nube de puntos sin relaciones claras entre ellos. No tiene sentido biológico, lo cual es lógico, dado que el cáncer es una enfermedad muy compleja que no se puede explicar solo por las relaciones entre 2 tipos de genes, son muchas más las variables y las relaciones entre ellas las que influyen en que cada tipo de cáncer sea el que es y no otro.

**IMAP:** Sin alterar los parámetros más que el número de variables se observa la misma nube de puntos que en el caso anterior. Reduciendo k a 2 y dimensiones a 2, los datos aparecen formando filas con ramificaciones también en forma de filas. Mismo razonamiento que en el caso anterior. Carece de sentido biológico.

Lo mismo ocurre para **t-SNE** y **UMAP.**