1. Tabla de contenido

2. **Abstract**

El estudio se centra en el análisis de metabolitos en muestras biológicas de pacientes con caquexia, caracterizada por una pérdida severa de masa muscular. Después de transformar el archivo human_cachexia.csv en un objeto SummarizedExperiment, analicé metabolitos clave como la creatinina, pi-Methylhistidine, oxoglutarate, succinato, acetato, y aminoácidos como alanina, glutamina, leucina y valina. Los resultados indican que metabolitos como la creatinina y pi-Methylhistidine reflejan un elevado catabolismo muscular, mientras que oxoglutarate, succinate y acetate se asocian a disfunciones en la producción de energía. Las correlaciones entre estos metabolitos sugieren interacciones clave dentro del ciclo de Krebs. Además, los aminoácidos juegan un papel importante en la preservación muscular y la respuesta inmune. Este análisis permitió comprender los cambios metabólicos asociados a la caquexia, lo que podría contribuir al desarrollo de terapias específicas para mejorar el tratamiento de la condición y la calidad de vida de los pacientes.

3. **Objetivos**

Los principales objetivos de este trabajo son:

1. **Explorar** los metabolitos involucrados en la caquexia para comprender los cambios metabólicos asociados con la pérdida de masa muscular.
2. **Identificar** metabolitos clave que reflejan el catabolismo muscular, la disfunción energética y la alteración de las rutas de síntesis proteica.

3. **Evaluar** las correlaciones entre los metabolitos para identificar interacciones metabólicas significativas.

4. **Proponer** un modelo metabólico basado en los resultados obtenidos para mejorar la comprensión de los procesos fisiopatológicos de la caquexia.

4. Métodos

1. Preparación de los Datos

El conjunto de datos lo descargué del repositorio https://github.com/nutrimetabolomics/metaboData en GitHub y contiene mediciones de metabolitos en **77 pacientes**, **47 con caquexia** y **30 controles**. Además de las concentraciones de metabolitos, los metadatos incluyen el identificador único de cada paciente (*Patient ID*) y la información sobre la pérdida muscular. El conjunto de datos fue organizado utilizando la clase *SummarizedExperiment* de Bioconductor.

(https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/human_cachexia.csv)

2. Creación del Objeto SummarizedExperiment

El siguiente paso que hice fue crear un objeto de la clase *SummarizedExperiment*, que organiza tanto los datos experimentales (concentraciones de metabolitos) como los metadatos (información clínica de los pacientes) en un formato estructurado para análisis (ver estructura en el apartado Resultados). Los datos provienen de un estudio de metabolómica en pacientes con caquexia, que incluye mediciones de varios metabolitos y metadatos clínicos como el identificador del paciente y el estado de pérdida muscular. El archivo **SummarizedExperiment_Diana_Gutiérrez.rda**, adjunto en el proyecto mio: **Gutierrez_Martinez_Diana_PEC1** (url en apartado Referencias), almacena estos datos de forma eficiente y facilita su análisis con herramientas de Bioconductor como **DESeq2** y **edgeR**. Lo que hice es procesar los datos eliminando valores faltantes (NA) y asegurando la alineación correcta entre los metadatos y los datos experimentales. El objeto *SummarizedExperiment* se creó utilizando la función SummarizedExperiment()(mirar código R), organizando las mediciones de metabolitos en el slot assays y los metadatos en colData.

3. Análisis Exploratorio de los Datos

El análisis comenzó preparando y organizando de los datos experimentales y metadatos usando la clase *SummarizedExperiment* de Bioconductor, esto me ayudo a manejar de manera más cómoda los datos ómicos. El conjunto de datos proviene del archivo llamado *SummarizedExperiment_Diana_Gutiérrez.rda*, y lo he llamado "**se"**. Este archivo rda contiene mediciones de metabolitos en 77 pacientes, 47 de ellos con cachexia y 30 controles.

El proceso comenzó con la instalación y carga de las librerías necesarias, como *usethis*, *BiocManager*, *SummarizedExperiment* y *readr*. Estas librerías permitieron cargar el archivo CSV denominado *human_cachexia*, que contiene datos clínicos y experimentales de los pacientes. Los datos se separaron en dos partes: los metadatos (que incluyen el ID del paciente y la información sobre la pérdida muscular) y los datos experimentales (mediciones de las concentraciones de metabolitos). Para asegurar la calidad de los datos, se eliminaron las filas con valores faltantes (NA) en los metadatos y se verificó que las filas en los datos experimentales coincidieran con los metadatos de los pacientes. También se eliminaron posibles duplicados. La alineación de los datos se validó utilizando las identificaciones de los pacientes, para garantizar que los metadatos y las mediciones de los metabolitos correspondieran correctamente.

Una vez preparados los datos, se creó un objeto de la clase *SummarizedExperiment* utilizando la función *SummarizedExperiment()*. Los datos experimentales, que corresponden a las

concentraciones de metabolitos, se almacenaron en el *slot assays*, bajo el nombre "counts". Los metadatos del paciente, que incluyen el *Patient ID* y el estado de la pérdida muscular, se almacenaron en el *slot colData*. Este objeto *SummarizedExperiment* se guardó en un archivo con extensión .rda para su posterior análisis y se adjuntó al proyecto (llamado *Gutierrez_Martinez_Diana_PEC1*).

3.1 Visualización de los Metadatos

Una vez estructurado el objeto *SummarizedExperiment*, se realizó un análisis exploratorio inicial mediante una visualización de la correlación entre los metabolitos, lo que permitió obtener una visión general de las interrelaciones entre ellos. Esta visualización es útil para explorar posibles vínculos entre las concentraciones de metabolitos y la pérdida muscular, dentro de un marco de datos organizado y estructurado, lo que facilita un análisis más profundo sobre el papel de los metabolitos en la caquexia. Este enfoque con la clase *SummarizedExperiment* ofrece una plataforma eficiente para trabajar con herramientas avanzadas de análisis de datos ómicos, como DESeq2 o edgeR, y proporciona una base sólida para explorar correlaciones metabólicas y realizar estudios más profundos sobre los mecanismos de la caquexia. A continuación, se visualizó la estructura de los metadatos, que incluye los identificadores de pacientes y el estado de pérdida muscular (Muscle loss).

3.2 Estadísticas Descriptivas

Se generaron estadísticas descriptivas para obtener una visión general de la variabilidad en las concentraciones de metabolitos. Se realizó un resumen estadístico de las concentraciones de metabolitos, observando que algunos metabolitos, como el **citrato** y la **creatinina**, mostraron valores mucho más altos en comparación con otros.

Código utilizado para calcular la media y desviación estándar:

```
mean_values <- apply(assays(se)$counts, 2, mean)
sorted_mean_values_desc <- sort(mean_values, decreasing = TRUE)
head(sorted_mean_values_desc, 10)
sd_per_metabolite <- apply(assays(se)$counts, 2, sd)
```

3.3 Análisis de Correlación entre Metabolitos

Empece calculando la matriz de correlación utilizando la función **cor()**, así me calculava las correlaciones entre todas las columnas del conjunto de datos. Llamé a la matriz de datos como **data_matrix**, esta contiene las mediciones de los metabolitos de los Patient ID. Usé el argumento **use = "pairwise.complete.obs"**, así me aseguro que solo se incluyan los pares de observaciones no faltantes. El resultado que me da es una matriz de correlación (**cor_matrix**), dandome la relación lineal entre los metabolitos. Guardé la matriz de correlación en un archivo CSV utilizando la función **write.csv()**, así tenia un registro de la matriz para que el professor pudiera hacer una visualización al entrar en mi directorio. Luego, la matriz de correlación se transforma en formato largo con la **función melt()** de la librería reshape2, ya que ggplot2 requiere los datos en este formato para generar un gráfico. Transformo los datos para convertir la matriz de correlación de una estructura de tabla cuadrada en una lista de pares de metabolitos con sus valores de correlación correspondientes. Una vez estan los datos transformados, procedo a crear un gráfico de mapa de calor con la función **ggplot()** de la librería ggplot2. En el gráfico, los metabolitos son representados en los ejes X e Y, mientras que el color de las celdas del mapa de calor varía según el valor de la correlación entre los metabolitos. La función **scale_fill_gradient2()** se usa para definir una escala de colores en el mapa de calor, los valores negativos se muestran en azul, los valores cercanos a cero en blanco, y los valores positivos en rojo. Finalmente, el gráfico de mapa de calor se guarda como una imagen PNG utilizando la función **ggsave()**, lo que permite almacenar el gráfico. El análisis realizado en este código permite observar visualmente las interrelaciones entre los metabolitos, destacando aquellos que están fuertemente correlacionados, como el succinato y el oxoglutarato.

3.4 Análisis de Componentes Principales (PCA)

El PCA se utilizó para reducir la dimensionalidad de los datos y observar la variabilidad en las muestras. Se estandarizaron los datos y se proyectaron sobre los dos primeros componentes principales (PC1 y PC2). Para realizar el análisis de Componentes Principales (PCA), primero se aplicó la función **prcomp** sobre la matriz de conteos transpuesta de los metabolitos, estandarizando las variables con **scale = TRUE**. Los scores de los componentes principales se extrajeron con **pca_result$x** y se guardaron en un dataframe, que luego se exportó a un archivo CSV. Se generó un gráfico de dispersión de los dos primeros componentes principales (PC1 vs PC2) para visualizar patrones y se guardó en formato PNG. Luego, se calculó la proporción de varianza explicada por cada componente principal y la varianza acumulada. Estos resultados se guardaron en un dataframe y también en un archivo CSV. Finalmente, se creó un gráfico de barras para mostrar la varianza explicada por cada componente y la varianza acumulada, guardando el gráfico en PNG para el informe.

5. Resultados

Los resultados que nos muestra el SummarizedExperiment son:

```
class: SummarizedExperiment
#dim: 63 63
#metadata(0):
#assays(1): counts
#rownames(63): PIF_178 PIF_087 ... NETCR_008_V1 NETCR_008_V2
#rowData names(1): MuscleLoss
#colnames(63): 1,6-Anhydro-beta-D-glucose 1-Methylnicotinamide ...
#  pi-Methylhistidine tau-Methylhistidine
#colData names(2): Patient ID Muscle loss
```

La clase SummarizedExperiment es una estructura en R diseñada para almacenar datos experimentales y metadatos. El objeto creado tiene una dimensión de 63 filas (pacientes) y 63 columnas (metabolitos), lo que indica que se midieron 63 metabolitos en 63 muestras de pacientes.
Metadatos: No contiene metadatos adicionales (metadata(0)).
Asymetría de datos (assays): Hay un conjunto de datos llamado counts, que contiene las concentraciones de metabolitos.
Filas (rownames): Representan identificadores de pacientes como PIF_178.
rowData (MuscleLoss): Indica la pérdida de masa muscular en cada paciente, una variable clave en el análisis.
Columnas (colnames): Corresponden a los metabolitos medidos (por ejemplo, 1,6-Anhydro-beta-D-glucose).
colData (Patient ID, Muscle loss): Almacena el ID del paciente y el grado de pérdida muscular, esencial para estudiar su relación con los metabolitos.

Estadísticas descriptivas: análisis inicial para obtener una visión general del dataset.

Podemos hacer un analisis inicial para poder ver una visión general de los datos entrando directamente al archivo rda. Y con el codigo: colData(se) # Vemos que efectivamente estan las dos variables Patient ID y Muscle loss y nos sale los metabolitos relacionados a la caquexia y a este proyecto. Han separado 2 grupos: los controles y pacientes con caquexia.

```
DataFrame with 63 rows and 2 columns
                Patient ID Muscle loss
                <character>    <factor>
```

```
1,6-Anhydro-beta-D-glucose      PIF_178    cachexic
1-Methylnicotinamide          PIF_087    cachexic
2-Aminobutyrate               PIF_090    cachexic
2-Hydroxyisobutyrate        NETL_005_V1    cachexic
2-Oxoglutarate                PIF_115    cachexic
...                         ...        ...
cis-Aconitate             NETCR_005_V1     control
myo-Inositol                 PIF_111     control
trans-Aconitate              PIF_171     control
pi-Methylhistidine        NETCR_008_V1     control
tau-Methylhistidine       NETCR_008_V2     control
```

Podemos ver el resumen de metadatos en columna colnames(assays(se)$counts) [1] “1,6-Anhydro-beta-D-glucose” “1-Methylnicotinamide” “2-Aminobutyrate”
[4] “2-Hydroxyisobutyrate” “2-Oxoglutarate” “3-Aminoisobutyrate”
[7] “3-Hydroxybutyrate” “3-Hydroxyisovalerate” “3-Indoxylsulfate”
[10] “4-Hydroxyphenylacetate” “Acetate” “Acetone”

Miramos las condiciones de cada paciente relacionadas a la massa muscular :

Genero una tabla de frecuencias de la variable Muscle.loss y luego creo un gráfico de barras para visualizar cómo se distribuyen los valores de esa variable. Efectivamente **no hay distribución homogenea**.

Utilización del código summary:

Al hacer un **summary(assays(se)$counts)** Si nos enfocamos en los resultados de los metabolitos obtengo en general una gran variabilidad en sus concentraciones. Si nos enfocamos en el citrato y la creatinina, vemos que estan mucho más elevados en comparación con otros. Estos valores son importantes en el análisis de condiciones fisiológicas y pueden ser claves en estudios relacionados con el metabolismo y patologías como la caquexia. En la tabla de los resúmenes estadísticos, las estadísticas clave incluyen el valor mínimo, primer cuartil (Q1), mediana (Q2), media, tercer cuartil (Q3) y valor máximo para cada metabolito. En el caso de **1,6-Anhydro-beta-D-glucose** , los valores muestran una gran variabilidad, con un rango muy amplio entre el mínimo y el máximo, el minimo es 4.71 y el máximo es 685.40. Esto ocurre tambien con **1-Methylnicotinamide** con un mínimo de 6.42 y un maximo de 1032.77, mostrando una variablidad con un máximo mucho más alta. Si nos fijamos en algunos metabolitos tienen valores máximos extremadamente altos en comparación con la media, esto puede indicar la aparición de **outliers** o un pequeño número de muestras con concentraciones mucho más altas de esos metabolitos. Se puede utilizar para identificar patrones anómalos en los datos.

Calculo la media de los metabolitos y varianzas

Los metabolitos involucrados en la cachexia reflejan alteraciones en el metabolismo energético, la inflamación y el catabolismo de proteínas. La glutamina, con una media de 337.10 y una mediana de 284.29, está estrechamente relacionada con el catabolismo muscular, un proceso clave en la cachexia. En esta condición, el cuerpo comienza a descomponer las proteínas musculares para obtener aminoácidos como la glutamina, que juega un papel crucial en la síntesis de proteínas y el metabolismo muscular. La alanina, con una media de 303.18 y una mediana de 237.46, también presenta niveles elevados en cachexia, reflejando la degradación muscular asociada al proceso catabólico. La creatinina, por su parte, es el metabolito con la mayor concentración en las muestras analizadas, con una media de 9521.7754. Este compuesto es un subproducto del metabolismo muscular y sus niveles elevados se asocian tanto con la función renal como con el estado muscular. Dado que la cachexia está vinculada a una degradación significativa de la masa muscular, la creatinina es un biomarcador clave para evaluar el daño muscular y la pérdida proteica en los

pacientes. Otros metabolitos relevantes incluyen el hippurato, con una media de 2614.1479, y el citrato, que tiene una media de 2423.47 y una mediana de 2079.74, siendo este último un componente importante del ciclo de Krebs y del metabolismo energético en general. El pi-Methylhistidine, con una media de 391.5184, es un marcador de degradación muscular y tiene especial relevancia en pacientes con cachexia, ya que esta condición se caracteriza por una acelerada pérdida de masa muscular. Estos metabolitos proporcionan información crucial sobre los cambios metabólicos que ocurren en la cachexia, ayudando a comprender mejor los mecanismos de daño muscular y la alteración del metabolismo energético en estos pacientes.

https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/varianza_explicada.csv

Calculo la corrección entre los metabolitos

El análisis de correlación entre metabolitos revela importantes relaciones metabólicas en la cachexia. El succinato y el 2-oxoglutarato muestran una correlación moderada (0.5909), lo que refleja su estrecha relación en el ciclo de Krebs. Ambos metabolitos están conectados en la producción de energía, donde el 2-oxoglutarato se convierte en succinato.

Por otro lado, la acetona tiene una correlación débil con 2-oxoglutarato (0.0375), mientras que acetato y acetona están moderadamente correlacionados (0.4516), sugiriendo una relación en procesos redox. La alanina se correlaciona con acetato (0.74) y acetona (0.67), sugiriendo su rol en la degradación muscular y producción de energía. Serina y valina están correlacionadas (0.61), mientras que leucina y serina muestran una correlación más baja (0.29), reflejando su distinto rol en el metabolismo de proteínas. El ácido acético y el ácido adipato tienen una correlación moderada (0.32), asociada al uso de grasas. El ácido pantoténico, acetato y nicotinamida presentan correlaciones moderadas (0.26-0.40), vinculadas a vías energéticas. Finalmente, el ácido fenilacético y el indoxilsulfato tienen una correlación moderada (0.5959), destacando su papel en la detoxificación y el metabolismo de fenoles. La glucosa, con su baja concentración, refleja cambios en el metabolismo energético de los pacientes con cachexia. ¡Matriz correlacion

Hago mapa de calor de la correlación entre metabolitos

Instalo el paquete igraph y cargo el paquete. Estableces un umbral (threshold) de 0.7. Este valor se usará para filtrar las correlaciones. Hago una copia de la matriz de correlación original (cor_matrix) y la asigno a una nueva variable llamada cor_matrix_filtered. Modifico **cor_matrix_filtered** estableciendo a cero todas las correlaciones cuya magnitud sea menor que 0.7. Por tanto las correlaciones débiles se eliminan, dejando solo aquellas con correlaciones fuertes (mayores o iguales a 0.7 o menores o iguales a -0.7). Creo un mapa de correlación y guardo ese gráfico en un archivo PNG en mi escritorio.

```
ggplot(cor_data, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlación") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   hjust = 1)) +
  labs(title = "Mapa de calor de la correlación entre metabolitos",
       x = "Metabolito",
       y = "Metabolito")
```

En este mapa de calor los colores reflejan la fuerza de las correlaciones, donde los valores cercanos a 1 (tonalidades rojas) indican una fuerte correlación positiva, y los valores cercanos a -1 indican una

correlación negativa (tonalidad azul) (**Anexo4**). Ver archivo png en Github. Este tipo de visualización es útil para identificar las relaciones más fuertes entre los metabolitos, lo cual es crucial al estudiar la cachexia y su impacto en el metabolismo energético y muscular.

![Mapa de calor]

(https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/heatmap.png)

Gráfico de correlación

En el gráfico de correlación (**Anexo 1**), el citrato y la glutamina están cerca el uno del otro, conectados, lo que indica que están relacionados. El citrato es un compuesto clave en el ciclo de Krebs, que es el proceso que las células usan para obtener energía; en situaciones como la caquexia, que es una pérdida de peso extrema y debilidad muscular, el metabolismo energético del cuerpo se ve alterado. Esto puede hacer que los niveles de citrato cambien, ya que el cuerpo intenta producir más energía de lo normal para compensar el desequilibrio. La glutamina es un aminoácido muy importante para mantener el funcionamiento energético de las células y para la síntesis de proteínas, como las que forman los músculos. En la caquexia, el cuerpo descompone rápidamente las proteínas, lo que provoca una disminución de la glutamina; además, la glutamina ayuda al sistema inmunológico, que también puede verse afectado durante la enfermedad. Cuando en el gráfico el citrato y la glutamina están muy cerca y correlacionados positivamente, es decir, tienden a aumentar o disminuir juntos, esto podría sugerir que ambos están siendo influenciados por el mismo proceso metabólico en la caquexia. Por lo tanto, cuando el cuerpo necesita más energía, lo que puede suceder cuando hay un estado catabólico o de descomposición de tejidos, el citrato y la glutamina están trabajando juntos para tratar de mantener el equilibrio energético y protegen el cuerpo de los efectos del catabolismo acelerado. Por lo tanto, la correlación positiva entre citrato y glutamina en este contexto indica que ambos están involucrados en una respuesta metabólica similar cuando el cuerpo enfrenta un déficit de energía o una pérdida de masa muscular.

![Gráfico de correlación]

(https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/grafico_correlacion.png)

Analisis de PCA

El primer paso que hago es estandarizar los datos, así cada variable tiene media 0 y desviación estándar 1. Cálculo de la matriz de covarianza o correlación y calculo los vectores y los valores propios de la matriz de covarianza. Proyecto datos sobre los componentes principales. Y tengo en cuenta los primeros dos componentes (PC1 y PC2) porque son los más relevantes para la visualización, ya que explican la mayor parte de la variabilidad en los datos. pca_result <- prcomp(t(assays(se)$counts), scale = TRUE)

```
   # t(assays(se)$counts): Se realiza una transposición de la matriz de conteos. Porque en el análisis de comp
onentes principales (PCA), cada fila es en este caso, un gen, y cada columna una muestra.
   #prcomp: función de R que realiza el análisis de componentes principales (PCA), descompone la matriz de
datos utilizando el método de descomposición en valores singulares.

   pca_scores <- pca_result$x
```

Puedo concluir que PC1 tiene una desviación estándar significativamente mayor (7.488), por tanto esto indica que representa una gran parte de la variabilidad en los datos. Esto es consistente con l a proporción de varianza de 0.890, lo que significa que el PC1 explica el 89% de la variabilidad en l os datos. PC2 tiene una desviación estándar mucho menor (1.6894) y una proporción de varianza de 0.0453, explicando solo el 4.53% de la variabilidad en los datos.

La **proporción acumulada** muestra que después de las primeras 10 componentes principales (PC), ya se ha explicado más del 99% de la variabilidad. Por tanto, esto indica que para la mayoría de l os análisis, podrías trabajar con solo las primeras 2-3 componentes y aun así capturar casi toda la variabilidad en los datos.

![Scores PCA](https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/pca_scores.csv)

Finalmente genero un gráfico de dispersión de los primeros dos componentes principales (PC1 vs. PC2), lo que permite identificar patrones, agrupaciones o posibles anomalías en los datos. Los puntos en el gráfico representan las muestras y su distribución puede revelar patrones biológicos interesantes, como agrupaciones de muestras similares (**Anexo 2**).

6. Discusión

El presente estudio se centró en el análisis de un conjunto de datos metabolómicos que incluye mediciones de metabolitos en pacientes con caquexia y en un grupo de control, utilizando la clase *SummarizedExperiment* para organizar los datos de manera estructurada. Sin embargo, es importante reflexionar sobre las limitaciones de este estudio en el contexto del problema biológico de interés: la caquexia, una condición caracterizada por la pérdida significativa de masa muscular. Uno de los principales desafíos es el tamaño limitado de la muestra, que incluye datos de 77 pacientes (47 con caquexia y 30 controles). Un tamaño de muestra pequeño puede restringir la capacidad del análisis para detectar cambios sutiles en las concentraciones de metabolitos y dificulta la generalización de los resultados a una población más amplia. En estudios de metabolómica, contar con una muestra mayor es fundamental para mejorar la solidez estadística y reducir el riesgo de obtener falsos positivos o negativos. Había manejo de datos faltantes, Durante la preparación de los datos, fue necesario eliminar filas con valores faltantes (NA) y duplicados. La eliminación de estos datos puede llevar a la pérdida de información valiosa y sesgar el análisis si los datos faltantes no son completamente aleatorios. Aunque este enfoque es común, una alternativa sería aplicar métodos de imputación de datos que podrían preservar más información sin comprometer la calidad del análisis. El análisis exploratorio de correlaciones entre los metabolitos permite identificar asociaciones preliminares entre ciertos metabolitos y la pérdida muscular. Sin embargo, es importante destacar que la correlación no implica causalidad. Aunque se observan asociaciones, no se puede concluir si los cambios en las concentraciones de metabolitos son una causa o una consecuencia de la caquexia. Para avanzar en este aspecto, se requerirían estudios experimentales adicionales que investiguen las vías metabólicas implicadas. Los datos incluyen pacientes con caquexia, pero no se consideró la heterogeneidad de las causas subyacentes de esta condición, como el cáncer o la insuficiencia cardíaca, que pueden influir en los perfiles metabólicos. La caquexia puede tener diferentes etiologías, lo que podría afectar las concentraciones de metabolitos de manera distinta según la patología. Sería valioso estratificar a los pacientes por la causa de la caquexia en futuros estudios para identificar biomarcadores específicos relacionados con cada origen de la enfermedad. El conjunto de datos incluye un número limitado de metabolitos, que aunque relevantes para las vías metabólicas relacionadas con la función muscular y el metabolismo energético, podrían no captar la totalidad de los cambios metabólicos implicados en la caquexia. Un análisis más exhaustivo que incorpore un mayor número de metabolitos y vías metabólicas podría proporcionar una visión más completa del impacto de la caquexia en el metabolismo de los pacientes. Este estudio tiene el potencial de aportar información valiosa sobre los mecanismos metabólicos subyacentes a la caquexia, una condición compleja y multifactorial que afecta significativamente la calidad de vida de los pacientes. Identificar biomarcadores metabólicos asociados con la pérdida de masa muscular podría ser útil para el diagnóstico temprano, el seguimiento de la progresión de la

enfermedad y el desarrollo de nuevas terapias dirigidas. El uso de *SummarizedExperiment* en el análisis ha facilitado la organización y manejo de grandes volúmenes de datos experimentales y clínicos, ofreciendo una base estructurada para realizar análisis más complejos. Sin embargo, las limitaciones mencionadas deben abordarse en estudios futuros para obtener resultados más sólidos y aplicables. Aumentar el tamaño de la muestra, integrar otros tipos de datos ómicos y emplear metodologías estadísticas más avanzadas permitirá avanzar en la comprensión de la caquexia y sus implicaciones clínicas.

Finalmente yo propondría desarrollar un modelo metabólico, enfocándome en los resultados obtenidos a partir de los análisis estadísticos y de correlación entre los metabolitos, así como en las observaciones del Análisis de Componentes Principales (PCA). Este modelo metabólico se desarrollaría a partir de la identificación de metabolitos clave involucrados en la caquexia, lo que permitirá explorar de manera más detallada las rutas metabólicas y sus interacciones. El modelo podría integrar datos experimentales de metabolitos analizados en un objeto **SummarizedExperiment** y las correlaciones entre metabolitos clave. A través del PCA, se identificarían patrones y agrupamientos que reflejan la variabilidad en las muestras, ayudando a mapear las vías metabólicas alteradas en la caquexia. Utilizando técnicas computacionales como redes metabólicas y análisis de rutas biológicas, se integraría la información en un mapa metabólico, identificando puntos críticos para intervenciones terapéuticas. Esto permitiría hacer predicciones precisas sobre los procesos fisiopatológicos de la caquexia, mejorando el diseño de terapias específicas para restaurar el equilibrio metabólico en los pacientes.

7.Conclusiones

En conclusión, los metabolitos analizados, como la **creatinina**, **pi-Methylhistidine**, **oxoglutarato**, **succinato**, **acetato**, y los **aminoácidos** (alanina, glutamina, leucina, y valina) ofrecen una visión integral del complejo estado metabólico presente en la caquexia. Creatinina y pi-Methylhistidine reflejan la descomposición muscular, un proceso central en la caquexia, que se caracteriza por la pérdida acelerada de masa muscular. Niveles elevados de estos metabolitos indican un catabolismo de proteínas aumentado, lo cual es un claro indicador de daño y pérdida de tejido muscular, como hemos visto en los resultados del cálculo la media de los metabolitos la **Creatinina** y **pi-Methylhistidine** hemos vistoo valores de **9521.7754** y **391.5184** de media. **Oxoglutarato**, **succinato** y **acetato** están involucrados en la producción de energía, especialmente a través del ciclo de Krebs. Por esta razon hemos visto fuertes correlaciones entre ellos. Ya que alteraciones en estos metabolitos sugieren una disfunción metabólica, donde el cuerpo intenta compensar la falta de energía utilizando vías alternativas, como la oxidación de ácidos grasos, lo que refleja un cambio en el metabolismo energético típico de la caquexia. Finalmente, los aminoácidos como **alanina, glutamina, leucina**, y **valina** están estrechamente relacionados con la degradación y síntesis de proteínas musculares. La presencia elevada de estos aminoácidos es indicativa de un estado catabólico donde el cuerpo busca compensar la pérdida muscular mediante la movilización de reservas de aminoácidos. Además, estos metabolitos ayudan a mantener funciones críticas como la respuesta inmune y la regulación de la síntesis proteica, aunque su efectividad es limitada debido a las condiciones catabólicas.

8. Referencias

Para el análisis presentado en este proyecto, el código y los datos utilizados se encuentran disponibles en el siguiente repositorio de GitHub:

Repositorio de GitHub: https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1

Este repositorio contiene los archivos y scripts necesarios para reproducir el análisis, incluyendo:

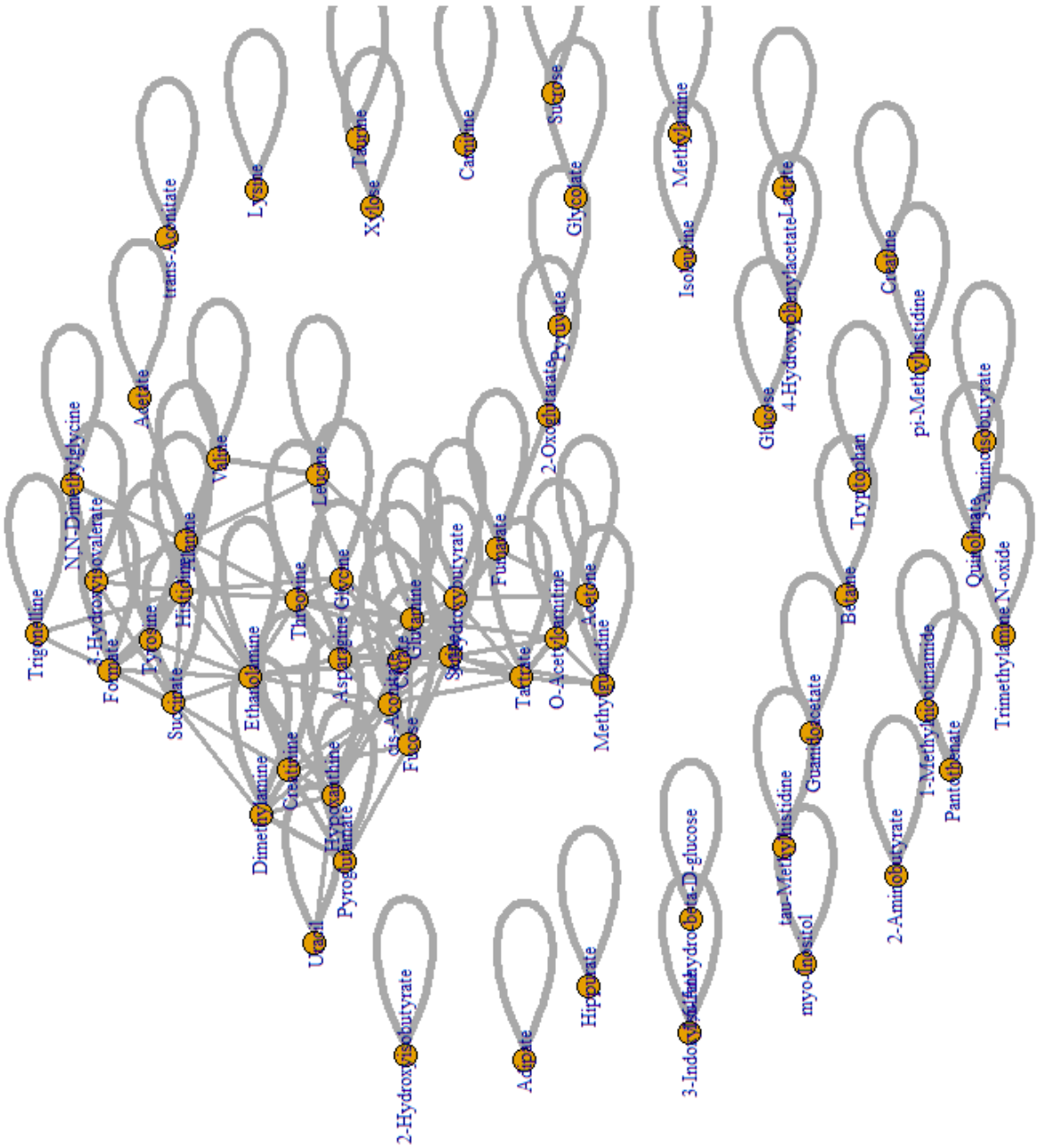
- **PEC 1.Rmd**: El código en formato RMarkdown donde se encuentra comentado el análisis.(**Anexo 5**).
- **Informe.html**: El informe generado a partir del código con los resultados del análisis., peero finalment he preferido pulir el informe una vez descargado en el ordenador.
- **SummarizedExperiment_Diana_Gutiérrez.rda**: El objeto SummarizedExperiment con los datos estructurados.
- **human_cachexia.csv** y **metabolite_data.csv**: Archivos con los datos experimentales y metadatos.
- **Imágenes y gráficos**: Capturas, gráficos de correlación, heatmap y PCA generados durante el análisis.(**Anexo 1, 2,4**)
- **Resultados numéricos**: Matrices de correlación, varianza explicada y puntuaciones de PCA.

9. Bibliografia

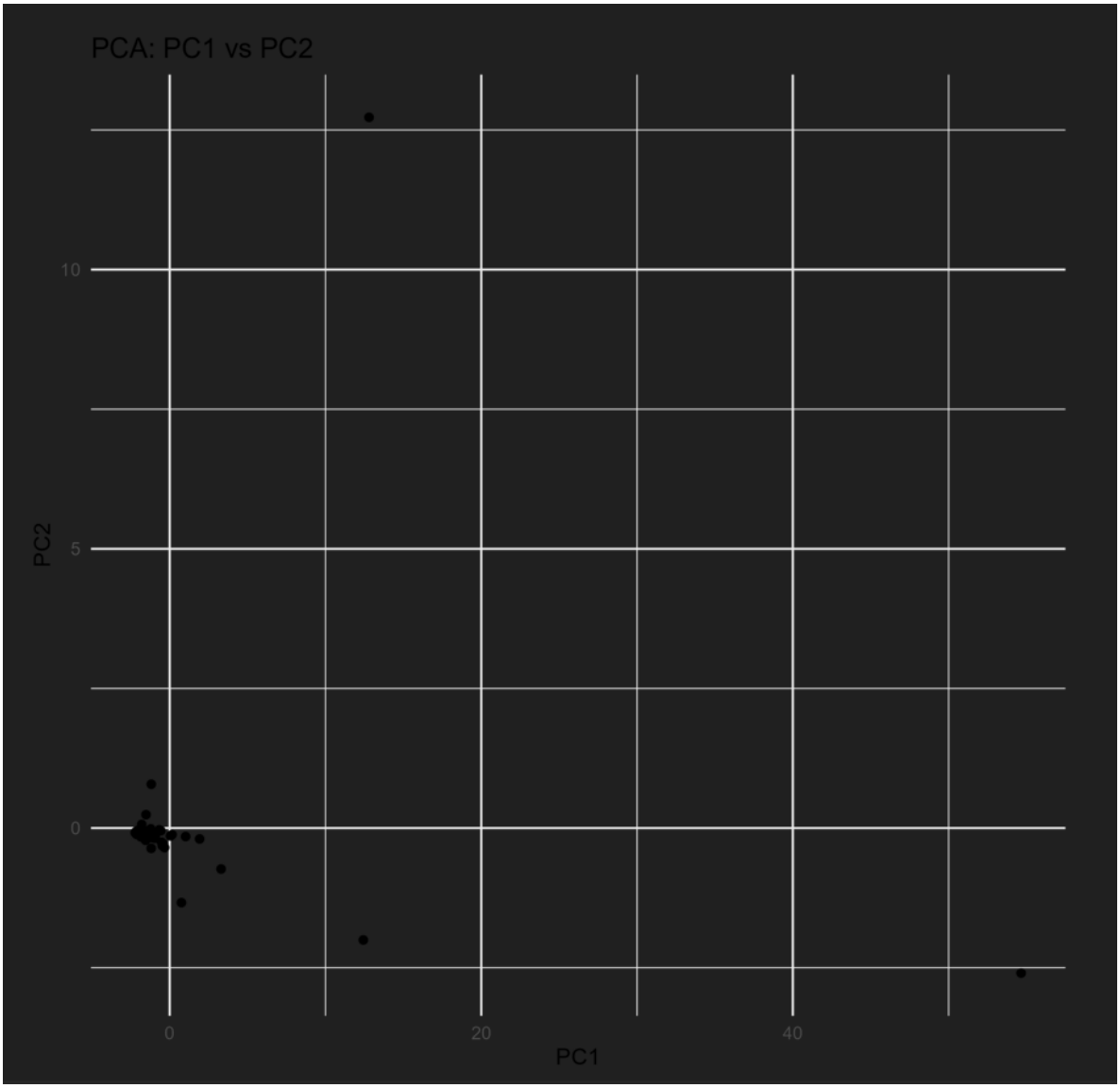
- Ciclo de Krebs (https://www.studocu.com/latam/document/universidad-tecnologica-de-santiago/bioquimica-i/cap-16-ciclo-de-krebs-harpers-bioquimica-ilustrada-30a-edicion/8335805)

- Resumen de ruta metabolica (https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1/blob/main/Captura%20de%20pantalla.png)

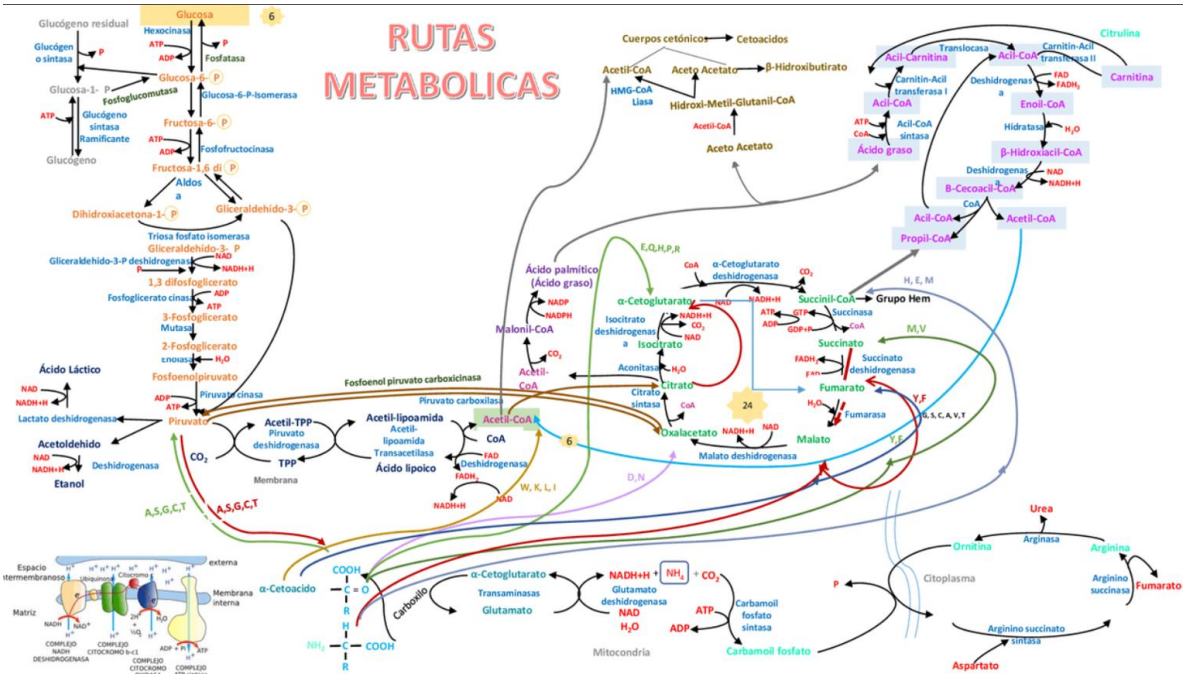
Anexo 1


Lysine
Taurine
Xylose
Caffeine
Sucrose
Glycolate
trans-Aconitate
Isoleucine
Methionine
Pyruvate
Acetate
2-Oxoglutarate
Glucose
4-Hydroxyphenylacetate
Lactate
Creatine
pi-Methylhistidine
Valine
Leucine
Tryptophan
3-Aminoisobutyrate
N,N-Dimethylglycine
Quinolinate
Trigonelline
4-Hydroxyvalerate
Tyrosine
Histidine
Betaine
Trimethylamine N-oxide
Formate
Threonine
Fumarate
Glutamine
Succinate
Ethanolamine
Asparagine
3-Hydroxybutyrate
Acetone
Tartrate
O-Acetylcarnitine
Methylguanidine
Fucose
cis-Aconitate
Creatinine
Guanidoacetate
Dimethylamine
Hypoxanthine
1-Methylnicotinamide
Pantothenate
Pyroglutamate
Uracil
tau-Methylhistidine
2-Aminobutyrate
myo-Inositol
2-Hydroxybutyrate
Adipate
Hippurate
3-Indoxyl sulfate
beta-D-glucose

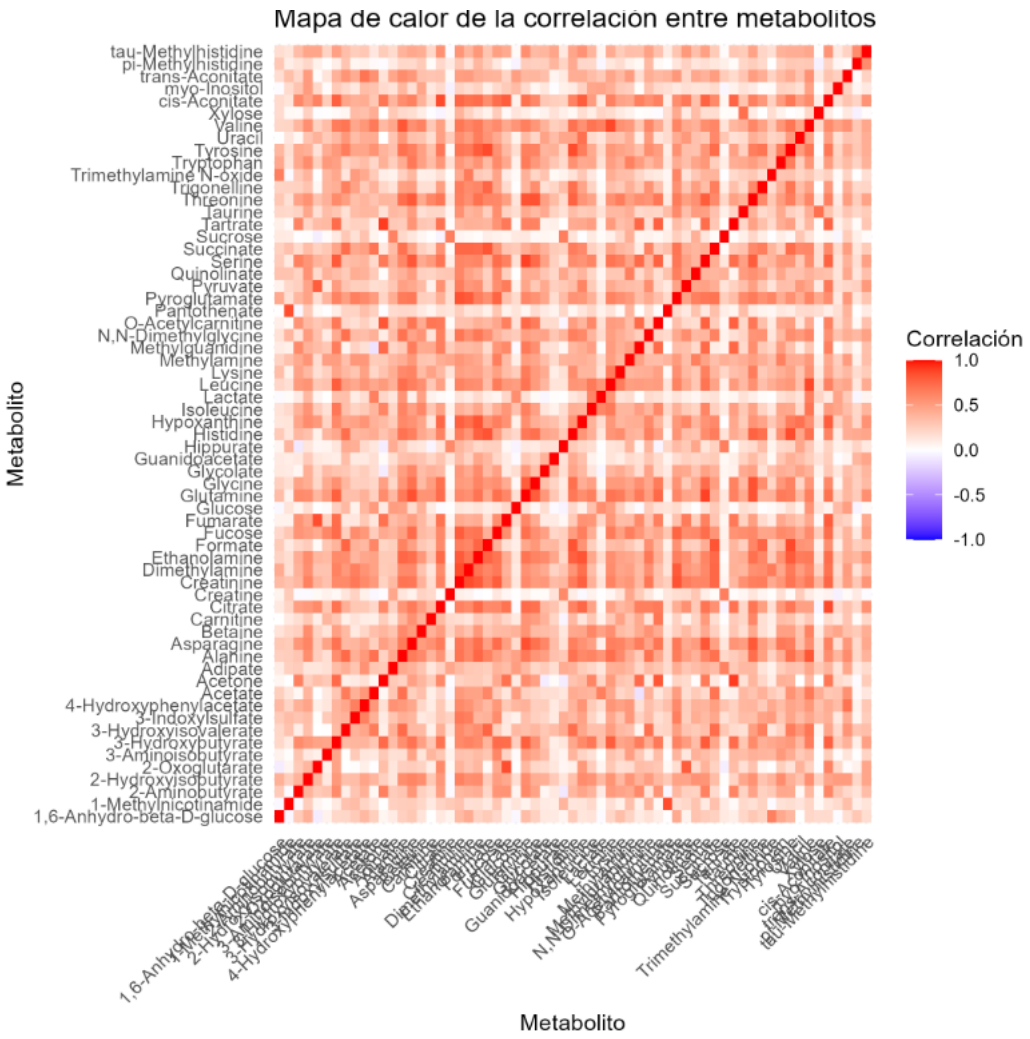
Anexo 2



Anexo 3


RUTAS
METABOLICAS
Glucosa
Glucógeno residual
Hexocinasa
Fosfatasa
Glucosa-6-P-isomerasa
Fosfofructocinasa
Aldosa
Triosa fosfato isomerasa
Gliceraldehido-3-P deshidrogenasa
1,3 difosfoglicerato
3-Fosfoglicerato Mutasa
2-Fosfoglicerato
Fosfoenolpiruvato
Piruvato cinasa
Piruvato
Lactato deshidrogenasa
Ácido Láctico
Acetoldehido
Etanol
Deshidrogenasa
Acetil-TPP
Piruvato deshidrogenasa
TPP
Membrana
Acetil-lipoamida
Transacetilasa
Ácido lipoico
Acetil-CoA
Fosfoenol piruvato carboxicinasa
Piruvato carboxilasa
Cuerpos cetónicos
Cetoacidos
Acetil-CoA
Aceto Acetato
β-Hidroxibutirato
HMG-CoA Liasa
Hidroxi-Metil-Glutanil-CoA
Ácido palmítico (Ácido graso)
Malonil-CoA
α-Cetoglutarato
Isocitrato deshidrogenasa
Isocitrato
Aconitasa
Citrato
Citrato sintasa
Oxalacetato
Malato
Malato deshidrogenasa
Fumarasa
Fumarato
Succinato
Succinato deshidrogenasa
Succinil-CoA
Grupo Hem
Succinasa
α-Cetoglutarato deshidrogenasa
Translocasa
Acil-Carnitina
Carnitin-Acil transferasa I
Carnitin-Acil transferasa II
Carnitina
Acil-CoA
Acil-CoA sintasa
Ácido graso
Deshidrogenasa
Enoil-CoA
Hidratasa
β-Hidroxiacil-CoA
β-Cecoacil-CoA
Propil-CoA
Citrulina
Urea
Arginasa
Ornitina
Arginina
Arginino succinasa
Arginino succinato sintasa
Aspartato
Citoplasma
Mitocondria
Carbamoil fosfato sintasa
Carbamoil fosfato
Glutamato deshidrogenasa
Glutamato
Transaminasas
α-Cetoacido
Carboxilo
Membrana interna
Matriz
Espacio intermembranoso
externa

Anexo 4


Mapa de calor de la correlación entre metabolitos
Metabolito
Metabolito
Correlación
1.0
0.5
0.0
-0.5
-1.0

Anexo 5 – Explicación codigo R script formato RMarkdown

```{r}
# Cargar las librerías necesarias, asegurando que estén instaladas
required_packages <- c("usethis", "BiocManager", "readr", "ggplot2", "dplyr", "patchwork", "GGally", "tidyr", "reshape2", "igraph", "KEGGREST")

new_packages <- required_packages[!(required_packages %in% installed.packages()[,"Package"])]
if(length(new_packages)) install.packages(new_packages)

lapply(required_packages, library, character.only = TRUE)

# Leer los datos CSV
data <- read.csv("C:/Users/merma/OneDrive/Escritorio/MASTER BIOINFORMATICA/analisis de datos omicos/Gutiérrez -Martínez -Diana-PEC1/human_cachexia.csv")

# Extraer los metadatos y datos experimentales
# Explicación de los codigos : Este código selecciona dos columnas del data y las guarda en el objeto sample_data.
sample_data <- data[, c("Patient.ID", "Muscle.loss")]

sample_data$Muscle.loss <- factor(sample_data$Muscle.loss)
# He convertido la columna Muscle.loss en un factor

# Quiero seleccionar todas las filas pero excluir las columnas 1 y 2.

exp_data <- data[, -(1:2)]
data_matrix <- as.matrix(exp_data)

# Asegurar que los nombres de fila están asignados correctamente

rownames(data_matrix) <- sample_data$Patient.ID

# Quiero eliminar las filas del data frame sample_data que tengan valores duplicados en la columna Patient.ID.
sample_data <- sample_data[!duplicated(sample_data$Patient.ID), ]

# Elimino las filas del objeto data_matrix (una matriz) que tengan nombres de fila duplicados.
data_matrix <- data_matrix[!duplicated(rownames(data_matrix)), ]

# Quiero encontrar los Patient.ID que están presentes tanto en sample_data como en data_matrix
common_ids <- intersect(sample_data$Patient.ID, rownames(data_matrix))

# Filtro el sample_data para que solo contenga las filas cuyos Patient.ID están presentes en common_ids
sample_data <- sample_data[sample_data$Patient.ID %in% common_ids, ]

# Filtro data_matrix para que solo contenga las filas cuyos nombres de fila coinciden con los common_ids.
data_matrix <- data_matrix[rownames(data_matrix) %in% common_ids, ]
```

```
# Y creo el objeto SummarizedExperiment
se <- SummarizedExperiment(assays = list(counts = data_matrix), colData = sample_data)

# Guardo el objeto SummarizedExperiment
save(se, file = "C:/Users/merma/OneDrive/Escritorio/MASTER BIOINFORMATICA/analisis de datos omicos/Gutiérrez -Martínez -Diana-PEC1/SummarizedExperiment_Diana_Gutiérrez.rda")

# Y hago un análisis exploratorio de los datos
# Ver los metadatos

colData(se)
# asi veo el dataframe con una tabla con los metadatos de las muestras (columnas) en el objeto SummarizedExperiment

summary(se)

# Hacer un t.test

head(se)

muscle_loss <- colData(se)$Muscle.loss  # Esto debería estar alineado con las columnas de la matriz

# Separar los grupos 'control' y 'cachexic'
control_group <- assays(se)$counts[, muscle_loss == "control"]  # Filtro por columnas
case_group <- assays(se)$counts[, muscle_loss == "cachexic"]

sample_data <- sample_data[match(rownames(assays(se)$counts), sample_data$Patient.ID), ]

head(sample_data$Muscle.loss)

p_values <- apply(assays(se)$counts, 1, function(x) {
  t.test(x[sample_data$Muscle.loss == "control"], x[sample_data$Muscle.loss == "cachexic"])$p.value
})

head(p_values)

sorted_p_values <- sort(p_values)

print(sorted_p_values)

# Los p-valores son mayores que 0.05, no podemos rechazar la H0 (que afirma  que no hay diferencias entre los grupos). Por tanto, no encontramos evidencia suficiente para decir que los metabolitos analizados difieren significativamente entre los grupos de pacientes cachexicos y controles.

# Calcular y ver la media de cada metabolito

mean_values <- apply(assays(se)$counts, 2, mean) # las medias de los datos.

sorted_mean_values_desc <- sort(mean_values, decreasing = TRUE)
```

```
head(sorted_mean_values_desc, 10) # Ver el encabezado de los datos.

# Respuesta: La Creatinine tiene el valor más alto de concentración entre los metabolitos estudiados, con 9521.7754, seguido por Hippurate (2614.1479) y Citrate (2423.4689). Esto sugiere que la creatinina podría ser uno de los metabolitos más presentes en las muestras que estás analizando.


# Calcular la desviación estándar por metabolito
sd_per_metabolite <- apply(assays(se)$counts, 2, sd)

# Tabla de frecuencias de la variable Muscle.loss

table(sample_data$Muscle.loss) # Aqui veo que la distribución no es homogénea.

    library(ggplot2)

    ggplot(sample_data, aes(x = Muscle.loss)) +
      geom_bar(fill = "skyblue", color = "black") +
      labs(title = "Distribución de muestras por grupo", x = "Grupo", y = "Frecuencia") +
      theme_minimal()


# Visualización de la correlación
cor_matrix <- cor(data_matrix, use = "pairwise.complete.obs")
write.csv(cor_matrix, "C:/Users/merma/OneDrive/Escritorio/MASTER BIOINFORMATICA/analisis de datos omicos/Gutiérrez -Martínez -Diana-PEC1/correlation_matrix.csv")

# Visualizar la matriz de correlación con un mapa de calor
cor_data <- melt(cor_matrix)
ggplot(cor_data, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(title = "Mapa de calor de la correlación entre metabolitos", x = "Metabolito", y = "Metabolito")
ggsave("C:/Users/merma/OneDrive/Escritorio/heatmap.png")

# PCA (Análisis de Componentes Principales)
pca_result <- prcomp(t(assays(se)$counts), scale = TRUE)

# t(assays(se)$counts): Se realiza una transposición de la matriz de conteos. Porque en el análisis de componentes principales (PCA), cada fila es en este caso, un gen, y cada columna una muestra.
#prcomp: función de R que realiza el análisis de componentes principales (PCA), descompone la matriz de datos utilizando el método de descomposición en valores singulares.

pca_scores <- pca_result$x
# Asi obtengo los datos de los scores.

# Que me lo transforme en un dataframe.

pca_scores_df <- data.frame(pca_scores)
```

```
write.csv(pca_scores_df, "C:/Users/merma/OneDrive/Escritorio/MASTER
BIOINFORMATICA/analisis de datos omicos/Gutiérrez -Martínez -Diana-PEC1/pca_scores.csv")
# Guardado en github.

# Graficar los primeros dos componentes principales (PC1 vs PC2)  en un gráfico de dispersión

ggplot(pca_scores_df, aes(x = PC1, y = PC2)) +
  geom_point() +
  labs(title = "PCA: PC1 vs PC2", x = "PC1", y = "PC2") +
  theme_minimal()

ggsave("C:/Users/merma/OneDrive/Escritorio/pca_plot.png")

# agregado el grafico en github (https://github.com/Dianaguma/Gutierrez_Martinez_Diana_PEC1)

# Varianza explicada y acumulada
varianza_explicada <- pca_result$sdev^2 / sum(pca_result$sdev^2)
varianza_acumulada <- cumsum(varianza_explicada)

varianza_df <- data.frame(PC = paste0("PC", 1:length(varianza_explicada)),
                          Varianza_Explicada = varianza_explicada,
                          Varianza_Acumulada = varianza_acumulada)
# resultados analizados en el informe.

write.csv(varianza_df, "C:/Users/merma/OneDrive/Escritorio/MASTER BIOINFORMATICA/analisis
de datos omicos/Gutiérrez -Martínez -Diana-PEC1/varianza_explicada.csv", row.names = FALSE)

# La descargo para ponerla en github.

ggplot(varianza_df, aes(x = PC)) +
  geom_bar(aes(y = Varianza_Explicada), stat = "identity", fill = "blue", alpha = 0.6) +
  geom_line(aes(y = Varianza_Acumulada * max(varianza_explicada), group = 1), color = "red",
linewidth = 1) +
  labs(title = "Varianza Explicada y Acumulada por los Componentes Principales", x = "Componente
Principal", y = "Proporción de Varianza / Acumulada") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(sec.axis = sec_axis(~./max(varianza_explicada), name = "Proporción
Acumulada"))
ggsave("C:/Users/merma/OneDrive/Escritorio/varianza_explicada.png", width = 10, height = 6, dpi =
300)

# Extraer y guardar los resultados del PCA y adjuntar al github.
write.csv(pca_scores_df, file = "C:/Users/merma/OneDrive/Escritorio/pca_scores.csv")


# Correlación filtrada y gráfico de la red de metabolitos
threshold <- 0.7
cor_matrix_filtered <- cor_matrix

# abs: obtener el valor absoluto de  en este caso una matriz.

cor_matrix_filtered[abs(cor_matrix_filtered) < threshold] <- 0
g <- graph_from_adjacency_matrix(cor_matrix_filtered, weighted = TRUE, mode = "undirected")
png("C:/Users/merma/OneDrive/Escritorio/grafico_correlacion.png", width = 800, height = 800)
```

```
plot(g, vertex.label = rownames(cor_matrix_filtered), vertex.size = 5, edge.width = E(g)$weight * 5)
dev.off()
```

```
```