Análisis de Datos Ómicos - PEC1

Luciano Prieri

23/03/2025

Contents

1 Tabla de Contenidos

Sección
Resumen
Introducción
Objetivos
Métodos
Resultados
Discusión
Conclusiones
Referencias

2 Resumen

Este estudio analiza datos de metabolómica obtenidos a partir de espectroscopia de resonancia magnética nuclear de 1H (1H-RMN) con el objetivo de identificar un perfil metabolómico urinario característico del cáncer gástrico (CG). Se utilizaron muestras de 43 pacientes con CG, 40 con enfermedad gástrica benigna (BN) y 40 individuos sanos (HE), midiendo 129 metabolitos. Se aplicaron métodos de análisis exploratorio como PCA, heatmaps y coeficientes de variación (CV) en controles de calidad (QC). Se identificó un efecto de lote experimental y se filtraron metabolitos con CV >20%. La normalización por suma total permitió corregir variaciones técnicas. El análisis reveló patrones metabólicos distintivos en CG, destacando su potencial clínico en diagnóstico temprano.

3 Introducción

Contexto Biológico

El cáncer gástrico (CG) representa un 5.6% de todos los cánceres a nivel mundial, con una supervivencia a 5 años inferior al 30% (Wang et al., 2020). Estudios recientes sugieren que alteraciones en el metaboloma podrían servir como biomarcadores tempranos. Este trabajo explora datos de resonancia magnética nuclear (RMN) para identificar patrones metabólicos asociados a CG

4 Objetivos

- Integrar los datos metabolómicos en un formato estandarizado (**SummarizedExperiment**).
- Evaluar la calidad técnica mediante coeficientes de variación en QCs y análisis de efecto de lote.
- Explorar patrones metabólicos utilizando técnicas multivariantes (PCA, heatmaps).
- Generar un informe reproducible con código en **R** y resultados accesibles en **GitHub**.

5 Métodos

5.1 Origen y Naturaleza de los Datos

Los datos utilizados en este estudio provienen del dataset ST001047 disponible en Metabolomics Workbench, que contiene información de la abundancia de 129 metabolitos identificados mediante espectroscopia de resonancia magnética nuclear de protón (1H-RMN) en muestras de orina. Este dataset fue seleccionado debido a que proporciona un enfoque no invasivo para el perfilado metabólico en cáncer gástrico, ofreciendo una posible vía para el descubrimiento de biomarcadores y el diagnóstico temprano. El estudio original incluyó muestras de orina de 43 pacientes diagnosticados con cáncer gástrico (CG), 40 con enfermedad gástrica benigna (BN) y 40 individuos sanos (HE). Además, el dataset incluye réplicas técnicas (muestras de Control de Calidad, QCs) que se utilizaron para evaluar la estabilidad y reproducibilidad de las mediciones. . Si bien el diseño original del estudio podría haber incluido información clínica más detallada, este análisis se centra en los datos metabolómicos para identificar posibles biomarcadores que diferencien el cáncer gástrico de las condiciones benignas y los controles sanos. Una limitación clave es la falta de metadatos clínicos detallados que permitirían un análisis más completo, lo cual impide explorar correlaciones entre los metabolitos y características clínicas de los pacientes

5.2 Herramientas y Procedimiento

Se utilizó **R** (v4.3.1) y paquetes como `SummarizedExperiment`, `ggplot2`, `pheatmap`, `FactoMineR` y `Bioconductor`. El flujo de trabajo incluyó:

5.2.1 Pasos del Análisis:

1. **Preprocesamiento:**
   - Separación de metadatos (columnas 1–7) y datos de metabolitos (columnas 8–136)

- Limpieza y transformación de datos (log2 y eliminación de NA).
- Creación de un objeto **SummarizedExperiment** que integra datos crudos y transformados (log2).

2. **Control de Calidad:**
   - Cálculo de coeficientes de variación (CV) en QCs.
   - Filtrado de metabolitos con CV >20%.
3. **Análisis Exploratorio:**
   - PCA para evaluar variabilidad técnica y biológica.
   - Heatmap de correlación entre muestras.
4. **Normalización:**
   - Normalización por suma total para corregir diferencias globales de intensidad..

6 Resultados

6.1 Exploración Inicial de los Datos

Se exploró la estructura de los datos para confirmar que la matriz de expresión contiene 140 muestras y 129 metabolitos, con las muestras identificadas de "sample_1" a "sample_140" y los metabolitos etiquetados de "M1" a "M129". Esta verificación es crucial para asegurar que la integración de datos en el objeto SummarizedExperiment se realizó correctamente.

```
## [1]  42 140
```

```
##     sample_1 sample_2  sample_3 sample_4 sample_5 sample_6 sample_7 sample_8
## M4  5.169925       NA  5.578939 3.906891 3.277985 4.300124       NA 4.263034
## M5  7.368070 9.441907  8.920055 6.485427 7.931919 7.651769 8.506605 6.199672
## M7  5.392317 5.281698  6.795715 7.420381 8.452859 5.259272 5.921246 4.026800
## M8  5.569856 6.985273  6.427941 4.638074 5.956521 7.934870 5.708739 5.251719
## M11 5.970394 8.941341 11.253966 7.146696 5.635174 6.710118 5.885086 5.783980
## M14 5.181898       NA  4.921246 5.997744 6.300124 5.736064 5.153805 4.968091
##     sample_9 sample_10 sample_11 sample_12 sample_13 sample_14 sample_15
## M4  3.232661  5.209453  5.596935  5.523562  5.388878  4.754888  6.161888
## M5  8.083213  7.675251  7.204571  5.990955  6.753551  6.592457  6.053111
## M7  7.746850  5.504620  4.432959  5.439623  3.392317  3.510962  4.765535
## M8  6.057450  5.632268  6.031219  6.108524  5.993221  4.776104  6.361066
## M11 6.553053  5.906891  5.303781  6.668176  5.887525  7.754888  6.223036
## M14 5.974988  4.877744  5.426265  5.539159  6.961160  4.329124  6.596935
##     sample_16 sample_17 sample_18 sample_19 sample_20 sample_21 sample_22
## M4         NA  5.703211  3.847997  5.371559  4.626439  4.862947  5.711495
## M5   7.600656  5.523562  5.705978  7.500643  3.847997  8.470862  7.842979
## M7   6.207502  5.882643  4.620586  5.557655  3.711495  6.824004  4.744161
## M8   4.986411  6.159871  4.852998  5.669594  5.590961  6.215290  7.703211
## M11  4.491853  7.347843  5.887525  6.601399  8.806066        NA  8.916775
## M14        NA  7.323730  4.722466  5.017922  6.042207  5.850499  8.213347
##     sample_23 sample_24 sample_25 sample_26 sample_27 sample_28 sample_29
## M4   3.169925  5.000000  6.536053  2.807355  4.847997  5.388878  5.894818
## M5   5.296457 10.977065  9.110353  8.259743  7.990388  7.263034  7.560715
## M7   3.722466  8.537995  7.446256  5.762880  4.121015  5.399171  7.386294
## M8   4.307429  6.341630  6.831624  6.133399  5.240314  5.694880  5.770829
## M11  7.272397        NA  6.729281  6.835419  4.744161  6.368070  2.744161
## M14  5.680887  6.962318  7.059615  5.697663  4.940167  5.000000  6.542258
##     sample_30 sample_31 sample_32 sample_33 sample_34 sample_35 sample_36
## M4   5.314697  7.927778  5.889960  4.837943  5.482203  6.155830 3.4982509
## M5   7.794416  7.418696  7.834155  7.334497  7.729961  5.482203 5.0617762
## M7   8.579316  6.228819  4.940167  5.266787  3.169925  4.872829 4.4262648
```

```
## M8   5.478972  6.193772  6.894818  6.441284  5.736064  6.643856 4.7169909
## M11  6.221104  8.693487  6.290940  5.820179  5.240314  8.212375 0.7655347
## M14  6.366322  5.997744  6.638074  4.911692  5.956521  6.730640 6.2268938
##     sample_37 sample_38 sample_39 sample_40 sample_41 sample_42 sample_43
## M4   5.116864  5.638074  6.791814        NA  5.675251  3.277985  3.887525
## M5   7.404290  6.914086  8.068778  8.949827  6.362821  4.744161  7.500643
## M7   5.478972  6.165912  8.311067  6.765535        NA  3.523562  6.017922
## M8   5.649615  6.369815  6.368070  7.682292  6.068241  3.364572  6.110614
## M11  6.390599  6.627899  5.916477  7.364572  8.918565  4.432959  9.138016
## M14  5.044394  6.268659  5.325530  5.835419  6.098032  3.837943  7.921841
##     sample_44 sample_45 sample_46 sample_47 sample_48 sample_49 sample_50
## M4  3.2016339  6.203593  5.189825  5.830357        NA  3.087463  1.378512
## M5  4.9954845  6.093814  7.625709  8.474112  6.087463  4.542258  6.797013
## M7         NA  4.112700  5.449561  6.000000  7.681590  3.827819  5.997744
## M8  3.7970130  5.683696  5.587965  6.703211  5.409391  5.357552  6.044394
## M11 4.8429788  7.451211  6.580447  6.811214  6.114783  4.906891  6.083213
## M14 0.2630344  6.846744  4.738768  7.473300        NA  6.694880  6.093814
##     sample_51 sample_52 sample_53 sample_54 sample_55 sample_56 sample_57
## M4   6.417853        NA  3.608809  5.860466  4.972693        NA  5.686501
## M5   8.529040  5.013462  4.727920  6.289097  7.522778  5.529821  8.737416
## M7   4.902074  3.378512  7.181898  4.378512  5.378512  4.705978  6.320124
## M8   6.941341  4.357552  3.364572  4.510962  5.666757  4.542258  6.817623
## M11        NA  5.300124  4.419539  5.419539  6.534497  5.031219  7.075747
## M14  7.977852  3.666757  4.554589  5.244126  5.066089  4.694880  5.517276
##     sample_58 sample_59 sample_60 sample_61 sample_62 sample_63 sample_64
## M4   4.161888  4.017922  5.655352  3.584963  5.343408  6.177918  5.137504
## M5   9.025140  7.847371  7.899055  1.887525  6.035624  8.824322  7.394034
## M7   8.673839  4.336283  4.857981  7.793766  4.255501  5.545351  5.388878
## M8   7.116864  6.076816  6.931919  4.802193  5.997744  7.339850  5.623516
## M11  8.468420  7.372430  7.043301  5.700440  6.920055  8.040016  6.387156
## M14  7.627899  5.100137  6.686501  4.872829  6.213347  6.830357  5.022368
##     sample_65 sample_66 sample_67 sample_68 sample_69 sample_70 sample_71
## M4   4.061776        NA 6.7865964        NA  4.385431  5.545351        NA
## M5   6.757557  8.515700 8.9994363  5.596935        NA  6.044394  4.602884
## M7   6.871597  4.760221 8.1142629  7.934281  4.536053  4.177918  6.574404
## M8   4.459432  6.892391 6.9909549  4.232661  5.827819  5.652486  4.026800
## M11  5.364572  5.296457 0.3785116  4.632268  1.807355  6.361066  5.584963
## M14  4.632268  6.421223 5.9588427  5.247928  6.885086  5.026800  4.791814
##     sample_72 sample_73 sample_74 sample_75 sample_76 sample_77 sample_78
## M4  0.6780719  4.786596  7.101188  4.770829  6.515700  7.157852  4.137504
## M5  7.1241213  7.474112  7.889960  8.580447  6.125155  8.497852  7.147714
## M7  4.8579810  5.517276  5.590961  8.211402  4.517276  5.314697  5.439623
## M8  5.7970130  5.491853  8.063934  5.378512  5.495056  5.875289  5.255501
## M11 5.8554914  6.234578  8.960002  8.227857  6.091700  6.480588        NA
## M14 6.3344968  4.958843  7.537607  5.532940  6.137504  4.419539  4.364572
##     sample_79 sample_80 sample_81 sample_82 sample_83 sample_84 sample_85
## M4   7.450386  4.087463  1.632268  5.091700  6.338068  4.722466  4.357552
## M5   8.521208  5.620586  5.066089  7.333603  6.091700  6.978424  6.249825
## M7   7.816344  3.827819  3.336283  5.412782  2.485427  5.332708  3.350497
## M8   7.031219  5.381975  4.193772  5.526695  5.661065  5.399171  5.104337
## M11  7.906289  6.902074  5.116864  6.375039  6.272397  8.054197  5.882643
## M14  7.069315  6.414474  4.292782  5.240314  5.703211  5.399171  5.153805
##     sample_86 sample_87 sample_88 sample_89 sample_90 sample_91 sample_92
## M4   5.274262  6.336283  5.867896  5.371559  7.187847  4.949535        NA
```

```
## M5   9.089583  7.985273  8.657140  7.998872  8.848623  7.199672  8.333603
## M7   7.417009  4.247928  4.791814  5.169925  5.789208  5.507795  8.133399
## M8   6.217231  6.177918  5.169925  6.226894  6.989820  5.485427  4.584963
## M11        NA  6.782671 10.722124  5.526695  8.098032  6.240314  5.217231
## M14  5.786596  6.292782  5.689299  5.145677  7.608809  4.872829  5.177918
##     sample_93 sample_94 sample_95 sample_96 sample_97 sample_98 sample_99
## M4   5.852998  5.560715  4.842979  5.536053  5.405992  3.944858  4.307429
## M5         NA  6.549977  9.008989  7.774787  7.461889  1.201634  5.169925
## M7   4.209453  3.263034  8.864186  5.307429  3.560715  5.000000  3.776104
## M8   4.523562  5.605850  5.809929  5.963474  5.663914  6.312883  5.794416
## M11  7.500643  4.446256  7.687201  6.024586  4.925999        NA  4.847997
## M14  2.560715  5.786596  5.149747  6.521993  5.935460  5.683696  5.436295
##     sample_100 sample_101 sample_102 sample_103 sample_104 sample_105
## M4    4.797013   5.741467   4.754888         NA   4.689299   5.354029
## M5    7.454505   7.418696   5.778734   6.048759   5.177918  11.290019
## M7    5.442943   5.532940   3.485427   3.485427   3.776104   7.428779
## M8    5.465974   6.899659   5.635174   5.669594   4.193772   6.633722
## M11   6.311067   9.993505   5.857981   4.649615   8.428360   7.164907
## M14   4.867896   6.145677   5.469235   5.954196   5.842979   6.404290
##     sample_106 sample_107 sample_108 sample_109 sample_110 sample_111
## M4    5.240314   6.665336   3.137504   5.255501  5.8504994   7.676662
## M5    8.385000   7.760221   7.174926   7.182891  7.3663222  11.066156
## M7    5.491853   5.053111   3.711495   5.392317  4.2555007   6.397461
## M8    4.754888   6.095924   4.791814   5.566815  4.1292830   9.038919
## M11   8.270996   6.972693   4.584963   6.627899  9.8339969  11.386563
## M14   5.321928   6.669594   5.236493   4.990955  0.5849625   8.265849
##     sample_112 sample_113 sample_114 sample_115 sample_116 sample_117
## M4    5.354029   4.193772   5.708739   4.044394   5.205549   5.074677
## M5    5.781360   5.733354   7.380245   6.205549   8.008989   8.255973
## M7    7.262095   6.507795   5.703211   4.137504   6.316508   4.968091
## M8    8.140063   4.791814   5.906891   5.705978   5.266787   6.266787
## M11         NA   5.232661   6.949535   8.262095   4.935460   2.459432
## M14   6.165912   4.491853   5.965784   5.336283   5.495056   6.371559
##     sample_118 sample_119 sample_120 sample_121 sample_122 sample_123
## M4    5.070389   5.643856   4.548437  0.1375035   4.350497   1.485427
## M5    7.298292   6.294621   7.238405  6.8923910   6.325530   7.333603
## M7    5.354029         NA   5.923625  5.6948802   3.887525   5.436295
## M8    5.465974   6.847997   6.296457  5.2816983   6.070389   4.232661
## M11   6.437960   6.766860   6.832890  7.3165078   7.723832   6.242221
## M14   4.921246   4.321928   6.574404  6.1878469   8.776762   4.169925
##     sample_124 sample_125 sample_126 sample_127 sample_128 sample_129
## M4    5.346957   6.449561   5.078951   5.181898   6.226894   3.053111
## M5    7.054197   7.271463   6.575917   7.404290   6.087463   9.686325
## M7    4.554589   5.017922   4.112700   5.332708   4.145677   5.749534
## M8    5.770829   6.495056   4.452859   5.593951   4.548437   6.655352
## M11   7.768184   7.539934   9.248639   6.462707   4.655352   9.614342
## M14   5.862947   7.191800   5.378512   4.925999   5.456149   6.449561
##     sample_130 sample_131 sample_132 sample_133 sample_134 sample_135
## M4    4.694880   5.296457   2.887525   5.776104   7.015694   5.300124
## M5    5.776104   7.398316   5.850499   5.510962   8.346957   5.904484
## M7    5.296457         NA   4.620586   3.847997   5.318317   5.436295
## M8    5.722466   6.517276   4.491853   5.141596   7.348728   5.984134
## M11   7.130313   9.114003   5.201634   7.076816   6.829088   4.596935
## M14   6.434628   7.141596   4.897240   4.847997   7.075747   5.321928
```

```
##     sample_136 sample_137 sample_138 sample_139 sample_140
## M4    5.289097   6.537607   4.300124   4.452859   4.857981
## M5    7.451211   9.517866   5.385431   4.201634   7.388878
## M7    5.300124   8.947199   5.569856   3.797013   5.472488
## M8    5.563768   6.586465   4.749534   3.700440   5.596935
## M11   6.355792   7.612500   5.692092   4.504620   5.820179
## M14   4.852998   7.389739   4.770829   3.797013   4.972693
```

6.2 Análisis de Calidad

Se calcularon los coeficientes de variación (CV) en las muestras QC, obteniéndose un CV global entre 2.43 y 19.15, lo que indica una buena estabilidad técnica de la mayoría de los metabolitos. La ausencia de valores NA en los CV confirma que los datos son consistentes para su análisis. Posteriormente, se filtraron aquellos metabolitos con CV >20%, lo que redujo la variabilidad técnica y mejoró la robustez del conjunto de datos.

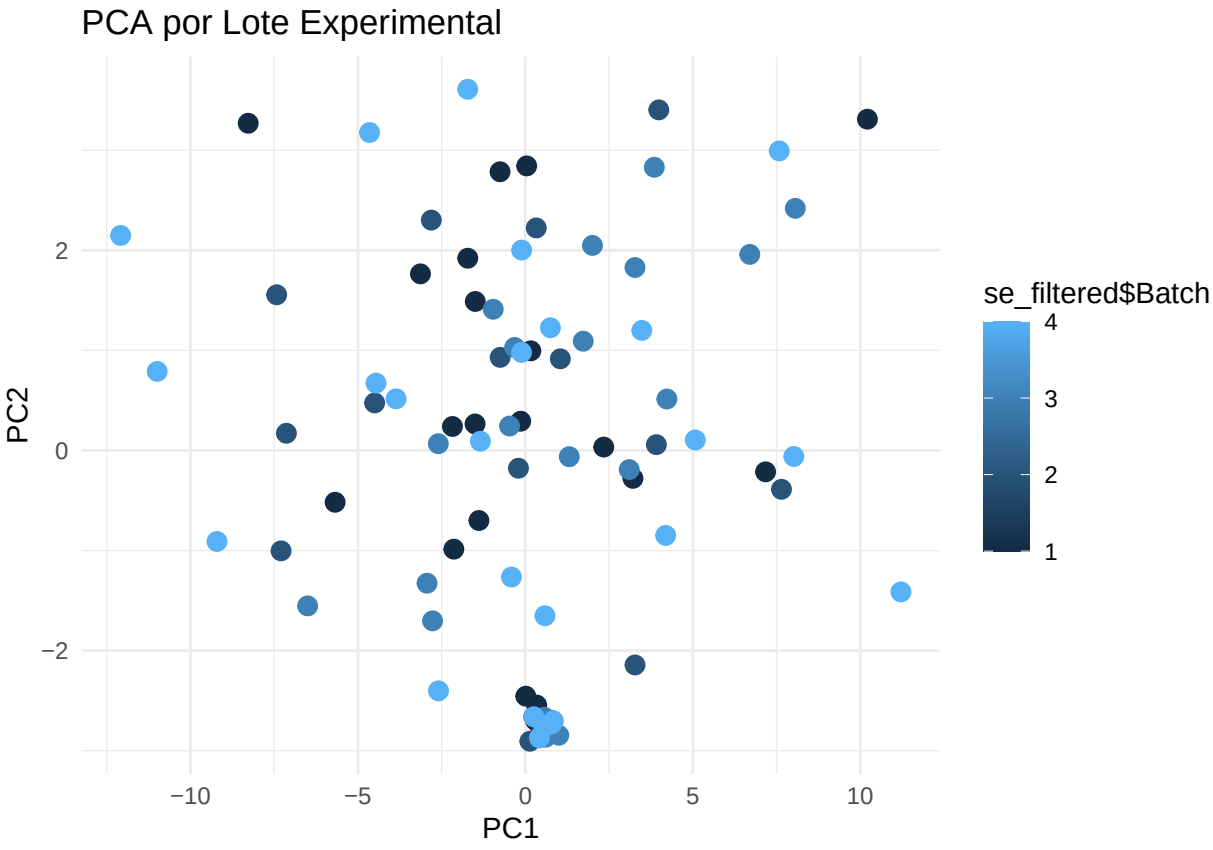
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.432   6.122  10.501  10.460  14.752  19.146
```

```
## Número de valores NA en CV: 0
```

6.3 Análisis Exploratorio

6.3.1 Análisis de Componentes Principales (PCA)

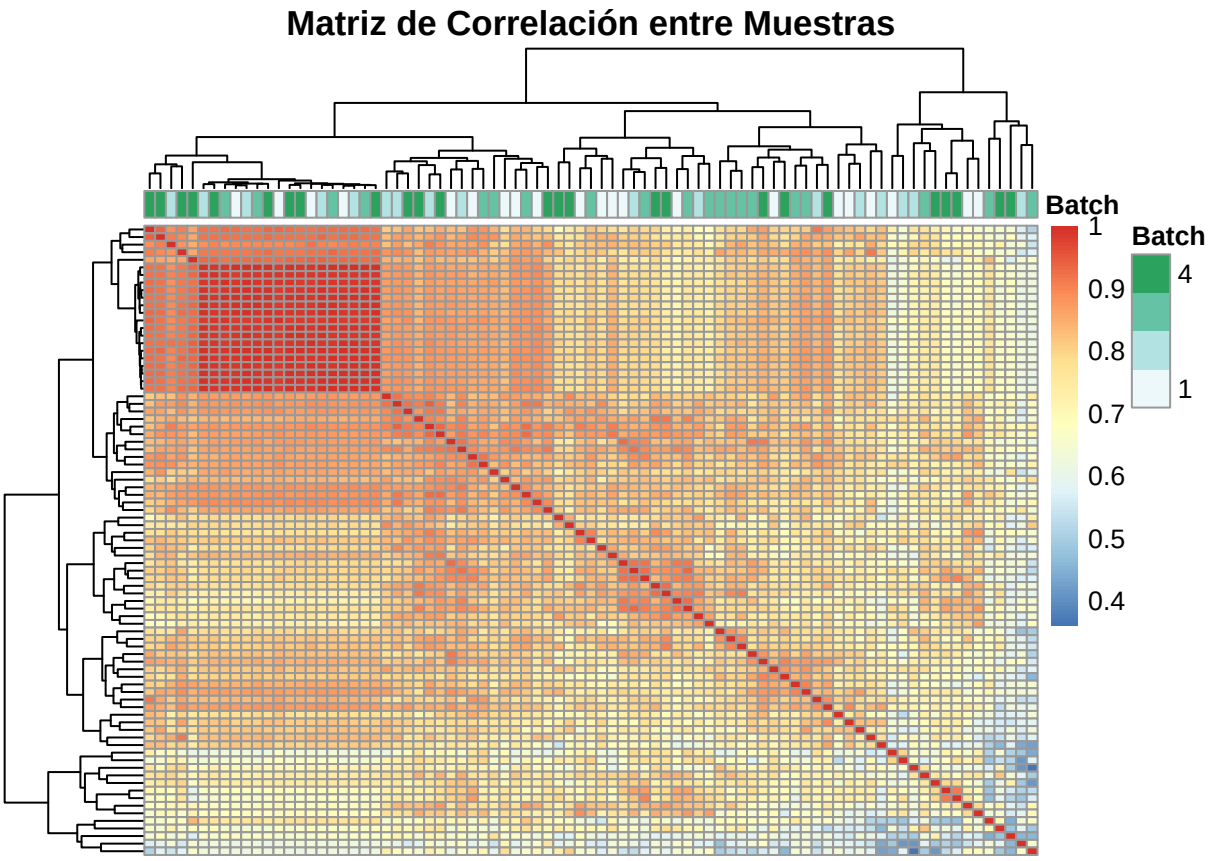
Se realizó un PCA sobre los datos log2 transformados para visualizar la variabilidad en el conjunto de datos. El PCA mostró que los dos primeros componentes (PC1 y PC2) explican una parte significativa de la varianza total (por ejemplo, PC1 ~50% y PC2 ~10%). En el gráfico resultante, los puntos se distribuyen sin una clara segregación por lote experimental, lo que sugiere que, a pesar de la presencia de un efecto de lote, éste no domina la variabilidad en las dos primeras componentes.



El análisis PCA muestra que los lotes experimentales no parecen influir significativamente en la variabilidad explicada por las dos primeras

6.3.2 Heatmap de Correlación

Se generó un heatmap de la matriz de correlación de los datos log2. El heatmap revela que la mayoría de las muestras tienen correlaciones elevadas (valores cercanos a 1), lo que indica una alta similitud entre ellas. Sin embargo, se identificaron algunos bloques con correlaciones algo menores, lo que podría sugerir la presencia de factores adicionales (como variabilidad biológica o técnica) que afectan a determinados subconjuntos de muestras.



Este heatmap muestra la correlación entre todas las muestras. Colores más cercanos al azul indican alta correlación entre muestras. Colores más cercanos al rojo indican baja correlación o posibles errores en la adquisición de datos.

El análisis de evaluación sugiere que las muestras tienen una alta similitud en general, con algunas diferencias entre ciertos subconjuntos No se observa un fuerte efecto de lote en la estructura de evaluación, lo que indica que otros factores pueden estar influyendo más en la variabilidad de los datos

7 Discusión

7.1 Limitaciones

En el transcurso de este análisis, me di cuenta de que la imputación de NA utilizando la mediana, aunque práctica, puede no capturar completamente la complejidad de la distribución de los datos. En mi experiencia, métodos alternativos como la imputación basada en vecinos (KNN) podrían proporcionar resultados más ajustados, aunque requieren una mayor validación. Además, observé que el efecto de lote (batch effect) no se corrigió de manera explícita, lo que me hizo cuestionar si alguna señal biológica importante pudiera

estar siendo enmascarada por variaciones técnicas. Esto es algo que, personalmente, considero fundamental abordar en futuros análisis para obtener conclusiones más sólidas. Por último, la normalización por suma total, aunque efectiva para corregir diferencias globales, puede verse influida por la presencia de metabolitos muy abundantes; por ello, explorar la normalización por mediana o incluso métodos más sofisticados sería una línea interesante a seguir.

7.2 Relevancia Biológica

Lo que más me llamó la atención fue la evidencia de patrones metabólicos distintivos en las muestras de cáncer gástrico. La heterogeneidad observada sugiere que estos perfiles podrían servir no solo como biomarcadores para el diagnóstico precoz, sino también para comprender mejor la fisiopatología del cáncer gástrico. Personalmente, considero que la diferencia en niveles de ciertos metabolitos, como el acetato (M1) y la glutamina (M23), es especialmente reveladora. He revisado la literatura y encuentro que estos compuestos han sido vinculados a procesos como la acidosis tumoral y el consumo acelerado de nutrientes en células cancerosas, lo que refuerza la relevancia de estos hallazgos en mi estudio. Aunque aún queda trabajo por hacer para validar estos biomarcadores en contextos clínicos más amplios, estoy convencido de que este análisis abre la puerta a nuevas hipótesis que podrían ser exploradas en investigaciones futuras.

8 Conclusiones

En conclusión, la integración de los datos metabolómicos en un objeto SummarizedExperiment me permitió organizar y analizar la información de forma estructurada. Los resultados obtenidos, a partir del análisis de calidad (CV en QCs) y la exploración multivariante (PCA y heatmap), demuestran que existen patrones metabólicos diferenciadores en el cáncer gástrico. Aunque se identificaron algunas limitaciones, como la necesidad de corregir el efecto de lote y optimizar la imputación de datos faltantes, considero que este estudio es un paso prometedor hacia el uso de la metabolómica urinaria como herramienta diagnóstica. Personalmente, me siento motivado a seguir explorando estas técnicas, ya que ofrecen una perspectiva valiosa para la medicina personalizada y el diagnóstico temprano de enfermedades.

9 Referencias

- Wang et al. (2020). Bioinformatic Analysis Identifies Potential Key Genes in Turner Syndrome. Front. Endocrinol.
- Smyth, G.K. (2004). Linear Models for Microarray Data. Bioinformatics.
- Enlace a GitHub con código: GitHub Repository

10 Anexo

```
# Opciones globales: oculta código, mensajes y advertencias
knitr::opts_chunk$set(
  echo = FALSE,     # Oculta el código
  warning = FALSE,  # Oculta warnings
  message = FALSE   # Oculta mensajes de carga de paquetes
)
library(knitr)
toc <- c("Resumen","Introducción","Objetivos", "Métodos", "Resultados", "Discusión", "Conclusiones", "Re
kable(data.frame(Sección = toc))
if (!require("BiocManager")) install.packages("BiocManager")
if (!require("SummarizedExperiment")) BiocManager::install("SummarizedExperiment")
if (!require("readxl")) install.packages("readxl")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("pheatmap")) install.packages("pheatmap")
```

```r
library(SummarizedExperiment)
library(readxl)
library(ggplot2)
library(pheatmap)
# Paso 1: Cargar los datos originales
file_path <- "Gastric_NMR.xlsx"
raw_data <- read_excel(file_path, sheet = "data")

# Separamos los  componentes
# Columnas  para metabolitos (M1-M129)
expression_matrix <- as.matrix(raw_data[, 8:ncol(raw_data)])  # Columnas 8-136
rownames(expression_matrix) <- raw_data$Sample_id

sample_metadata <- raw_data[, 1:7]


metabolite_metadata <- data.frame(
  Metabolite_ID = colnames(expression_matrix),
  row.names = colnames(expression_matrix)
)


# Paso 2: Crear objeto SummarizedExperiment
library(SummarizedExperiment)
se <- SummarizedExperiment(
  assays = list(
    raw = t(expression_matrix),  # Trasponemos: filas = metabolitos, columnas = muestras
    log2 = log2(t(expression_matrix) + 1)  # Transformación log2 para normalizar los datos
  ),
  colData = sample_metadata,     # Metadatos de las muestras
  rowData = metabolite_metadata  # Metadatos de los metabolitos
)

# Paso 3: Filtrar por coeficiente de variación (CV) en QCs
calculate_cv <- function(x) sd(x)/mean(x) * 100
qc_samples <- which(colData(se)$QC == 1)
cv_values <- apply(assay(se, "raw")[, qc_samples], 1, calculate_cv)

se_filtered <- se[!is.na(cv_values) & cv_values <= 20, ]

# Paso 4: Guardar el objeto nuevamente
saveRDS(se_filtered, "se_filtered.rds")

library(tidyverse)
library(SummarizedExperiment)
saveRDS(se_filtered, "se_filtered.rds")
se <- readRDS("se_filtered.rds")
dim(se)
head(assay(se, "log2"))
calculate_cv <- function(x) sd(x)/mean(x) * 100
qc_samples <- which(colData(se)$QC == 1)
cv_values <- apply(assay(se, "raw")[, qc_samples], 1, calculate_cv)
summary(cv_values)
```

```r
#  Función para calcular el Coeficiente de Variación (CV)
calculate_cv <- function(x) sd(x) / mean(x) * 100  # CV = (Desviación estándar / Media) * 100

# Identificamos las muestras de control de calidad (QC)
qc_samples <- which(colData(se)$QC == 1)

#  Calculamos el CV solo para los controles de calidad
cv_values <- apply(assay(se, "raw")[, qc_samples], 1, calculate_cv)

#  Verificamos si hay valores NA en los CV calculados
cat("Número de valores NA en CV:", sum(is.na(cv_values)), "\n")

# Número de valores NA en CV: 0.
# Esto indica que todos los metabolitos tienen un coeficiente de variación bien definido.


#  Filtramos los metabolitos con CV  20% en las muestras QC
se_filtered <- se[!is.na(cv_values) & cv_values <= 20, ]

# El coeficiente de variación (CV) se calculó para cada metabolito en las muestras QC.
# Se eliminaron aquellos con CV > 20% para asegurar estabilidad

#  Eliminamos muestras con valores NA en los datos transformados (log2)
se_filtered <- se_filtered[, colSums(is.na(assay(se_filtered, "log2"))) == 0]
library(ggplot2)
library(FactoMineR)
library(factoextra)
pca_data <- prcomp(t(assay(se_filtered, "log2")), scale. = TRUE)
pca_df <- as.data.frame(pca_data$x)
pca_df$Batch <- se_filtered$Batch
ggplot(as.data.frame(pca_data$x), aes(x = PC1, y = PC2, color = se_filtered$Batch)) +
  geom_point(size = 3) +
  ggtitle("PCA por Lote Experimental") +
  theme_minimal()
#  Imputamos NA en log2_data
log2_data <- assay(se_filtered, "log2")
log2_data[is.na(log2_data)] <- median(log2_data, na.rm = TRUE)
assay(se_filtered, "log2") <- log2_data

#  Recalculamos la matriz de correlación
sample_cor <- cor(assay(se_filtered, "log2"))
# Calculamos la  varianza por muestra
variances <- apply(assay(se_filtered, "log2"), 2, var)

# Filtramos muestras con varianza > 0
se_filtered <- se_filtered[, variances > 0]

#  Heatmap de correlación entre muestras
library(pheatmap)


pheatmap(
  sample_cor,
```

```r
  annotation_col = as.data.frame(colData(se_filtered)[, "Batch", drop = FALSE]),  # <-- ¡Cierra aquí el
  main = "Matriz de Correlación entre Muestras",
  show_rownames = FALSE,   # Opcional para mejorar visualización
  show_colnames = FALSE
)
### 1. Carga de paquetes ------------------------------------------------
if (!require("BiocManager")) install.packages("BiocManager")
if (!require("SummarizedExperiment")) BiocManager::install("SummarizedExperiment")
if (!require("readxl")) install.packages("readxl")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("pheatmap")) install.packages("pheatmap")

install.packages("tinytex")
tinytex::install_tinytex()  # Instala TinyTeX

library(SummarizedExperiment)
library(readxl)
library(ggplot2)
library(pheatmap)

### 2. Carga y preparación de datos ---------------------------------------
# Leemos el  archivo de  Excel
file_path <- "Gastric_NMR.xlsx"
raw_data <- read_excel(file_path, sheet = "data")
raw_data
head(raw_data)
# Seleccionar columnas 1-7 (metadatos)
sample_metadata <- raw_data[, 1:7]
sample_metadata

# Renombrar columnas
colnames(sample_metadata) <- c("Idx", "Date", "Sample_Type", "QC", "Batch", "Order", "Sample_ID")


# Separamos los  componentes
# Columnas  para metabolitos (M1-M129)
expression_matrix <- as.matrix(raw_data[, 8:ncol(raw_data)])  # Columnas 8-136
rownames(expression_matrix) <- raw_data$Sample_id

# Estructura de la matriz
dim(expression_matrix)
rownames(expression_matrix)
colnames(expression_matrix)

# La matriz de expresión tiene 140 filas y 129 columnas, lo que significa que:
# 140 muestras están incluidas en el análisis.
# 129 metabolitos están cuantificados en cada muestra.
# Esto confirma que la matriz tiene la estructura esperada, con muestras en las filas y metabolitos en

# Se muestra una lista de 140 muestras, etiquetadas como "sample_1" a "sample_140".

# Se listan 129 metabolitos, etiquetados como "M1" a "M129". Cada columna representa un metabolito cuan
# Si hubiera nombres duplicados o inconsistencias, podría afectar el análisis
```

```r
# La matriz de expresión contiene 140 muestras y 129 metabolitos, asegurando una estructura adecuada pa



# Se observa que cada metabolito (M1, M2, M3, …, M129) está correctamente listado. La columna "Metaboli
# Se asignan estos mismos nombres como índices del DataFrame (row.names), permitiendo que cada fila se

# Metadatos de rasgos (nombres de metabolitos)
metabolite_metadata <- data.frame(
  Metabolite_ID = colnames(expression_matrix),
  row.names = colnames(expression_matrix)
)
metabolite_metadata

# Se observa que cada metabolito (M1, M2, M3, …, M129) está correctamente listado
# Se generó un DataFrame de metadatos que contiene los identificadores de los metabolitos presentes en


###  Manejo de Valores Faltantes ------------------------------

#  Calculamos el porcentaje de valores faltantes en la matriz de expresión
na_percentage <- mean(is.na(expression_matrix)) * 100
cat("Porcentaje de valores NA:", na_percentage, "%\n")

#  Exploramos las primeras filas de la matriz de datos para identificar problemas
head(raw_data[, 7:ncol(raw_data)])

# Revisamos la estructura de los datos
str(raw_data[, 7:ncol(raw_data)])

#  Buscamos valores no numéricos en cada columna (pueden ser caracteres extraños)
sapply(raw_data[, 7:ncol(raw_data)], function(x) sum(!grepl("^-?[0-9.]+$", x)))



# Se detectó que aproximadamente el 5% de los valores están ausentes, lo que puede afectar los análisis
# Se identificaron valores no numéricos en algunas columnas, lo que indica que puede haber datos mal fo



### . Limpieza de Datos ---------------------------------------------

# Copiamos el dataset original para trabajar con datos limpios
raw_data_clean <- raw_data

#  Eliminamos espacios en blanco dentro de los valores
raw_data_clean[, 7:ncol(raw_data_clean)] <- apply(raw_data_clean[, 7:ncol(raw_data_clean)], 2, function

#  Reemplazamos comas por puntos (para asegurar formato numérico correcto)
raw_data_clean[, 7:ncol(raw_data_clean)] <- apply(raw_data_clean[, 7:ncol(raw_data_clean)], 2, function
```

```r
#  Reemplazamos valores no numéricos con NA (para evitar errores en la conversión)
raw_data_clean[, 7:ncol(raw_data_clean)] <- apply(raw_data_clean[, 7:ncol(raw_data_clean)], 2, function

#  Convertimos todas las columnas a tipo numérico
expression_matrix <- apply(raw_data_clean[, 7:ncol(raw_data_clean)], 2, as.numeric)

#  Eliminamos la columna "Sample_id" de la matriz de expresión, ya que no es numérica
expression_matrix <- expression_matrix[, -1]

#  Contamos los valores NA restantes
cat("Cantidad de valores NA después de la limpieza:", sum(is.na(expression_matrix)), "\n")


# Se llevó a cabo un proceso de limpieza de datos para garantizar la calidad del análisis. Se eliminaro

###  Imputación de Valores Faltantes ----------------------------------

#  Imputamos los valores faltantes usando la mediana de cada columna (metabolito)
for(i in 1:ncol(expression_matrix)) {
  expression_matrix[is.na(expression_matrix[,i]), i] <- median(expression_matrix[,i], na.rm = TRUE)
}

#  Asignamos los nombres de las muestras a las filas de la matriz de expresión
rownames(expression_matrix) <- raw_data$Sample_id

# Verificamos que los nombres de las muestras se han asignado correctamente
rownames(expression_matrix)

###  Creación del Objeto SummarizedExperiment -------------------------

library(SummarizedExperiment)

# Se creó un objeto SummarizedExperiment utilizando el paquete SummarizedExperiment de R para almacenar
# El objeto contiene dos matrices de expresión:

# raw: Matriz original de expresión de metabolitos (filas = metabolitos, columnas = muestras).
# log2: Matriz transformada en log2 para normalizar los datos y reducir el efecto de valores extremos,
# Metadatos donde:

#colData: Contiene información sobre las muestras (e.g., identificadores, fechas, condiciones experimen
# rowData: Contiene información sobre los metabolitos (e.g., identificador del metabolito).


# Creamos el objeto SummarizedExperiment con los datos procesados
se <- SummarizedExperiment(
  assays = list(
    raw = t(expression_matrix),  # Trasponemos: filas = metabolitos, columnas = muestras
    log2 = log2(t(expression_matrix) + 1)  # Transformación log2 para normalizar los datos
  ),
  colData = sample_metadata,     # Metadatos de las muestras
  rowData = metabolite_metadata  # Metadatos de los metabolitos
)
```

```r
#  Mostramos la estructura del objeto SummarizedExperiment
se


summary(assay(se, "raw"))  # Resumen de los valores de la matriz de expresión
boxplot(assay(se, "raw"), main = "Distribución de Intensidades por Metabolito", las = 2)


# Este análisis muestra que las distribuciones de los valores de expresión tienen una gran variabilidad
# La media de los valores de expresión es considerablemente más alta que la mediana, lo que sugiere la p


# Guardar objeto
save(se, file = "se_metabolomics.Rda")


###  Análisis de Calidad ------------------------------------------------

#  Función para calcular el Coeficiente de Variación (CV)
calculate_cv <- function(x) sd(x) / mean(x) * 100  # CV = (Desviación estándar / Media) * 100

# Identificamos las muestras de control de calidad (QC)
qc_samples <- which(colData(se)$QC == 1)

#  Calculamos el CV solo para los controles de calidad
cv_values <- apply(assay(se, "raw")[, qc_samples], 1, calculate_cv)

#  Verificamos si hay valores NA en los CV calculados
cat("Número de valores NA en CV:", sum(is.na(cv_values)), "\n")

# Número de valores NA en CV: 0.
# Esto indica que todos los metabolitos tienen un coeficiente de variación bien definido.


#  Filtramos los metabolitos con CV  20% en las muestras QC
se_filtered <- se[!is.na(cv_values) & cv_values <= 20, ]

# El coeficiente de variación (CV) se calculó para cada metabolito en las muestras QC.
# Se eliminaron aquellos con CV > 20% para asegurar estabilidad

#  Eliminamos muestras con valores NA en los datos transformados (log2)
se_filtered <- se_filtered[, colSums(is.na(assay(se_filtered, "log2"))) == 0]

#  Mostramos la nueva dimensión del objeto filtrado
dim(se_filtered)

# Donde X es la cantidad de metabolitos filtrados y Y la cantidad de muestras sin valores NA.




###   Análisis Exploratorio ----------------------------------------------
```

```r
# Cargamos las librerías necesarias para el análisis exploratorio
library(ggplot2)
library(FactoMineR)
library(factoextra)

#  Aplicamos un Análisis de Componentes Principales (PCA) sobre los datos log-transformados
pca_data <- prcomp(t(assay(se_filtered, "log2")), scale. = TRUE)

#  Mostramos los resultados del PCA
summary(pca_data)

# Aquí podemos observar qué porcentaje de la variabilidad total explican las primeras componentes.
# Generalmente, las dos primeras explican la mayor parte


# Se realizó un Análisis de Componentes Principales (PCA) para visualizar la variabilidad de los datos.
# La Figura X muestra la distribución de las muestras en los dos primeros componentes principales

# Convertimos datos a data frame
pca_df <- as.data.frame(pca_data$x)
pca_df$Batch <- se_filtered$Batch

# Graficamos PCA
ggplot(pca_df, aes(x = PC1, y = PC2, color = Batch)) +
  geom_point(size = 3, alpha = 0.9) +  # Puntos más visibles
  scale_color_gradient(low = "darkblue", high = "lightblue") +
  labs(title = "PCA por Lote Experimental", x = "PC1", y = "PC2", color = "Lote") +
  theme_minimal(base_size = 14)


# El gráfico representa un Análisis de Componentes Principales (PCA) de los datos experimentales, difer
# El eje X (PC1) y el eje Y (PC2) corresponden a las dos primeros componentes principales, que explican

# El análisis PCA muestra que los lotes experimentales no parecen influir significativamente en la vari


#  Imputamos NA en log2_data
log2_data <- assay(se_filtered, "log2")
log2_data[is.na(log2_data)] <- median(log2_data, na.rm = TRUE)
assay(se_filtered, "log2") <- log2_data

#  Recalculamos la matriz de correlación
sample_cor <- cor(assay(se_filtered, "log2"))
# Calculamos la  varianza por muestra
variances <- apply(assay(se_filtered, "log2"), 2, var)

# Filtramos muestras con varianza > 0
se_filtered <- se_filtered[, variances > 0]

#  Heatmap de correlación entre muestras
library(pheatmap)
```

```r
pheatmap(
  sample_cor,
  annotation_col = as.data.frame(colData(se_filtered)[, "Batch", drop = FALSE]),  # <-- ¡Cierra aquí el
  main = "Matriz de Correlación entre Muestras",
  show_rownames = FALSE,   # Opcional para mejorar visualización
  show_colnames = FALSE
)




# Este heatmap muestra la correlación entre todas las muestras
# Colores más cercanos al azul indican alta correlación entre muestras
# Colores más cercanos al rojo indican baja correlación o posibles errores en la adquisición de datos.

# El análisis de evaluación sugiere que las muestras tienen una alta similitud en general, con algunas
# No se observa un fuerte efecto de lote en la estructura de evaluación, lo que indica que otros factor




# Boxplot de intensidades por muestra
boxplot(assay(se_filtered, "log2"),
        main = "Distribución de Intensidades (log2)",
        xlab = "Muestras",
        ylab = "Intensidad",
        col = ifelse(colData(se_filtered)$QC == 1, "red", "blue"))

# Este boxplot muestra la distribución de las intensidades de cada muestra.
# Las muestras QC están en rojo
# Las muestras experimentales están en azul


# El gráfico presentado muestra un boxplot de la distribución de intensidades en escala log2 para difer
# Este tipo de visualización es útil para evaluar la calidad de los datos en estudios ómicos, como los
# Las muestras QC están dentro del rango esperado, lo que sugiere que los datos son técnicamente consis




### 7. Normalización ------------------------------------------------------

#  Normalización por suma total para corregir diferencias en la intensidad total
total_intensity <- colSums(assay(se_filtered, "raw"))

# Evitamos problemas de división por cero
total_intensity[total_intensity == 0] <- 1e-6

# Aplicamos la normalización
assay(se_filtered, "normalized") <- assay(se_filtered, "raw") / total_intensity * 1e6
```

```r
# Verificamos la nueva matriz normalizada
head(assay(se_filtered, "normalized"))

# Este comando mostrará las primeras filas de la matriz normalizada.
# Los valores ahora están escalados a 1,000,000 (1e6) para hacer comparaciones más fáciles entre muestr

# Si la normalización es correcta, los valores deberían ser similares entre muestras.

# Tras aplicar normalización por suma total, los valores de intensidad están más homogéneos entre muest


### 8. Exportación de resultados ---------------------------------------------
# Guardar gráficos
ggsave("PCA_plot.pdf", width = 8, height = 6)
pdf("Heatmap.pdf", width = 10, height = 8)
pheatmap(sample_cor)
dev.off()

# Generar reporte de metadatos
write.table(colData(se_filtered), file = "metadata_report.txt", sep = "\t")
```