Análisis de datos ómicos - PEC1

Erise Pérez Pascual

2025-03-26

Contents

Abstract

El trabajo presenta un análisis sobre datos de metabólomica sobre un conjunto de datos de caquexia humana. Este análisis se realiza primeramente creando un objeto de clase `SummarizedExperiment`, ideal para el manejo de este tipo de datos, y posteriormente mediante un análisis exploratorio de los datos. El análisis exploratorio incluye técnicas de análisis univariante y multivariante como el análisis de componentes principales y el agrupamiento jerárquico. El análisis multivariante informa sobre las tendencias de los datos, que en este caso no han aportado mucha información diferencial sobre los dos grupos esudiados (caquexia vs control).

Objetivos

Este trabajo tiene como objetivo analizar un conjunto de datos de metabolómica para aprender a utilizar algunas de las herramientas básicas para el análisis de datos ómicos. Con ello, nos familiarizaremos con los conjuntos de datos de esta ciencia ómica y con los objetos de clase `SummarizedExperiment` para su análisis y manejo. Además, el trabajo incluye la realización de un análisis exploratorio de los datos, permitiendo

que profundicemos en las técnicas principales tanto del análisis univariante como multivariante de lo que compone uno de los primeros pasos en el proceso de análisis de datos ómicos. Por tanto, el objetivo del trabajo es múltiple, desde profundizar en el manejo de herramientas vistas en el curso, hasta la redacción del propio informe para sacar conclusiones de los datos dándoles una interpretación biológica.

Métodos

Para la realización del trabajo se seleccionó y descargó el dataset de metabolómica de *human-cachexia* en formato .csv accesible desde el siguiente enlace web: https://rest.xialab.ca/api/download/metaboanalyst/human_cachexia.csv. Este dataset contiene muestras no apareadas, todos los valores son numéricos y no contiene valores perdidos. Contiene 77 muestras de orina de las cuales 47 pertenecen a pacientes con caquexia y 30 a pacientes control.

Los datos del dataset original se organizan en unas filas representando los pacientes y columnas representando las variables medidas, de las cuales una es nominal y el resto son numéricas. El dataset original posteriormente será procesado para convertirlo en un objeto de clase `SummarizedExperiment`, función contenida dentro del paquete del mismo nombre y que actúa como una extensión de la conocida clase `ExpressionSet` [1, 2]. Cuando los datos son convertidos a este tipo de objeto se trabajará únicamente con las variables numéricas y los ejes se transponen. De esta manera, las muestras pasan a ocupar las columnas, mientras que las variables, denominadas *features* en el análisis de datos ómicos, pasan a ocupar las filas. Las muestras se categorizan según la variable nominal del dataset original que indicaba la condición de las muestras. Se analizará este nuevo objeto y los datos generados se guardarán en formato texto(.txt) para poder realizar un análisis exploratorio con ellos, mientras que el objeto de clase `SummarizedExperiment` se guardará en formato binario (.Rda).

Posteriormente, los datos con los que se construye el `SummarizedExperiment` son sometidos a un análisis exploratorio, utilizando los paquetes y las funciones básicas de R. Primero, las muestras se someten a un análisis descriptivo básico mediante las funciones `str()` y `summary()` o mediante algunos gráficos que permiten el análisis univariante como los histogramas o los diagramas de cajas. Posteriormente, se realiza un análisis exploratorio multivariante que se centra en dos análisis: el análisis de componentes principales (PCA) y el agrupamiento jerárquico. Ambos se realizan utilizando funciones báse de R. Para el PCA se utiliza la función `prcomp()` y se representan los dos primeros componentes, mientras que para el análisis jerárquico se utiliza la función `hclust()`.

Finalmente, se analizan los metabolitos diferencialmente expresados mediante un test t que permite determinar la expresión diferencial entre dos condiciones. Al ser unestudio de metabolómica, el objetivo del test t en este caso es determinar si las concentraciones o expresiones de los metabolitos entre las condiciones caquexia y control son diferentes, y en caso de serlo, cuáles son los metabolitos más interesantes a analizar. Para este análisis se utiliza la función `t.test()`.

Resultados

Para comenzar, se cargan los datos que se han descargado para poder trabajar con ellos. Este dataset contiene datos de metabolómica sobre la caquexia humana. La caquexia es un síndrome metabólico que se caracteriza por una pérdida de masa muscular inintencionada que puede ir acompañada o no de pérdida de grasa. Esta pérdida de masa muscular y grasa puede conllevar efectos fatales como un fallo multiorgánico o la reducción de la eficacia terapéutica contra el cáncer [3]. El dataset ha sido seleccionado debido al interés sobre la salud pública y las implicaciones clínicas que el entendimiento de los mecanismos metabólicos de esta enfermedad podría conllevar.

Para comenzar se cargan los datos a partir de un archivo .csv indicando que la primera fila y la primera columna serán los nombres para las columnas y filas, respectivamente.

```r
# Cargamos los datos
data <- read.csv("human_cachexia.csv", header=TRUE, row.names = 1)
```

Construcción del objeto de clase `SummarizedExperiment`

Para trabajar con el dataset se creará un objeto de clase `SummarizedExperiment` que contenga los datos y los metadatos y que nos permita trabajar más fácilmente con ellos. La clase `SummarizedExperiment` es una extensión de la clase `ExpressionSet` que tiene una ventaja principal para trabajar con experimentos basados en la secuenciación, como son los datos de metabolómica [1].

Una vez instalado y cargado el paquete de `SummarizedExperiment`, se crea un objeto de esa clase para los datos con los que se trabajará. Para ello hay que tener en cuenta la arquitectura de estas clases de objetos. Estos objetos tienen una arquitectura tipo matriz donde las filas representan las variables de interés o *features*, que en nuestro caso serían los parámetros que se miden en las muestras de orina (63 variables que en el dataset original se muestran en las columnas); y donde las columnas representan las muestras (77 muestras que en el dataset original son las filas). Se observa que el dataset original y el objeto `SummarizedExperiment` tienen la arquitectura transpuesta, por lo que se transpone el dataset original para obtener la matriz deseada.

A continuación, para construir el objeto observamos que la variable *Muscle.loss* es nominal y no aporta valores numéricos a la matriz, por lo que se elimina del dataframe transpuesto. Una vez tenemos únicamente los valores numéricos podemos construir la matriz de *counts*.

Finalmente, construimos el objeto `SummarizedExperiment`que contiene la matriz de *counts* y la información de las columnas (*muestras*) y el tipo de pérdida muscular que tienen esas muestras y la información de las filas (*features*).

```r
# Creamos un objeto de clase SummarizedExperiment con los datos de caquexia
# Cargamos el paquete que nos permite transponer el dataframe
library(data.table)

# Transpoenemos el dataframe del dataset original
datat <- transpose(data)

# Eliminamos la feature nominal
datat <- datat[-1,]

# Construimos la matriz de datos counts
counts <- data.matrix(datat)

# Construimos el objeto SummarizedExperiment
colData <- DataFrame(Type=data$Muscle.loss)
rowData <- DataFrame(Metabolites=colnames(data[,-1]))
mySumExp <- SummarizedExperiment(assays=SimpleList(counts=counts),
                                 colData=colData, rowData=rowData)
mySumExp
```

```
## class: SummarizedExperiment
## dim: 63 77
## metadata(0):
## assays(1): counts
## rownames(63): 2 3 ... 63 64
## rowData names(1): Metabolites
## colnames(77): V1 V2 ... V76 V77
## colData names(1): Type
```

Tenemos un objeto con 63 filas que corresponden a las *features* y 77 columnas que pertenecen a las muestras. La información de las dimensiones se puede acceder mediante el accesor `dim()`. La información experimental se puede acceder mediante el accesor `assays()` que muestra la matriz de *counts* que se ha creado anteriormente.

Por otra parte, se puede acceder a la información de las filas mediante el accesor `rowData()` que mostrará los nombres de los 63 metabolitos que se han medido en las muestras de orina del estudio. También se puede acceder a los datos de las columnas mediante el accesor `colData()` que indica a qué tipo de muestra pertenece cada medición de metabolitos (caquexia vs control).

Finalmente, el accesor `metadata()` permite acceder a los metadatos del estudio; sin embargo, para este estudio no están disponibles.

Análisis exploratorio del dataset

En el análisis exploratorio se analizarán los datos producidos por el procesamiento anterior, al generar el objeto de clase `SummarizedExperiment`, que se han guardado en formato texto (.txt). Como se ha comentado anteriormente, este dataset está compuesto por 77 columnas correspondientes a las muestras bajo las dos condiciones que se quieren estudiar: las primeras 47 muestras pertenecen a muestras de orina de pacientes con caquexia y las restantes 30 pertenecen a pacientes control. Por otra parte, el dataset contiene 63 filas que corresponden a las concentraciones de los metabolitos estudiados.

El análisis exploratorio comienza con un análisis univariante. Se puede analizar la estructura del conjunto de datos con la función `str()`, que muestra que todas las variables del dataset son numéricas; o un resumen descriptivo estadístico con la función `summary()` que muestra gran variabilidad en los rangos que toman los datos, sin una tendencia diferencial clara entre las muestras de caquexia y las muestras control. Las variables pueden ser analizadas una a una mediante un histograma, o pueden representarse todas juntas mediante un diagrama de cajas (Figura 1). Debido a la gran asimetría de los datos, el diagrama de cajas se muestra como un conjunto de puntos desplazados hacia la parte baja del eje Y, que indica la expresión. Esta asimetría se muestra tanto en muestras de caquexia como en muestras control.

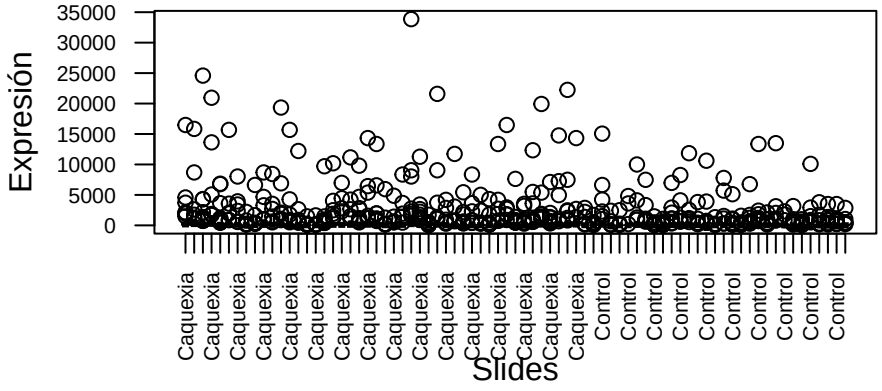

Figure 1: Diagrama de cajas

Esta asimetría se puede corregir utilizando una transformación logarítmica de los datos. Esta vez, el diagrama de cajas muestra todas las muestras pero sin mostrar ninguna tendencia diferencial clara entre las dos condiciones (Figura 2).

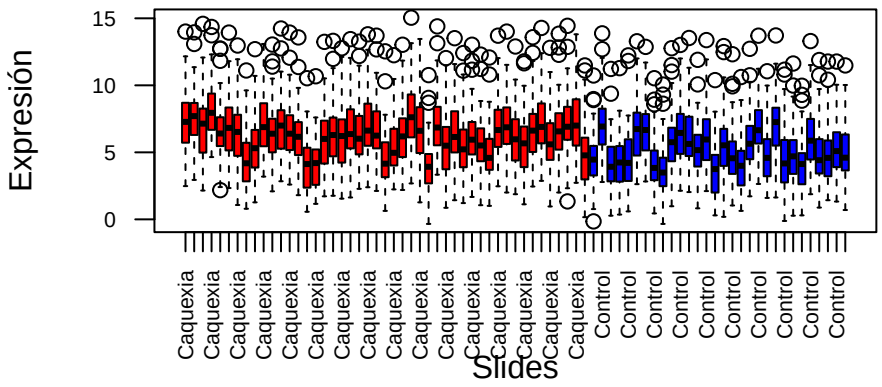

Figure 2: Diagrama de cajas de los datos transformados logarítmicamente

Posteriormente, se realiza un análisis multivariante. Este análisis se aborda de dos maneras: mediante un análisis de componentes principales y mediante un agrupamiento jerárquico. El PCA se realiza con la función `prcomp()` sobre los datos transformados, se computan las cargas o *loads* y se representan las dos primeras componentes (Figura 3). Los puntos se distribuyen en base a la variabilidad debida a estas dos componentes y se representan en colores según el grupo (rojo para la caquexia y azul para el control). Tal y como muestra la información de los ejes, la primera componente es la que más influye en la variabilidad, teniendo una carga de 57.4%, frente a la carga de 4.9% de la segunda componente. El efecto de esta carga se ve sobre los datos que parecen seguir levemente una tendencia más afectada por la primera componente, de manera que las muestras control (azul) se sitúan mayoritariamente hacia la izquierda, mientras que las muestras de caquexia (rojo) se sitúan hacia la derecha.

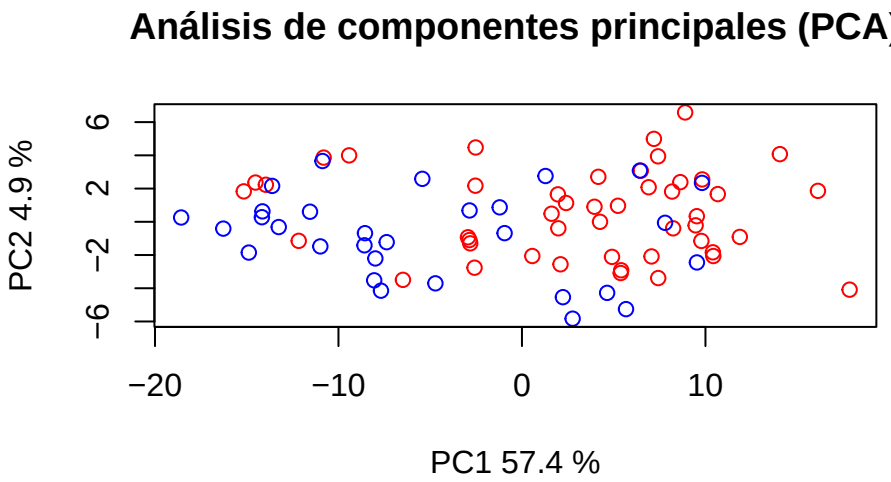

Figure 3: Análisis PCA

Respecto al agrupamiento jerárquico, este se realiza utilizando la función `hclust()`. El resultado es un dendrograma que representa las agrupaciones de cada muestra (Figura 4). Esta representación no muestra tendencias claras entre los dos tipos de muestras.

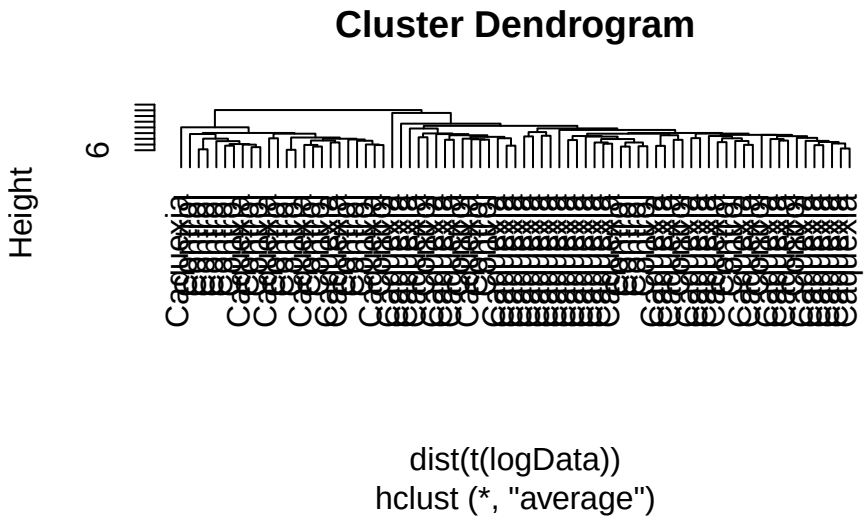

Figure 4: Dendrograma del agrupamiento jerárquico

Finalmente, se aplica un test t mediante la función `t.test()` para determinar si existe una expresión diferencial de los metabolitos entre las dos condiciones estudiadas. Mediante el estudio del p-valor para cada metabolito, se puede determinar cuáles de los metabolitos tienen un efecto más significativo sobre las muestras.

```
# Se genera una función que aplique el test t y devuelva los pvalores
ttest <- function(x){tt=t.test(x[1:47],x[48:77])
return(tt$p.value)}

# Se aplica la función a los datos y se extraen los pvalores
resp <- apply(logData,1,ttest)
pvals <- resp

# Utilizando los pvalores detectamos cuantos metabolitos son de interés
for (i in 0.05)
  print(paste("Metabolitos de interés:", length(which(pvals<i))))
```

```
## [1] "Metabolitos de interés: 54"
```

```
# Se puede restringir más la búsqueda
for (i in 0.0001)
  print(paste("Metabolitos de interés (restricción):", length(which(pvals<i))))
```

```
## [1] "Metabolitos de interés (restricción): 11"
```

Como resultado de este test, se puede observar que 54 metabolitos tienen un efecto estadísticamente significativo sobre las muestras.

Discusión

Para comenzar, los datos de metabolómica han sido tratados creando un objeto de clase `SummarizedExperiment` en lugar del clásico `ExpressionSet`. Esto se debe a que `SummarizedExperiment` es más flexible con la información de las filas [3], lo que le concede cierta ventaja respecto a `ExpressionSet`, además de ser utilizado por muchos repositorios de datos de metabolómica. Por otra parte, al igual que `ExpressionSet`, `SummarizedExperiment` permite la coordinación entre los metadatos y los *assays* al hacer el *subsetting*.

Por otra parte, cabe destacar que al estar trabajando con datos de metabolómica surge una principal diferencia con los estudios clásicos de genómica o transcriptómica y es que, mientras que en estos últimos las *features* suelen consistir en largas filas de genes o transcritos, en el estudio realizado se tiene una lista de los metabolitos estudiadosen las muestras. Por ello, será la concentración de estos metabolitosla que se estudie en el análisis exploratorio y la que determinará las diferencias entre los grupos de estudio en lugar de los genes diferencialmente expresados.

En este trabajo, el análisis exploratorio, y más concretamente el análisis multivariante, se ha hecho desde un enfoque ómico utilizando el PCA y el agrupamiento jerárquico en lugar de técnicas clásicas como los análisis de regresión. El PCA realizado en este caso ha utilizado los dos primeros componentes, que son los componentes que explican la mayor fuente de variabilidad. Estas representaciones permiten visualizar tendencias de agrupación de los datos, tanto las naturales como las debidas al efecto *batch*. De igual manera, las representaciones del dendrograma de la agrupación jerárquica pueden mostrar tendencias de agrupación de los datos. En el caso de estudio, la PCA ha mostrado una variabilidad principalmente debida la primera componente, que es la de mayor carga, mientras que la agrupación jerárquica no ha mostrado ninguna tendencia interesante entrelos dos grupos. Esta falta de tendencia de agrupación se ha mostrado asimismo en el análisis univariante. Concretamente, el diagrama de cajas no mostraba que los datos siguieran una tendencia definida dependiendo de las muestras a las que pertenecían. Sin embargo, cabe mencionar que el test t ha determinado que existen características diferenciales entre los dos grupos y es capaz de identificar una serie de metabolitos de interés que podrían propiciar esas diferencias entre los grupos.

Desde un punto de vista biológico, la ausencia de una tendencia clara dificulta el estudio ya que en un principio se esperaría que hubiera claras diferencias entre las muestras de pacientes con caquexia y pacientes control. La presencia de metabolitos que pudieran estar diferencialmente expresados abre la posibilidad de seguir estudiando dichos metabolitos ya que podrían ser indicadores de la enfermedad, lo cual tiene un alto interés médico a nivel molecular.

Dicho esto, el estudio presenta algunas limitaciones en lo referido a las conclusiones que se pueden sacar de él. Sería conveniente acompañar el estudio de análisis clásicos estdísticos como modelos de regresión que podrían ser llevados a cabo debido al bajo número de variables que contiene en este caso. Asimismo, es posible que un mayor número de muestras pudiera aportar más luz a los resultados y permitiría detectar más tendencias en los datos.

Conclusiones

Como conclusión, el estudio ha permitido la profundiación en el proceso de análisis de datos ómicos permitiendo la familiarización con los objetos `SummarizedExperiment` y el posterior análisis exploratorio de los datos que es principal enlas primeras etapas de estos procesos. Sin embargo, las implicaciones biológicas del análisis son tempranas y no se puede concluir que los datos sigan ninguna tendencia que diferencia a los dos grupos de estudio.

Referencias

1. *SummarizedExperiment* for Coordinating Experimental Assays, Samples, and Regions of Interest

2. SummarizedExperiment-class: SummarizedExperiment objects

3. Dodson, S., Baracos, V. E., Jatoi, A., Evans, W. J., Cella, D., Dalton, J. T., & Steiner, M. S. (2011). Muscle wasting in cancer cachexia: clinical implications, diagnosis, and emerging treatment strategies. Annual review of medicine, 62, 265–279. https://doi.org/10.1146/annurev-med-061509-131248

Dirección del repositorio GitHub

https://github.com/erisep13/Perez-Pascual-Erise-PEC1.git