PEC1 Anàlisi de Dades Òmiques

Eulalia Gispert Serrano

2024-10-29

Tabla de Contenidos

Abstract

Durante esta práctica procederemos a la creación y exploración de un conjunto de datos y archivos relacionados con la metabolómica. El objetivo será la construcción de un objeto tipo Summarized Experiment mediante el programa R Studio. Dentro de los datos tenemos los siguientes archivos, “DataInfo_S013.csv) que contiene iformación de cada columna del archivo”DataValues_S013.csv” que es un archivo que contiene valores de diferentes variables relacionadas con la metabolómica. Además contamos también con el archivo “description.md” que nos ofrece información relativa a los otros dos archivos y con el que trabajaremos como nuestros metadatos dentro de nuetsro objeto sumexp.

Se construye este objeto, de ahora en adelante “sumexp” donde podremos encontrar, de forma estructurada, nuestros datos ómicos y clínicos junto con los respectivos metadados. Mediante la construcción de este objeto, veremos como se evalua la calidad de nuestros datos, su estructuración y se detectan los posiblñes valores faltantes. Además se incluye finalmente un breve Análisi de Componentes Principales (PCA).

Los resultados de nuestra PEC nos mostrarán que trabajar con objectos de tipo Summarized Experiment nos facilita la organización y manipulación de nuestros datos. Se permite el mantenimiento de estos de forma integrada y nos facilita la interpretación entre variables.i

Objetivos

El objetivo de esta práctica es la creación de un objeto del tipo Summarized experiment y su exploración mediante el programa R. Así, conseguimos profundizar en los conceptos tocados durante esta primera PEC, poniéndolos a prueba.

Ademas, también destacaría como objetivo aprendizaje de la plataforma GitHub, su funcionamiento y estructuración, trabajando con datos provinientes de la plataforma y incluso creando nuestro propio repositorio.

Materiales y Métodos

Para llevar a cabo esta primera PEC hemos usado el programa R Studio, versión 4.4.1 y la plataforma GitHub, así como el programa Git, versión 2.47.0.windows.1.

Dentro del programa R Studio, hemos necesitado los siguientes paquetes y liberías:

- BiocManager

    -SummarizedExperiment

-readxl

-dplyr

-plotly

Los datos con los que hemos trabajado provienen del repositorio de Git Hub https://github.com/nutrimetabolomics/metaboData/tree/main . En concreto en nuestro caso hemos usado del Dataset “2018-MetabotypingPaper.csv”. Este dataset contiene dos archivos csv que hemos podido visualizar previamente mediante Excel. Respecto al procedimiento seguido, este estará detallado durante el desarrollo de la PEC.

Desarrollo del Informe

Instalación y carga de paquetes necesarios

Para empezar a realizar nuestro Summarized Experiment, vamos a tener que descargar la librería SummarizedExperiment. Debido a que forma parte de BiocManager, vamos a tener que instalarla a través de este.

También vamos a descargar el paquete dplyr que nos facilitará el uso y manipulación de los datos organizándolos por ejemplos con el formato data frame.

```
options(repos = c(CRAN = "https://cran.rstudio.com/"))

if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("SummarizedExperiment")

install.packages("dplyr")
```

Carga de Datos

Establecemos nuestro directorio de trabajo y realizamos una lectura y guardado de nuestros datos. Nuestros datos se encuentran en formato csv y no nos hace falta la descarga de ninguna librería en concreto, ya que la función read.csv() es parte del paquete base de R, igual que readLines().

```
# Establecer el directorio de trabajo
setwd("~/Omiques/metaboData/Datasets/2018-MetabotypingPaper")


# Cargar los archivos
data_info <- read.csv("DataInfo_S013.csv", stringsAsFactors = FALSE)
data_values <- read.csv("DataValues_S013.csv", stringsAsFactors = FALSE)
descripcion <- readLines("description.md")
```

Estructuración de los datos

En esta parte, estructuramos y manipulamos nuestros datos para quedarnos con las partes que realmente nos hacen falta. Eliminamos la columna X.1 del conjunto de datos “data_values”. A continuación asignamos la columna “SUBJECTS” como nombre de las filas i la eliminamos como columna en si. Finalmente convertimos “data_values” de data frame a matriz.

Un paso importante que se realiza aquí es la sustitución de los valores faltantes, NA, por la mediana de cada columna. Así podremos trabajar con el archivo, obviando las consecuencias de tener grandes cantidades de valores faltantes que no nos permiten aprofundir en el análisis.

```
# Arreglamos la estructura
data_values <- data_values %>% select(-X.1)
rownames(data_values) <- data_values$SUBJECTS
data_values <- data_values %>% select(-SUBJECTS)
data_matrix <- as.matrix(data_values)
col_data <- data_info %>% select(VarName, varTpe, Description)

# Aseguramos que las columnas de data_matrix coincidan con los nombres en
col_data
common_vars <- intersect(colnames(data_matrix), col_data$VarName)

# Filtramos data_matrix y col_data para mantener solo las variables
comunes
data_matrix <- data_matrix[, common_vars, drop = FALSE]
col_data <- col_data[col_data$VarName %in% common_vars, ]
```

Creación del Objeto SummarizedExperiment

Una vez tenemos estructurados correctamente nuestros datos procedimos a la creación de nuestro objeto Summarized Experiment. En nuestro caso estará formado por los datos relativos al archivo “DataValues_S013.csv” como componente assay, los datos del archivo “DataInfo_S013.csv” como componente col_data, que contiene información sobre los datos de cada columna y, finalmente, como metadata tendremos el archivos description.md que nos ofrece un breve resumen de los archivos que componen el Dataset y su estructura.

```
# Crear el objeto SummarizedExperiment
sumexp <- SummarizedExperiment(
    assays = SimpleList(counts = data_matrix),
    colData = col_data
)


# Comparamos la primera columna de data_matrix con la primera fila de
col_data para comprobar que nuestros datos cuadran.

first_match <- colnames(data_matrix)[1] == col_data$VarName[1]
last_match <- colnames(data_matrix)[ncol(data_matrix)] ==
col_data$VarName[nrow(col_data)]


# Agregamos nuestro archivo description como nuestra metadata
metadata(sumexp)$description <- paste(descripcion, collapse = "\n")
```

```r
# Revisar el objeto
print(sumexp)
```

```
## class: SummarizedExperiment
## dim: 39 694
## metadata(1): description
## assays(1): counts
## rownames(39): 1 2 ... 38 39
## rowData names(0):
## colnames(694): SURGERY AGE ... SM.C24.0_T5 SM.C24.1_T5
## colData names(3): VarName varTpe Description
```

```r
save(sumexp, file = "sumexp.Rda")
```

Exploración del Dataset

Ahora que tenemos creado nuestro SummarizedExperiment podemos explorarlo. Observamos que tenemos un data set de 39 filas y 694 columnas. Si quisiéramos trabajar con estos datos, por ejemplo para observar los valores medios de las variables numéricas podríamos proceder como se detalla a continuación. Convertimos nuestro conjunto a formato data frame. Convertimos aquellas columnas que por sus valores lo permiten a tipo numérico, nos aseguramos que nos quedamos solo con las columnas de tipo numéricos, y para poder trabajar y deshacernos de los valores faltantes lo que haremos será sustituir los NAs por el valor medio de esa columna.

```r
# Dimensiones del objeto SummarizedExperiment
dim(sumexp)
```

```
## [1]  39 694
```

```r
# Nombres de las filas y columnas para una visión general
head(rownames(sumexp))
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```r
head(colnames(sumexp))
```

```
## [1] "SURGERY"   "AGE"       "GENDER"    "Group"     "MEDDM_T0"
"MEDCOL_T0"
```

```r
#Exploramos diferentes componentes de nuestro Summarized Experiment
colData(sumexp)
```

```
## DataFrame with 694 rows and 3 columns
##                 VarName      varTpe Description
##             <character> <character> <character>
## SURGERY         SURGERY   character    dataDesc
## AGE                 AGE     integer    dataDesc
## GENDER           GENDER   character    dataDesc
```

```
## Group             Group     integer    dataDesc
## MEDDM_T0       MEDDM_T0     integer    dataDesc
## ...                 ...         ...         ...
## SM.C18.0_T5 SM.C18.0_T5     numeric    dataDesc
## SM.C18.1_T5 SM.C18.1_T5     numeric    dataDesc
## SM.C20.2_T5 SM.C20.2_T5     numeric    dataDesc
## SM.C24.0_T5 SM.C24.0_T5     numeric    dataDesc
## SM.C24.1_T5 SM.C24.1_T5     numeric    dataDesc
```

```
rowData(sumexp)
```

```
## DataFrame with 39 rows and 0 columns
```

```
assayNames(sumexp)
```

```
## [1] "counts"
```

```
summary(colData(sumexp))
```

```
## [1] "DataFrame object of length 3 with 0 metadata columns"
```

```
# Convertir la matriz assay(sumexp) a un data frame temporal
assay_df <- as.data.frame(assay(sumexp))

# Intentar convertir cada columna de assay_df a numérica
assay_df_numeric <- as.data.frame(lapply(assay_df, function(x)
as.numeric(as.character(x))))
```

```
## Warning in FUN(X[[i]], ...): NAs introducidos por coerción
## Warning in FUN(X[[i]], ...): NAs introducidos por coerción
```

```
# Filtrar solo las columnas numéricas
numeric_data <- assay_df_numeric[, sapply(assay_df_numeric, function(x)
!all(is.na(x)))]

# Calcular la media de las columnas numéricas, ignorando NA
meancol <- apply(numeric_data, 2, mean, na.rm = TRUE)

# Mostrar los primeros resultados
head(meancol)
```

```
##         AGE       Group    MEDDM_T0   MEDCOL_T0   MEDINF_T0
MEDHTA_T0
## 40.79487179  1.38461538  0.00000000  0.02631579  0.13157895
0.23684211
```

Análisi de Componentes Principales

```
# Filtramos las columnas con varianza mayor que cero
numeric_data_nonconstant <- numeric_data[, apply(numeric_data, 2, var,
na.rm = TRUE) != 0]
```

```r
# Escalamos los datos y reemplazamos los valores NA por la mediana de esa columna
numeric_data_scaled <- scale(numeric_data_nonconstant, center = TRUE, scale = TRUE)
numeric_data_scaled[is.na(numeric_data_scaled)] <- apply(numeric_data_scaled, 2, function(x) median(x, na.rm = TRUE))

## Warning in numeric_data_scaled[is.na(numeric_data_scaled)] <-
## apply(numeric_data_scaled, : número de elementos para sustituir no es un
## múltiplo de la longitud del reemplazo

# Realizamos el PCA con los datos ya limpios y escalados
pca_result <- prcomp(numeric_data_scaled, center = TRUE, scale. = TRUE)

# Extraemos las tres primeras componentes
pca_scores <- pca_result$x[, 1:3]  # Accede a las tres primeras columnas de las puntuaciones
head(pca_scores)

##             PC1       PC2        PC3
## [1,] -6.705906 -1.009610  1.6525825
## [2,] -3.277563 -2.916287  4.2700827
## [3,] -9.008891 -0.143118 -0.4475671
## [4,] -1.150242 -3.058083  4.4892309
## [5,] -6.543450  1.907986  0.6842150
## [6,]  1.362066 -7.289844 15.2188359
```

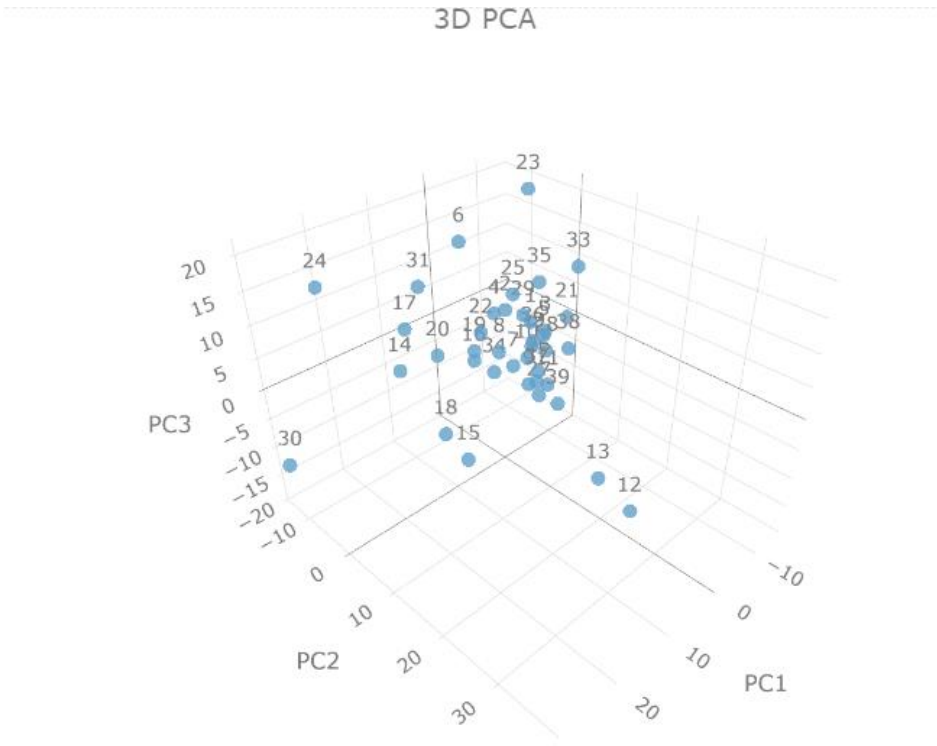
Con esta exploración podemos observar que hemos creado un Summarized Experiment de 39 filas y 694 columnas.

Con rowData() podemos observar que nos devuelve 39 filas y 0 columnas. Esto se debe a que para las filas no existen metadatos específicos ni descripciones. En cambio, con colData() observamos qu 694 filas y 3 columnas. Esto se debe a que tenemos metadatos asociados a las columnas, es decir, datos que describen las características de cada columna de nuestro experimento.

Finalmente observamos que assayNames() nos devuelve “counts”. Esto será así ya que tenemos un solo ensayo que contiene valores cuantitativos principalmente.

Si tuviéramos metadatos relacionados con las filas podríamos añadirlos como rowData de forma que nuestro Summarized Experiment contuviera también la información relativa a cada elemento fila, en nuestro caso, seguramente paciente.

Después de realizar de forma muy breve nuestro pequeño PCA, y elaborando un gráfico 3D con las 3 primeras componentes obtenemos lo siguiente:



A partir de un PCA más exhaustivo y elaborado se podrían identificar los patrones de agrupación que se forman, como ya podemos observar en la imagen adjunta (gráfico 3D). Si los pacientes se agruparan de acuerdo a perfiles de metabolitos estos nos podrían estar indicando que estarían relacionados los biomarcadores con diferentes condiciones de salud o respuestas a tratamiento.

Creación de nuestro Repositorio en GitHub

Finalmente, creamos nuestro repositorio en GitHub que contenga todos los archivos requeridos en esta PEC1. En mi caso, lo he creado de forma manual y de la misma forma he añadido los archivos. De cara en adelante, tengo la intención de aprender a realizarlo bien desde R para un mejor mantenimiento de los datos.

La dirección de mi repositorio es la siguiente:
https://github.com/LaiaGispert/Gispert-Serrano-Eulalia-PEC1