PEC1

Enrique Pouso-Vázquez Sánchez

2025-03-31

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

## Warning: package 'matrixStats' was built under R version 4.3.3

##   
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':  
##   
## colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,  
## colCounts, colCummaxs, colCummins, colCumprods, colCumsums,  
## colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,  
## colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,  
## colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,  
## colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,  
## colWeightedMeans, colWeightedMedians, colWeightedSds,  
## colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,  
## rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,  
## rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,  
## rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,  
## rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,  
## rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,  
## rowWeightedMads, rowWeightedMeans, rowWeightedMedians,  
## rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':  
##   
## IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, aperm, append, as.data.frame, basename, cbind,  
## colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,  
## get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,  
## match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,  
## Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,  
## table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 4.3.2

##   
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':  
##   
## findMatches

## The following objects are masked from 'package:base':  
##   
## expand.grid, I, unname

## Loading required package: IRanges

##   
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':  
##   
## windows

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.3.3

## Loading required package: Biobase

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

##   
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':  
##   
## rowMedians

## The following objects are masked from 'package:matrixStats':  
##   
## anyMissing, rowMedians

*Tabla de contenido:*

* Abstract
* Objetivos
* Métodos
* Resultados
* Discusión
* Conclusiones
* Referéncias

*Abstract:*

*Objetivos:*

1. Aprender a buscar y descargar un conjunto de datos y metadatos de un estudio real de metabolómica en la base de datos metabolomicsWorkbench.
2. Aprender a generar un objeto de clase SummarizedExperiment del paquete Bioconductor (Assay, colData, rowData y metadata) a partir de la información de un estudio real de metabolómica.
3. Realizar un estudio exploratorio de los datos contenidos en el SummarizedExperiment.
4. Analizar e interpretar los resultados obtenidos del estudio exploratorio y generar un informe ‘Rmarkdown’ con los resultados.
5. Aprender a utilizar las herramientas que proporciona Git en RStudio y familiarizarse con la plataforma de GitHub.

*Métodos:*

Los datos utilizados en este informe se han obtenido a partir de la web metabolomicsWorkbench, que funciona como plataforma de recursos, repositorio y herramienta de análisis de datos metabolómicos. El dataset utilizado en este informe proviene de un estudio titulado Metabolomics Analysis: Opioid Addiction Project de la universidad de North Carolina at Chapel Hill. El estudio contiene información de 2811 metabolitos detectados en orina de 298 pacientes. Los pacientes se dividen en 2 categorias “User” y “Non-User” segun si son consumidores habituales de opioides. Cabe destacar que existe una tercera categoria llamada “study\_pools”, que contiene 38 observaciones y son mezclas de todas las muestras individuales, su finalidad es ser un control de calidad. El objetivo (Pregunta biológica) del estudio es identificar como los opioides impactan en las rutas metabólicas, determinar biomarcadores de abuso e identificar metabolitos importantes en la diferenciación de los fenotipos del estudio. Se ha decidido trabajar con este dataset porque el tema es de interés para el autor de este informe y el conjunto de datos contiene la información estandar que se podría obtener en cualquier estudio de metabolómica.

Los datos se encuentran en un archivo .txt que contiene las intensidades de todos los metabolitos (obtenidas a partir de LC-MS) por cada muestra obtenida. Ademas existen otros archivos .txt con los metadatos del estudio. Para poder trabajar el dataset es conveniente crear un SummarizedExperiment, que es una extensión del Expressionset creado en un principio para tratar datos de microarrays. Ambos son estructuras del paquete Bioconductor y se diferencian en su estructura y flexibilidad, aunque la mayor diferencia es que el SummarizedExperiment permite manejar múltiples matrices de datos. El SummarizedExperiment contiene assay (matriz de datos principal), colData (Información de las observaciones), rowData (Información de las variables) y metadata (Información sobre el experimento). El objeto SummarizedExperiment funciona como un todo, de forma que si se modifica algún campo, automaticamente este es modificado en todos los apartados.

La herramienta estadística utilizadas para llevar a cabo tanto la creación del SummarizedExperiment como el análisis exploratorio de los datos ha sido RStudio y se ha utilizado Git como control de versiones.

El procedimiento de análisis de los datos ha incluido determinar la presentacia de missing values (NA) así como la realización de un analisis de componentes principales y un clustering jerarquico. Es importante aclarar que el análisis contenido en este informe se encuadra entre el preprocesado de datos y los análisis estadísticos dentro del diseño experimental de un estudio estándar de metabolómica.

*Resultados:*

* SummarizedExperiment

El archivo .txt con las intensidades de los metabólitos se ha cargado a RStudio mediante la función readtable(), especificando que la primera fila es el ‘header’ y que las separaciones de los campos son tabulaciones “.

datos <- read.table("ST001618\_AN002653\_Results.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)

El primer problema ha surgido en la interpretación de los datos por RStudio, el archivo original utilizaba puntos (.) para la separacion del millar (Ej: 542.345.122) y al subir el archivo se han añadido puntos de forma aleatoria en los datos sin un sentido lógico. Esto se ha resuelto eliminando todos los puntos del dataset mediante el código siguiente:

datos[-1] <- lapply(datos[-1], function(x) gsub("\\.", "", x))

La primera columna es un string que representa el identificador de las variables (metabólitos), por ese motivo se ha obviado al elimnar los puntos.

Para crear el SummarizedExperiment es necesario generar 4 subclases: Assay (matriz de datos principal), colData (Información de las observaciones), rowData (Información de las variables) y metadata (Información sobre el experimento). Una vez creadas se combinarán para crear el objeto SummarizedExperiment. Para crear el colData se ha utilizado la información contenida en la primera y segunda fila de la matriz de datos principal que corresponden al identificador de las muestras y al grupo que pertenecen (User, Non-User, Study\_Pools).

col\_data <- datos[1, , drop = F]  
col\_data <- as.data.frame(t(col\_data))  
colnames(col\_data) <- c("Grupo")  
col\_data$Grupo <- gsub("Phenotype:", "", col\_data$Grupo)

Es importante transponer el dataframe, ya que de forma habitual el colData contiene las observaciones en las filas y las variables en las columnas (De forma inversa al assay).

A continuación se creó la subclase rowData, que contiene la información de las variables. Este estudio no disponia de ninguna información relacionada con los metabolitos más alla de su identificaciñon (primera columna del dataset). Por este motivo, con el fin de aprender a hacer un SummarizedExperiment completo, se ha decidido crear el rowData unicamente con el ID de la variable y una columna que numera todas las variables.

row\_data <- datos[, 1, drop = F]  
row\_data <- as.data.frame(row\_data[-1,])  
row\_data$numeracion <- seq(from = 1, to = dim(row\_data)[1])

La siguiente subclase es metadata. Esta contiene una lista de strings que representa información relevante del estudio. En este caso contiene información sobre el centro donde se realizo el estudio; Nombre del estudio, nombre del investigador, institución, departamento y contacto, y también información de las muestras; Especie, rango de edad, género y tipo de muestra.

metadatos <- list(ExperimentName = "Metabolomics Analysis: Opioid Addiction Project (Golestan Cohort Study)", ResearcherName = "Susan Sumner", Institution = "University of North Carolina at Chapel Hill", Department = "Nutrition", Contact = "susan\_sumner@unc.edu", Species = "Homo Sapiens", AgeRange = "40-75", Gender = "Male & Female", SampleType = "Urine")

En este punto se procede a preparar el assay (maytriz principal de datos). Para ello se eliminan las filas que no son datos de intensidad y se modifican el tipo de datos de carácteres a numéricos. Este último paso es necesario para que los siguientes análisis (PCA y clustering), puedan identificar y procesar los datos numéricos.

datos <- datos[-1,]  
datos[,-1] <- as.data.frame(lapply(datos[,-1], as.numeric))

Finalmente, se utiliza la función SummarizedExperiment() para crear el objeto.

sumexp <- SummarizedExperiment(assays = list(Opioids = datos), rowData = row\_data, colData = col\_data, metadata = metadatos)

* Valores NA

Una vez se ha creado el objeto SummarizedExperiment se puede proceder a realizar el análisis exploratorio de los datos. En este caso el análisis contiene una busqueda de valores NA, un análisis de componentes principales y un clustering jerarquico.

Antes de realizar cualquier tipo de análisis estadístico es básico buscar si existe algun valor faltante en el conjunto del dataset.

sum(is.na(assay(sumexp)))

## [1] 0

En este caso no existe nigún valor NA, por lo que no es necesario hacer nada más. En caso contrario, se deberian eliminar los datos faltantes ya que pueden interferir en estudios posteriores.

* Análisis de componentes principales (PCA)

En estudios ómicos es muy habitual trabajar con analisis de reducción de la dimensinalidad porque, a diferencia de otros conjuntos de datos, se tienen muchas más variables que observaciones. El PCA permite reducir las dimensiones manteniendo la mayor cantidad posible de información contenida en los datos originales. Cada componente es una combinación lineal de todas las variables que explica un % de la variabilidad inicial de los datos. Si los dos primeros componenetes explican una alta proporción de la variabilidad, es posible generar un gráfico de dispersión de dos dimensiones y simplificar la visualización de los datos.

Para realizar el PCA es necesario que los datos estén escalados, ya que de lo contrario las variables con mayor escala tendrían más peso en el análisis. Para escalar las variables se resta la media y se divide por la desviación estándar.

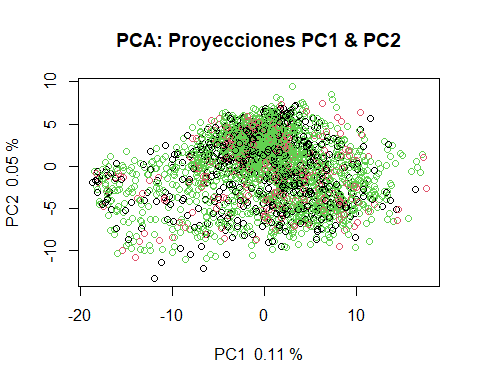
matriz <- assay(sumexp)  
matriz\_num <- matriz[, sapply(matriz, is.numeric)]  
pca <- prcomp(matriz\_num, scale = TRUE)  
summary(pca)$importance[, 1:6]

## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 6.031744 3.901646 2.725097 2.118374 1.977229 1.478306  
## Proportion of Variance 0.108280 0.045310 0.022100 0.013360 0.011640 0.006500  
## Cumulative Proportion 0.108280 0.153590 0.175690 0.189040 0.200680 0.207180

Se ha utilizado la función prcomp() para realizar el PCA porque es más eficiente en datos de alta dimensionalidad.

Se ha rezalido el gráfico de dispersión de los dos primeros componentes principales y se ha asignado un color a cada categoría de las observaciones.

var\_pc1 <- pca$sdev[1]^2/sum(pca$sdev^2)  
var\_pc2 <- pca$sdev[2]^2/sum(pca$sdev^2)  
  
col\_grupo <- colData(sumexp)[-1, "Grupo"]  
  
plot(pca$x[,1:2], col = as.factor(col\_grupo), main = "PCA: Proyecciones PC1 & PC2", xlab = paste("PC1 ", round(var\_pc1, 2),"%"), ylab = paste("PC2 ", round(var\_pc2,2), "%"))



* Clustering

En análisis se realizará un cluster jerárquico basado en distancias euclídeas y enlaces promedio (average linkage). Este método es un enfoque intermedio entre el “single” y el “complete”, que es útil cuando se busca un balance en la formación de los clusters.

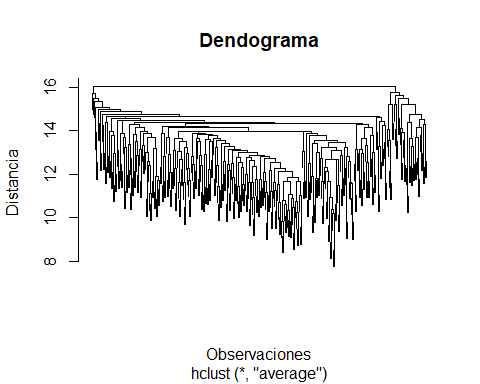
matriz\_num\_scal <- scale(matriz\_num)  
sd\_matriz <- apply(matriz\_num\_scal, MARGIN=1, FUN="sd")  
sel\_matriz <- (sd\_matriz > quantile(sd\_matriz, 0.975))  
matriz\_clust <- matriz\_num\_scal[sel\_matriz,]  
dim(matriz\_clust)

## [1] 71 336

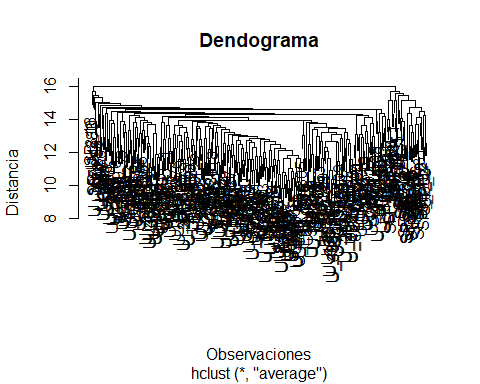
t\_matriz\_clust <- t(matriz\_clust)

Para realizar el dendograma únicamente se utilizarán los variables con mayor desviación estandar, concretamente el 2.5% (percentil 97.5) de las variables con mayor variabilidad entre las muestras. En este caso se seleccionan 71 de las 2811 variables.

dist\_matriz <- dist(t\_matriz\_clust)  
clust <- hclust(dist\_matriz, method = "average")  
plot(clust, main = "Dendograma", xlab = "Observaciones", ylab = "Distancia", labels = F)



plot(clust, main = "Dendograma", xlab = "Observaciones", ylab = "Distancia")



Se ha representado el dendograma del clustering jerarquico. En los resultados se muestran dos dendogramas, uno con las etiquetas de las observaciones y otro sin las etiquetas, para que sea más facil de visualizar.

*Discusión*

El uso de PCA (Análisis de Componentes Principales) y clustering jerárquico en estudios de metabolómica permite explorar patrones en los datos y reducir la dimensionalidad. Sin embargo, al trabajar con 2811 variables (alta dimensionalidad), es importante tener en cuenta las limitaciones que pueden afectar la interpretación de los resultados.

Una de las principales dificultades es la propia dimensionalidad, ya que el elevado número de variables puede generar ruido y redundancia, complicando la identificación de patrones biológicos. En PCA, esto puede diluir la señal real, mientras que en el clustering puede generar agrupaciones poco representativas.

Otra limitación es la necesidad de escalar y normalizar los datos, tanto en PCA como en clustering. En metabolómica, las concentraciones de los metabolitos pueden estar en diferentes órdenes de magnitud. Si no se realiza una normalización, los metabolitos con una escala mayor tendrán un peso mayor en la varianza explicada por los primeros componentes del PCA y afectarán la formación de clusters.

El clustering jerárquico, por su parte, es sensible a la elección de la métrica de distancia y del método de enlace, lo que puede generar agrupaciones artificiales.

Otro aspecto a tener en cuenta en estudios de metabolómica es la posible presencia de efectos de lote (batch effects). En estos casos, es recomendable verificar si se observa este tipo de efecto en PCA y, de ser necesario, aplicar métodos de corrección.

*Conclusiones*

Los resultados del análisis exploratorio arroja a la luz varios aspectos importantes de los datos que hay que tener en cuenta antes de proceder a un análisis estadístico. El PCA no permite determinar si se produce un efecto de lote ya que no se dispone de información sobre lotes, personal que ha realizado el LC-MS, dias en que se han realizado los análisis …

Porque es importante el PCA en este estudio: El PCA permite generar gráficos de dispersión de las primeras dos o tres componentes principales (PC1 vs. PC2 o PC1 vs. PC2 vs. PC3), donde: 🔹 Se pueden detectar grupos naturales en los datos (e.g., diferencias entre individuos sanos vs. enfermos). 🔹 Se pueden identificar patrones metabólicos asociados a condiciones específicas. 🔹 Se pueden encontrar outliers o muestras que se comportan de manera anómala. Cada componente principal es una combinación de metabolitos con diferentes pesos (loadings). 🔹 Metabolitos con mayores pesos en PC1 o PC2 pueden ser clave en la diferenciación de grupos. 🔹 Esto ayuda a seleccionar los metabolitos más relevantes para biomarcadores potenciales.

Clustering (Busqueda de patrones)

En este caso se realizará un cluster jerárquico basado en distancias euclídeas y enlaces promedio (“average linkage’’) (Como en el ejemplo). El método”average linkage” es un enfoque intermedio entre el “single” y el “complete”, que puede ser útil cuando se busca un balance en la formación de clusters

Ventajas Produce clusters equilibrados y suaves. Más robusto a la forma de los clusters que “single linkage” Desventajas Menos sensible a outliers que “single linkage”. Puede ser menos preciso con datos muy dispersos

Su objetivo es descubrir grupos en los datos, es decir grupos de individuos (o eventualmente de variables) que pueden considerarse más similares entre ellos que a los individuos de los otros grupos. Realizaremos un Clustering jerárquico debido a que no sabemos cuantos clusteres utilizar ( el K-means necesitas decidir cuántos clusters quieres de antemano).

Seleccionar solo las variables (genes) que tienen una mayor variabilidad es una técnica común para mejorar la interpretación de los análisis de clustering, ya que reduce el ruido y se enfoca en las características más relevantes para distinguir entre los grupos. En este caso, seleccionar los genes con las desviaciones estándar más altas es una forma sencilla de capturar aquellos que varían más entre las observaciones.

EJEMPLO DE DISCUSIÓN:

“El dendrograma generado con 336 observaciones muestra una clara división en dos grandes grupos, lo que sugiere que las muestras pueden agruparse en dos categorías principales. Al cortar el dendrograma a una distancia de 10, identificamos dos clusters, uno de los cuales contiene principalmente muestras de pacientes usuarios de opioides, mientras que el otro agrupa a los pacientes no usuarios. Esta separación podría reflejar diferencias metabólicas entre los dos grupos. Sin embargo, el tamaño desigual de los clusters (con un grupo considerablemente mayor que el otro) puede indicar que más análisis son necesarios para explorar subgrupos dentro de estas categorías. Los metabolitos seleccionados con alta variabilidad parecen ser relevantes para la separación entre los grupos, y una futura validación con datos adicionales podría confirmar estos hallazgos.”