

Análisis de datos Ómicos - PEC1

Zaida Munilla

2024-10-31

Contents

ABSTRACT	1
OBJETIVOS	1
MATERIALES Y MÉTODOS	2
RESULTADOS	5
DISCUSIÓN Y LIMITACIONES. CONCLUSIONES DEL ESTUDIO	10
ENLACE A REPOSITORIO GITHUB	10

ABSTRACT

En la primera PEC de la asignatura Análisis de Datos Ómicos he comenzado creando una cuenta en github para poder relacionar el Proyecto en R con un repositorio de mi cuenta en dicha aplicación. Posteriormente he realizado la exploración de los datos del dataset `human_cachexia.csv` creando previamente un contenedor del tipo `SummarizedExperiment`. Para realizar este contenedor he revisado los siguientes enlaces:

<https://bioconductor.org/packages/release/bioc/manuals/SummarizedExperiment/man/SummarizedExperiment.pdf>

<https://www.uv.es/ayala/docencia/tami/tami13.pdf> (Este documento me ha resultado de gran ayuda)

Finalmente, a partir del objeto `SEca` de clase `SummarizedExperiment`, he realizado una primera visualización de los datos para tener una idea de los datos contenidos en el dataset y sus posibles problemas a la hora de realizar un análisis estadístico.

OBJETIVOS

A lo largo del desarrollo de la PEC mi objetivo se ha centrado en entender la estructura de los objetos de tipo `SummarizedExperiment` para poder construir uno a partir del dataset `human_cachexia.csv` y posteriormente iniciar un análisis de los datos con la intención de detectar la necesidad de realizar una depuración de los datos previamente al inicio de su futuro análisis estadístico.

Por su parte, los objetivos del estudio que dieron lugar a los datos que se van a trabajar fueron los siguientes (fuente: <http://darwin.di.uminho.pt/metabolomicspackage/cachexia.html>):

La caquexia es un síndrome metabólico complejo asociado con una enfermedad subyacente (como el cáncer) y caracterizado por la pérdida de músculo con o sin pérdida de masa grasa. Mejores enfoques para detectar el inicio y la evolución de la atrofia muscular ayudarían a controlar los síndromes de atrofia y facilitarían la intervención temprana. Como es probable que los metabolitos producidos a partir de la descomposición

del tejido sean un indicador sensible de atrofia muscular, se recolectaron muestras de orina ya que varios productos finales del catabolismo muscular se excretan específicamente en la orina.

MATERIALES Y MÉTODOS

Los datos empleados han sido los del dataset `human_cachexia.csv` extraídos del siguiente link:

<https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2024-Cachexia>

Este dataset en formato csv contiene los datos de 77 individuos, en concreto, se recogieron un total de 77 muestras de orina, siendo 47 de ellos pacientes con caquexia y 30 pacientes control. Se adquirieron todos los espectros de RMN unidimensionales de muestras de orina y luego se detectaron y cuantificaron los metabolitos, es decir, para cada metabolito se midió su concentración.

Inicialmente, comencé la PEC creando la cuenta de github (<https://github.com/zmunilla>). Después siguiendo las instrucciones del siguiente enlace creé un repositorio y lo nombré como “Munilla-Garcia-Zaida-PEC1”

fuelle: http://destio.us.es/calvo/asignaturas/ge_esco/tutorialusargitgithubrstudio/UsarGitGithubconRStudio.html

En RStudio creé un nuevo proyecto con control de versiones indicando la url de mi repositorio.

Para ir copiando los nuevos archivos creados en RStudio en el repositorio github, desde la pestaña “Git” selecciono los archivos que quiero volcar en el repositorio, selecciono Commit, incluyo un mensaje en el cuadro de texto y de nuevo Commit y posteriormente “Push”.

Para iniciar el ejercicio descargué los archivos del enlace <https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2024-Cachexia> y los copié en la carpeta que había asignado al proyecto de R.

Para crear el contenedor de tipo “SummarizedExperiment” procedo con el siguiente código:

Primero cargo la librería “SummarizedExperiment”

```
library(SummarizedExperiment)

## Cargando paquete requerido: MatrixGenerics
## Cargando paquete requerido: matrixStats
##
## Adjuntando el paquete: 'MatrixGenerics'
## The following objects are masked from 'package:matrixStats':
##
##   colAlls, colAnyNAs, colAnys, colAveragesPerRowSet, 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, rowAveragesPerColSet,
##   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
## Cargando paquete requerido: GenomicRanges
```

```

## Cargando paquete requerido: stats4
## Cargando paquete requerido: BiocGenerics
##
## Adjuntando el paquete: '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, table,
##     tapply, union, unique, unsplit, which.max, which.min
## Cargando paquete requerido: S4Vectors
##
## Adjuntando el paquete: 'S4Vectors'
## The following object is masked from 'package:utils':
##
##     findMatches
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Cargando paquete requerido: IRanges
##
## Adjuntando el paquete: 'IRanges'
## The following object is masked from 'package:grDevices':
##
##     windows
## Cargando paquete requerido: GenomeInfoDb
## Cargando paquete requerido: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname)".
##
## Adjuntando el paquete: 'Biobase'
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

```

Cargo el dataset y veo una parte de los datos para ir haciéndome una idea del formato

```
dfca <- read.csv("human_cachexia.csv", header=TRUE, sep=",")
head(dfca)[1:3,1:3]
```

```
## Patient.ID Muscle.loss X1.6.Anhydro.beta.D.glucose
## 1 PIF_178 cachexic 40.85
## 2 PIF_087 cachexic 62.18
## 3 PIF_090 cachexic 270.43
```

```
dim(dfca)
```

```
## [1] 77 65
```

Transformo los datos en una matriz y selecciono únicamente los valores de los distintos metabolitos. Para poder crear el SummarizedExperiment hago la traspuesta de la matriz dado que necesitamos que las distintas muestras (en este caso las muestras de orina de cada individuo) se dispongan en las columnas, y los valores de los metabolitos en las filas:

```
mat <- data.matrix(subset.data.frame(dfca[,3:65], row.names=1, col.names=dfca$Patient.ID))
```

```
## Warning: In subset.data.frame(dfca[, 3:65], row.names = 1, col.names = dfca$Patient.ID) :
## extra arguments 'row.names', 'col.names' will be disregarded
```

```
matt <- t(mat)
colnames(matt) <- dfca$Patient.ID
dim(matt)
```

```
## [1] 63 77
```

Para realizar el otro dataframe que conforma el SummarizedExperiment selecciono las dos primeras columnas del dataset de inicio e indico que la primera columna se trata de los nombre de las filas. Así el único atributo de las muestras será si se trata de muestra control o de individuos con cachexia.

```
colca <- data.frame(dfca[,1:2], row.names=1)
colca$Muscle.loss <- as.factor(colca$Muscle.loss)
head(colca,5)
```

```
## Muscle.loss
## PIF_178 cachexic
## PIF_087 cachexic
## PIF_090 cachexic
## NETL_005_V1 cachexic
## PIF_115 cachexic
```

```
dim(colca)
```

```
## [1] 77 1
```

```
table(colca)
```

```
## Muscle.loss
## cachexic control
## 47 30
```

Procedemos a crear la lista con los metadatos del estudio que he extraído de la página: <http://darwin.di.uminho.pt/metabolomicspackage/cachexia.html>

```
met <- c(name='Eisner et al.',
        lab='Varios',
        contact="chrisbcl@hotmail.com",
        title='Learning to predict cancer-associated skeletal muscle wasting from 1h-nmr profil
```

```
abstract='Cachexia is a complex metabolic syndrome associated with an underlying illness'
url='https://www.metaboanalyst.ca/resources/data/human_cachexia.csv')
```

Con el siguiente código uno las 3 piezas y creo el contenedor que denomino SEca:

```
SEca <- SummarizedExperiment(assays=list(counts=matt),
                             colData=colca,
                             metadata = met)
```

SEca

```
## class: SummarizedExperiment
## dim: 63 77
## metadata(6): name lab ... abstract url
## assays(1): counts
## rownames(63): X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide ...
##   pi.Methylhistidine tau.Methylhistidine
## rowData names(0):
## colnames(77): PIF_178 PIF_087 ... NETL_003_V1 NETL_003_V2
## colData names(1): Muscle.loss
```

Para guardarlo por separado del resto de archivos:

```
save(SEca, file="SEca.RData")
```

RESULTADOS

Procedo a obtener un análisis básico de los datos del contenedor creado:

```
head(colData(SEca))
```

```
## DataFrame with 6 rows and 1 column
##           Muscle.loss
##           <factor>
## PIF_178      cachexic
## PIF_087      cachexic
## PIF_090      cachexic
## NETL_005_V1   cachexic
## PIF_115      cachexic
## PIF_110      cachexic
```

```
dim(colData(SEca))
```

```
## [1] 77  1
```

Puedo acceder a los metadata con el siguiente comando:

```
metadata(SEca)
```

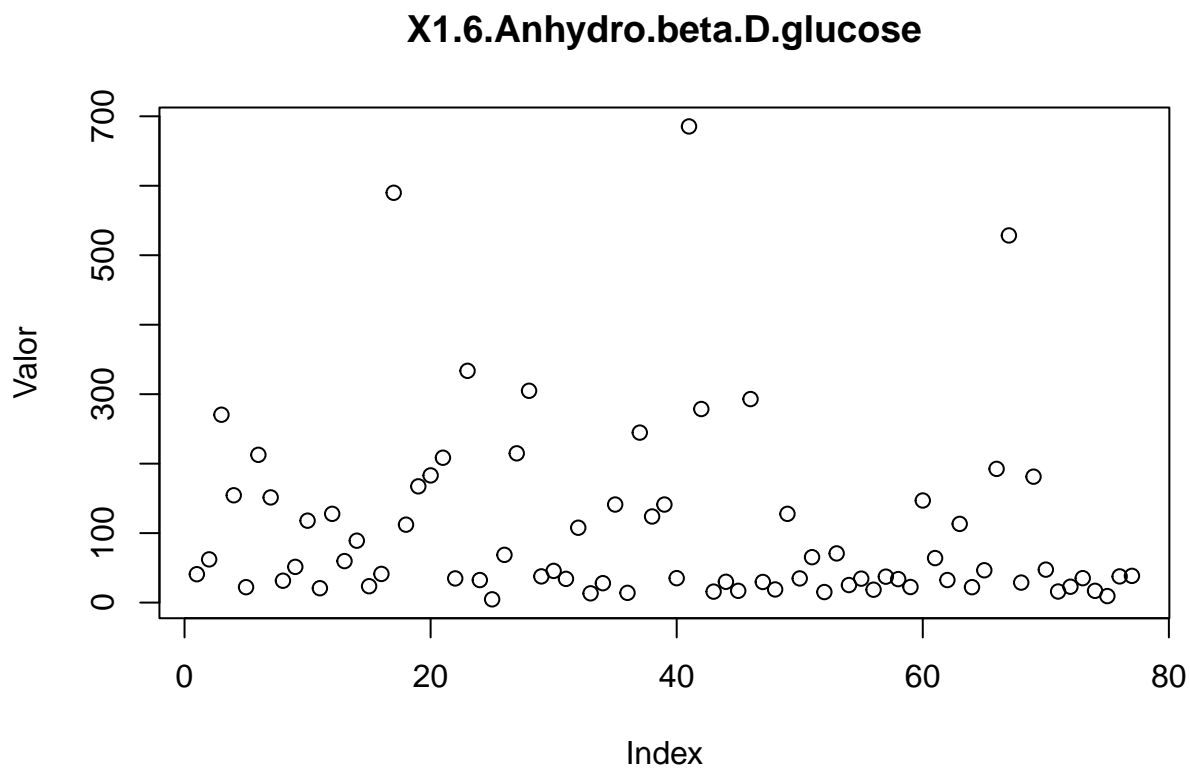
```
## $name
## [1] "Eisner et al."
##
## $lab
## [1] "Varios"
##
## $contact
## [1] "chrisbcl@hotmail.com"
##
```

```
## $title
## [1] "Learning to predict cancer-associated skeletal muscle wasting from 1h-nmr profiles of urinary m
34, 2010."
##
## $abstract
## [1] "Cachexia is a complex metabolic syndrome associated with an underlying illness (such as cancer)
##
## $url
## [1] "https://www.metaboanalyst.ca/resources/data/human_cachexia.csv"
head(assay(SEca))[1:3,1:3]
```

```
##
## X1.6.Anhydro.beta.D.glucose 40.85 62.18 270.43
## X1.Methylnicotinamide      65.37 340.36 64.72
## X2.Aminobutyrate           18.73 24.29 12.18
```

Podría plasmar un plot de cada uno de los metabolitos con su distribución en un plot de la siguiente manera:

```
plot(assay(SEca)[1,], main=rownames(assay(SEca))[1], ylab="Valor")
```



Esto me permitiría poder hacerme una idea de la presencia de outliers. Además, teniendo en cuenta que las primeras 47 muestras se trata de los individuos que presentaban cachexia y que los 30 últimos los individuos control, con estos gráficos podríamos intuir alguna relación entre la cachexia y los valores de alguno de los metabolitos.

Ahora realizaré un summary de la distribución de cada uno de los metabolitos, de manera que se pueda detectar también la presencia de valores atípicos (valores máximos o mínimos muy alejados de la media, por ejemplo).

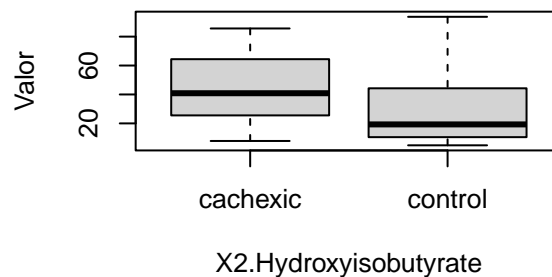
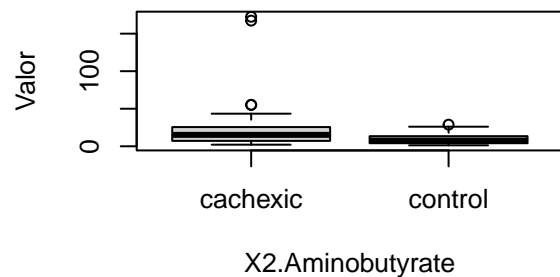
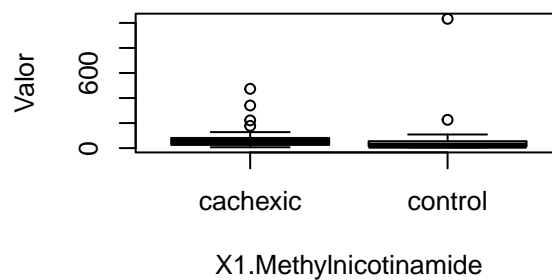
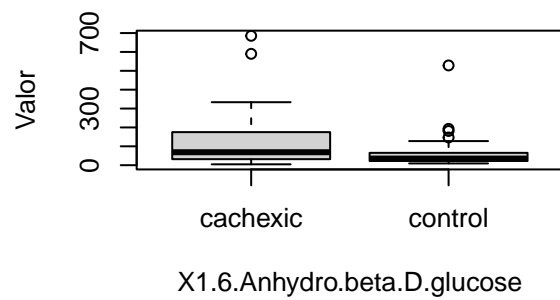
```
apply(t(assay(SEca)),2,summary)
```

```
##          X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide X2.Aminobutyrate
## Min.                4.7100                6.42000        1.28000
## 1st Qu.             28.7900             15.80000        5.26000
## Median             45.6000             36.60000       10.49000
## Mean              105.6304             71.57364       18.15974
## 3rd Qu.            141.1700             73.70000       19.49000
## Max.               685.4000            1032.77000      172.43000
##          X2.Hydroxyisobutyrate X2.Oxoglutarate X3.Aminoisobutyrate
## Min.                4.85000         5.5300         2.61000
## 1st Qu.             15.80000         22.4200         11.70000
## Median             32.46000         55.1500         22.65000
## Mean              37.25065         145.0871         76.75636
## 3rd Qu.            54.60000         92.7600         56.26000
## Max.              93.69000        2465.1300        1480.30000
##          X3.Hydroxybutyrate X3.Hydroxyisovalerate X3.Indoxylsulfate
## Min.                1.70000         0.92000         27.6600
## 1st Qu.             5.99000         5.26000         82.2700
## Median             11.70000         12.55000        144.0300
## Mean              21.71701         21.64779         218.8792
## 3rd Qu.            29.96000         30.27000        333.6200
## Max.             175.91000        164.02000        1043.1500
##          X4.Hydroxyphenylacetate  Acetate  Acetone  Adipate  Alanine
## Min.                15.490   3.49000   2.29000   1.55000   16.7800
## 1st Qu.             41.680  16.28000   4.95000   6.11000   78.2600
## Median             70.110  39.65000   7.10000  10.18000  194.4200
## Mean             112.021  66.14143  11.42701  24.75636  273.5623
## 3rd Qu.            145.470  86.49000  10.49000  19.11000  399.4100
## Max.             796.320 411.58000 206.44000 327.01000 1312.9100
##          Asparagine  Betaine  Carnitine  Citrate  Creatine  Creatinine
## Min.             6.69000   2.29000   2.18000   59.740    2.7500   1002.250
## 1st Qu.          20.49000  28.79000  14.44000   788.400   17.6400   3498.190
## Median          42.10000  64.72000  23.81000  1790.050   44.2600   7631.200
## Mean           62.28364  90.32468  52.08506  2235.346  126.8319   8733.972
## 3rd Qu.          89.12000 127.74000  60.95000  3071.740  117.9200  12332.580
## Max.           273.14000 391.51000 487.85000 13629.610 1863.1100  33860.350
##          Dimethylamine  Ethanolamine  Formate  Fucose  Fumarate  Glucose
## Min.             41.2600         16.1200    6.420    5.70000   0.79000   26.8400
## 1st Qu.          142.5900         86.4900   53.520   29.37000   2.23000   80.6400
## Median          304.9000        204.3800   95.580   61.56000   4.10000  210.6100
## Mean           358.1661        276.2604  147.403   88.66883   8.44013  559.8445
## 3rd Qu.          454.8600        407.4800  167.340  123.97000   7.85000  407.4800
## Max.           1556.2000       1436.5500 1480.300 407.48000  96.54000 8690.6200
##          Glutamine  Glycine  Glycolate  Guanidoacetate  Hippurate  Histidine
## Min.             23.3400    38.0900    5.4200         7.03000    92.760   14.1500
## 1st Qu.          113.3000   262.4300   50.9100        33.78000   492.750   66.6900
## Median          225.8800   528.4800   130.3200        64.72000  1224.150  174.1600
## Mean           306.8716   880.7174   187.9894        86.37052  2286.838  292.6375
## 3rd Qu.          445.8600  1096.6300  267.7400       108.85000  2921.930  419.8900
## Max.           1685.8100  5064.4500  720.5400       561.16000 19341.340 1863.1100
##          Hypoxanthine  Isoleucine  Lactate  Leucine  Lysine  Methylamine
## Min.             3.78000    1.790000    7.3200    2.51000   10.4900    1.51000
## 1st Qu.          20.70000    3.900000   35.5200    9.12000   30.2700    5.26000
```

## Median	40.04000	7.170000	81.4500	19.11000	69.4100	14.73000
## Mean	61.09766	8.709091	158.4565	24.36364	108.7942	17.37623
## 3rd Qu.	83.93000	11.250000	139.7700	31.19000	121.5100	24.05000
## Max.	265.07000	40.040000	3640.9500	103.54000	788.4000	52.46000
##	Methylguanidine N.N.Dimethylglycine O.Acetylcarnitine Pantothenate					
## Min.	1.70000		0.79000		1.23000	2.59000
## 1st Qu.	4.26000		7.03000		3.94000	11.13000
## Median	7.85000		21.98000		11.47000	22.65000
## Mean	15.32455		26.34961		19.73338	44.88377
## 3rd Qu.	19.30000		40.04000		20.91000	41.26000
## Max.	141.17000		120.30000		254.68000	692.29000
##	Pyroglutamate Pyruvate Quinolinatate Serine Succinate Sucrose					
## Min.	21.3300	0.90000	5.21000	16.1200	1.72000	6.4900
## 1st Qu.	68.7200	4.85000	26.58000	83.1000	8.58000	19.3000
## Median	157.5900	13.46000	51.42000	142.5900	30.88000	40.8500
## Mean	211.4478	21.29442	66.43948	197.6869	60.22909	113.2278
## 3rd Qu.	301.8700	29.08000	87.36000	270.4300	74.44000	94.6300
## Max.	1064.2200	184.93000	259.82000	1248.8800	589.93000	2079.7400
##	Tartrate Taurine Threonine Trigonelline Trimethylamine.N.oxide					
## Min.	2.20000	17.8100	8.2500	10.0700		55.7000
## 1st Qu.	6.89000	99.4800	31.8200	53.5200		175.9100
## Median	12.94000	249.6400	64.0700	114.4300		383.7500
## Mean	40.00403	525.1235	95.3574	270.4361		652.1569
## 3rd Qu.	25.79000	665.1400	137.0000	340.3600		735.1000
## Max.	837.15000	4272.6900	450.3400	2252.9600		5486.2500
##	Tryptophan Tyrosine Uracil Valine Xylose cis.Aconitate					
## Min.	8.67000	4.22000	3.10000	4.10000	10.0700	12.9400
## 1st Qu.	21.33000	23.57000	11.94000	12.18000	29.9600	36.2300
## Median	46.99000	60.34000	27.39000	33.12000	50.4000	129.0200
## Mean	66.24312	81.75727	35.55766	35.66701	100.9334	204.2197
## 3rd Qu.	96.54000	113.30000	44.26000	50.40000	89.1200	254.6800
## Max.	259.82000	539.15000	179.47000	160.77000	2164.6200	1863.1100
##	myo.Inositol trans.Aconitate pi.Methylhistidine tau.Methylhistidine					
## Min.	11.5900		4.90000		11.3600	8.00000
## 1st Qu.	30.2700		12.43000		67.3600	27.39000
## Median	78.2600		26.84000		162.3900	68.72000
## Mean	135.3975		40.63039		370.2883	89.68688
## 3rd Qu.	167.3400		57.40000		387.6100	130.32000
## Max.	854.0600		217.02000		2697.2800	317.35000

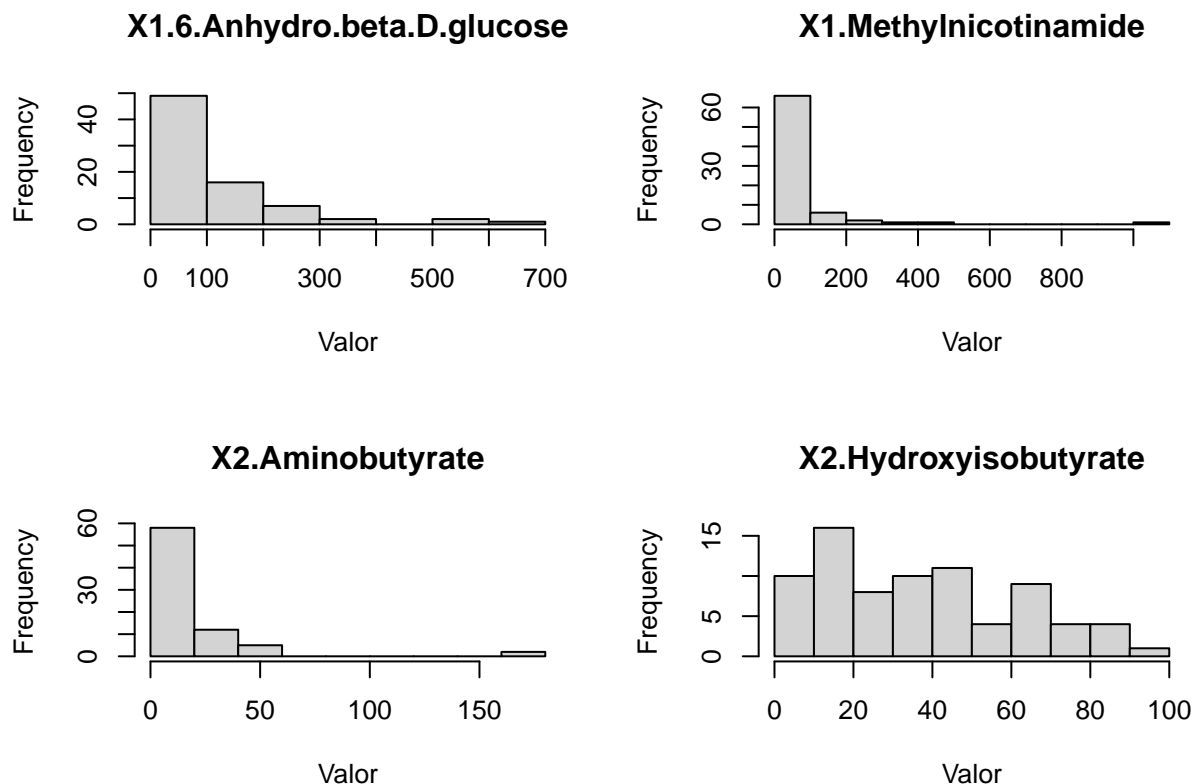
Otro ejemplo de gráfico sería aplicar a todos los metabolitos un boxplot diferenciando los dos grupos “cachexia” y “control”. En el siguiente código sólo se muestran los 4 primeros metabolitos como ejemplo.

```
par(mfrow=c(2,2))
for (i in 1:4)
  boxplot(assay(SEca)[i,] ~ colca$Muscle.loss, ylab="Valor",xlab=rownames(assay(SEca))[i])
```

Y del mismo modo se podrían realizar histogramas con la distribución de frecuencias de los valores de los distintos metabolitos. Se muestran los primeros 4 metabolitos.

```
par(mfrow=c(2,2))
for (i in 1:4)
  hist(assay(SEca)[i,], xlab= "Valor", main=rownames(assay(SEca))[i])
```



Para crear el archivo con los metadatos en un archivo .md he creado un nuevo archivo desde File>New file>Markdown file y he copiado los metadatos y los he pegado en el archivo creando el archivo metadatos_PEC1.md

Para crear el archivo de los datos en formato texto:

```
write.table(dfca, file="human_cachexia.txt", row.names=FALSE, sep=",")
```

DISCUSIÓN Y LIMITACIONES. CONCLUSIONES DEL ESTUDIO

La creación del contenedor en formato SummarizedExperiment me ha llevado un tiempo, dado que no lo había trabajado antes, pero una vez realizado el primero ya resulta más sencillo poder aplicarlo en futuras ocasiones.

La extracción básica de datos me ha permitido ver que existen muchos valores atípicos que deberían estudiarse de cara a plantear eliminar ciertos registros del estudio.

Según algunos de los boxplots sí que parece haber relación entre los valores de ciertos metabolitos en orina y la presencia de cachexia en los individuos, por lo que parece interesante su estudio en profundidad (mediante análisis estadísticos) para valorar la utilidad de ciertos marcadores a la hora de determinar una cachexia incipiente en individuos que aún no presentan signos físicos de la misma.

ENLACE A REPOSITORIO GITHUB

<https://github.com/zmunilla/Munilla-Garcia-Zaida-PEC1/tree/main>