

Informe PEC1

Alejandro García

2025-03-24

Contents

Resumen	1
Objetivos	1
Materiales y métodos	2
Selección de los datos	2
creación del <i>SummarizedExperiment</i>	4
Resultados	6
Estructura de los datos	6
Exploración de los datos	9
Análisis univariante	9
Análisis multivariante	11
Discusión	13
Conclusión	14
Bibliografía	14

Resumen

En este trabajo hemos realizado una exploración de un dataset que contiene medidas de metabolitos procedentes de pacientes con caquexia. La caquexia es una afección caracterizada por una pérdida de más del 10 % del peso corporal, que incluye pérdida de masa muscular y grasa, en una persona que no está tratando de perder peso. También está asociada con otras enfermedades tales como el cáncer. El objetivo es identificar diferencias en los perfiles metabólicos de estos pacientes para así poder entender mejor la enfermedad y tratar de obtener alguna terapia para abordarla. Hemos escogido un dataset proveniente de (https://rest.xialab.ca/api/download/metaboanalyst/human_cachexia.csv) y en primer lugar hemos creado un objeto *SummarizedExperiment* con los datos de dicho dataset. La clase *SummarizedExperiment* es una extensión de *ExpressionSet*, utilizada por muchas aplicaciones y bases de datos como *metabolomicsWorkbench*. A continuación, hemos procedido a acceder a los datos una vez creado el objeto y hemos realizado dos tipos de análisis: un análisis univariante para ver la dispersión de los datos y comprobar que es mejor trabajar tomando logaritmos de los datos; y un análisis multivariante en el cual hemos podido observar diferentes patrones entre los dos grupos de pacientes del dataset.

Objetivos

El objetivo principal de esta PEC es realizar un análisis exploratorio de unos datos obtenidos de (https://rest.xialab.ca/api/download/metaboanalyst/human_cachexia.csv) utilizando el programa estadístico R y

las librerías de datos ómicos integradas en Bioconductor. En concreto, se requiere que los datos se carguen en un objeto de clase `SummarizedExperiment` y que se acceda a ellos a través de este objeto. Otro objetivo es familiarizarnos con el entorno de GitHub como sistema de control de versiones. Crearemos un repositorio en el que almacenaremos los resultados de este trabajo.

Materiales y métodos

Los datos con los que se va a realizar este trabajo son los obtenidos de (https://rest.xialab.ca/api/download/metaboanalyst/human_cachexia.csv). Estos datos provienen de un estudio realizado en pacientes con caquexia. La caquexia es una afección caracterizada por una pérdida de más del 10 % del peso corporal, que incluye pérdida de masa muscular y grasa, en una persona que no está tratando de perder peso. También está asociados con otras enfermedades tales como el cáncer. El estudio del perfil metabólico de estos pacientes comparados con el de personas sanas podría ayudarnos a entender mejor cómo actúa la enfermedad en los pacientes y dar indicios para una posible cura de la misma.

El dataset se compone de 77 muestras de orina. Estas muestras de orina provienen de dos grupos de pacientes: un grupo de 47 pacientes que sufren caquexia; y otro grupo compuesto de 30 pacientes control. El dataset se compone de: Una columna de identificación de los pacientes, una columna donde se indica a qué grupo pertenece el paciente (control o caquético) y el resto son columnas donde se mide la concentración de metabolitos. Excepto las dos primeras columnas, todas las demás contienen valores numéricos. Las muestras no están apareadas, es decir, todos los pacientes representados en el dataset son distintos y no se han detectado valores ausentes en el dataset. Vamos a realizar una exploración de los datos que va a consistir en: - Análisis univariante de los datos, mediante boxplots y/o histogramas. - Análisis multivariante de los datos, mediante Análisis de componentes principales y agrupamiento jerárquico. (Ref: https://github.com/ASPteaching/Análisis_de_datos_omicos-Ejemplo_o-Microarrays)

En primer lugar, vamos a explicar como vamos a crear el objeto *SummarizedExperiment*

Selección de los datos

Vamos a cargar los datos usando la función *read.csv* ya que los datos están en formato .csv

```
datos <- read.csv("human_cachexia.csv")
head(datos)
```

```
## Patient.ID Muscle.loss X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide
## 1 PIF_178 cachexic 40.85 65.37
## 2 PIF_087 cachexic 62.18 340.36
## 3 PIF_090 cachexic 270.43 64.72
## 4 NETL_005_V1 cachexic 154.47 52.98
## 5 PIF_115 cachexic 22.20 73.70
## 6 PIF_110 cachexic 212.72 31.82
## X2.Aminobutyrate X2.Hydroxyisobutyrate X2.Oxoglutarate X3.Aminoisobutyrate
## 1 18.73 26.05 71.52 1480.30
## 2 24.29 41.68 67.36 116.75
## 3 12.18 65.37 23.81 14.30
## 4 172.43 74.44 1199.91 555.57
## 5 15.64 83.93 33.12 29.67
## 6 18.36 80.64 47.94 17.46
## X3.Hydroxybutyrate X3.Hydroxyisovalerate X3.Indoxylsulfate
## 1 56.83 10.07 566.80
## 2 43.82 79.84 368.71
## 3 5.64 23.34 665.14
## 4 175.91 25.03 411.58
## 5 76.71 69.41 165.67
## 6 31.82 35.16 183.09
```

##	X4.Hydroxyphenylacetate	Acetate	Acetone	Adipate	Alanine	Asparagine	Betaine	
## 1		120.30	126.47	9.49	38.09	314.19	159.17 109.95	
## 2		432.68	212.72	11.82	327.01	871.31	157.59 244.69	
## 3		292.95	314.19	4.44	131.63	464.05	89.12 116.75	
## 4		214.86	37.34	206.44	144.03	589.93	273.14 278.66	
## 5		97.51	407.48	44.26	15.03	1118.79	42.52 391.51	
## 6		132.95	81.45	14.44	25.28	237.46	157.59 66.69	
##	Carnitine	Citrate	Creatine	Creatinine	Dimethylamine	Ethanolamine	Formate	
## 1	265.07	3714.50	196.37	16481.60	632.70	645.48	441.42	
## 2	120.30	2617.57	212.72	15835.35	607.89	487.85	252.14	
## 3	25.03	862.64	221.41	24587.66	735.10	407.48	249.64	
## 4	200.34	13629.61	85.63	20952.22	1064.22	820.57	468.72	
## 5	84.77	854.06	105.64	6768.26	242.26	365.04	114.43	
## 6	40.04	1958.63	200.34	15677.78	614.00	459.44	314.19	
##	Fucose	Fumarate	Glucose	Glutamine	Glycine	Glycolate	Guanidoacetate	Hippurate
## 1	336.97	7.69	395.44	871.31	2038.56	685.40	154.47	4582.50
## 2	198.34	18.92	8690.62	601.85	1107.65	651.97	109.95	1737.15
## 3	186.79	7.10	1352.89	301.87	620.17	141.17	183.09	4315.64
## 4	407.48	96.54	862.64	1685.81	5064.45	70.81	102.51	757.48
## 5	26.05	19.69	6836.29	432.68	395.44	26.58	52.98	1152.86
## 6	123.97	5.05	512.86	298.87	482.99	428.38	57.97	3568.85
##	Histidine	Hypoxanthine	Isoleucine	Lactate	Leucine	Lysine	Methylamine	
## 1	925.19	97.51	5.58	106.70	42.10	146.94	52.46	
## 2	845.56	82.27	8.17	368.71	77.48	284.29	23.57	
## 3	284.29	114.43	9.30	749.95	31.50	97.51	18.73	
## 4	1043.15	223.63	37.71	368.71	103.54	290.03	48.91	
## 5	327.01	66.69	40.04	3640.95	101.49	122.73	27.94	
## 6	459.44	62.80	8.17	113.30	28.79	120.30	36.97	
##	Methylguanidine	N.N.Dimethylglycine	O.Acetylcarnitine	Pantothenate				
## 1	9.97	23.34	52.98	25.79				
## 2	7.69	87.36	50.40	186.79				
## 3	4.66	24.53	5.58	145.47				
## 4	141.17	40.04	254.68	42.52				
## 5	5.31	46.06	45.60	74.44				
## 6	43.38	24.29	13.46	35.52				
##	Pyroglutamate	Pyruvate	Quinolinat	Serine	Succinate	Sucrose	Tartrate	Taurine
## 1	437.03	21.12	165.67	284.29	154.47	45.15	97.51	1919.85
## 2	437.03	36.97	72.97	391.51	244.69	459.44	32.79	1261.43
## 3	713.37	29.37	192.48	295.89	142.59	160.77	16.28	4272.69
## 4	566.80	64.07	86.49	1248.88	144.03	111.05	837.15	1525.38
## 5	184.93	12.30	38.09	206.44	68.72	75.19	4.53	468.72
## 6	432.68	32.79	112.17	387.61	33.45	336.97	24.05	2059.05
##	Threonine	Trigonelline	Trimethylamine.N.oxide	Tryptophan	Tyrosine	Uracil		
## 1	184.93	943.88	2121.76	259.82	290.03	111.05		
## 2	198.34	208.51	639.06	83.10	167.34	46.99		
## 3	109.95	192.48	1152.86	82.27	60.34	31.50		
## 4	376.15	992.27	1450.99	235.10	323.76	30.57		
## 5	64.07	86.49	172.43	103.54	142.59	44.26		
## 6	105.64	862.64	880.07	239.85	127.74	29.67		
##	Valine	Xylose	cis.Aconitate	myo.Inositol	trans.Aconitate	pi.Methylhistidine		
## 1	86.49	72.24	237.46	135.64	51.94	157.59		
## 2	109.95	192.48	333.62	376.15	217.02	307.97		
## 3	59.15	2164.62	330.30	86.49	58.56	145.47		
## 4	102.51	125.21	1863.11	247.15	75.94	249.64		

```
## 5 160.77 186.79      101.49      749.95      98.49      84.77
## 6  36.97  89.12      287.15      129.02     121.51     399.41
##   tau.Methylhistidine
## 1      160.77
## 2      130.32
## 3       83.93
## 4      254.68
## 5       79.84
## 6       68.72
```

creación del *SummarizedExperiment*

Como hemos avanzado, las dos primeras columnas indican la identificación del paciente y el grupo al que pertenecen. Estas dos columnas nos servirán como información de las filas a la hora de construir el objeto *SummarizedExperiment*. Vemos también que las mediciones empiezan a partir de la tercera columna.

```
library(SummarizedExperiment)
```

```
## Cargando paquete requerido: MatrixGenerics
## Warning: package 'MatrixGenerics' was built under R version 4.4.2
## Cargando paquete requerido: matrixStats
## Warning: package 'matrixStats' was built under R version 4.4.2
##
## Adjuntando el paquete: '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
## 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, saveRDS, 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
## Warning: package 'IRanges' was built under R version 4.4.2
##
## Adjuntando el paquete: 'IRanges'
## The following object is masked from 'package:grDevices':
##
##   windows
## Cargando paquete requerido: GenomeInfoDb
## Warning: package 'GenomeInfoDb' was built under R version 4.4.2
## 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
# Separar los datos (valores de metabolitos) y metadatos (información como pacientes y categorías)
metabolitos <- as.matrix(datos[, -c(1, 2)]) # creamos la matriz con los datos de las medidas de los me
rownames(metabolitos) <- datos$Patient.ID
metadatos <- data.frame(Grupo = datos$Muscle.loss, row.names = datos$Patient.ID )
se <- SummarizedExperiment(
  assays = list(counts = t(metabolitos)),

```

```
colData = metadatos
)
se

## class: SummarizedExperiment
## dim: 63 77
## metadata(0):
## 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): Grupo
```

Resultados

Estructura de los datos

SummarizedExperiment es un contenedor de tipo matriz en el que las filas representan características de interés (en este caso, la información de los pacientes) y las columnas representan muestras (en este caso los metabolitos). Los objetos contienen uno o más ensayos, cada uno representado por un objeto tipo matriz de modo numérico o de otro tipo. Las filas de un objeto *SummarizedExperiment* representan características de interés. La información sobre estas características se almacena en un objeto *DataFrame*, accesible mediante la función `rowData()`. Cada fila del *DataFrame* proporciona información sobre la característica en la fila correspondiente del objeto *SummarizedExperiment*. Las columnas del *DataFrame* representan diferentes atributos de las características de interés. Para recuperar los datos del experimento de un objeto *SummarizedExperiment* se puede utilizar el accesorio `assays()`. Un objeto puede tener múltiples conjuntos de datos de ensayos, a cada uno de los cuales se puede acceder utilizando el operador `$`. El conjunto de datos de las vías respiratorias contiene sólo un ensayo (recuentos). Aquí cada fila representa una transcripción genética y cada columna una de las muestras.

```
head(assays(se)$counts)
```

```
##               PIF_178 PIF_087 PIF_090 NETL_005_V1 PIF_115 PIF_110
## X1.6.Anhydro.beta.D.glucose  40.85  62.18  270.43    154.47  22.20  212.72
## X1.Methylnicotinamide       65.37  340.36  64.72     52.98  73.70  31.82
## X2.Aminobutyrate            18.73   24.29  12.18    172.43  15.64  18.36
## X2.Hydroxyisobutyrate       26.05  41.68  65.37     74.44  83.93  80.64
## X2.Oxoglutarate             71.52  67.36  23.81    1199.91  33.12  47.94
## X3.Aminoisobutyrate        1480.30 116.75  14.30     555.57  29.67  17.46
##               NETL_019_V1 NETCR_014_V1 NETCR_014_V2 PIF_154
## X1.6.Anhydro.beta.D.glucose  151.41      31.50     51.42  117.92
## X1.Methylnicotinamide        36.60      6.82     30.27   52.46
## X2.Aminobutyrate              8.67      4.18      7.54   19.49
## X2.Hydroxyisobutyrate        42.52     12.94     34.81   72.24
## X2.Oxoglutarate              223.63     25.03     80.64   73.70
## X3.Aminoisobutyrate          56.26      8.67     17.99   57.97
##               NETL_022_V1 NETL_022_V2 NETL_008_V1 PIF_146 PIF_119
## X1.6.Anhydro.beta.D.glucose   20.70    127.74     59.74   89.12  23.57
## X1.Methylnicotinamide        221.41    177.68     50.91   32.79   6.89
## X2.Aminobutyrate              15.18     12.68      6.82   10.38   2.12
## X2.Hydroxyisobutyrate        28.79     15.03     46.06   32.14   7.85
## X2.Oxoglutarate              357.81     68.03    111.05   32.46   8.33
## X3.Aminoisobutyrate           93.69    105.64      8.08   43.38   2.97
```

##		PIF_099	PIF_162	PIF_160	PIF_113	PIF_143
##	X1.6.Anhydro.beta.D.glucose	41.26	589.93	112.17	167.34	183.09
##	X1.Methylnicotinamide	8.67	21.98	25.28	19.89	90.92
##	X2.Aminobutyrate	2.56	15.18	15.49	13.46	8.94
##	X2.Hydroxyisobutyrate	7.85	46.06	47.94	31.19	64.07
##	X2.Oxoglutarate	6.89	32.79	28.79	47.94	20.49
##	X3.Aminoisobutyrate	6.36	31.82	16.12	79.04	18.73
##		NETCR_007_V1	NETCR_007_V2	PIF_137	PIF_100	
##	X1.6.Anhydro.beta.D.glucose	208.51		34.81	333.62	32.46
##	X1.Methylnicotinamide	53.52		95.58	35.87	9.68
##	X2.Aminobutyrate	5.26		23.57	7.92	3.90
##	X2.Hydroxyisobutyrate	47.94		68.03	54.60	11.02
##	X2.Oxoglutarate	212.72		287.15	20.49	170.72
##	X3.Aminoisobutyrate	50.40		104.58	63.43	2.97
##		NETL_004_V1	PIF_094	PIF_132	PIF_163	NETCR_003_V1
##	X1.6.Anhydro.beta.D.glucose	4.71	68.72	214.86	304.90	37.71
##	X1.Methylnicotinamide	11.13	13.87	127.74	25.79	10.80
##	X2.Aminobutyrate	43.38	12.18	31.50	27.11	5.00
##	X2.Hydroxyisobutyrate	30.88	25.03	33.78	40.45	8.25
##	X2.Oxoglutarate	104.58	28.22	88.23	70.81	11.70
##	X3.Aminoisobutyrate	54.05	72.97	64.07	126.47	8.41
##		NETL_028_V1	NETL_028_V2	NETCR_013_V1	NETL_020_V1	
##	X1.6.Anhydro.beta.D.glucose	45.60		34.12	107.77	13.33
##	X1.Methylnicotinamide	473.43		92.76	16.61	50.91
##	X2.Aminobutyrate	16.28		8.25	26.84	2.92
##	X2.Hydroxyisobutyrate	63.43		16.61	32.46	40.85
##	X2.Oxoglutarate	221.41		55.15	62.80	46.99
##	X3.Aminoisobutyrate	15.49		3.39	29.67	22.42
##		NETL_020_V2	PIF_192	NETCR_012_V1	NETCR_012_V2	
##	X1.6.Anhydro.beta.D.glucose	27.94	141.17		14.01	244.69
##	X1.Methylnicotinamide	80.64	68.03		46.06	116.75
##	X2.Aminobutyrate	15.80	40.85		29.08	40.04
##	X2.Hydroxyisobutyrate	64.72	12.81		24.53	61.56
##	X2.Oxoglutarate	88.23	26.05		64.07	174.16
##	X3.Aminoisobutyrate	11.70	21.76		13.07	53.52
##		PIF_089	NETCR_002_V1	PIF_179	PIF_114	NETCR_006_V1
##	X1.6.Anhydro.beta.D.glucose	123.97		141.17	35.16	685.40
##	X1.Methylnicotinamide	81.45		28.50	26.58	36.23
##	X2.Aminobutyrate	55.15		20.29	5.21	32.46
##	X2.Hydroxyisobutyrate	70.81		14.30	30.27	85.63
##	X2.Oxoglutarate	92.76		97.51	7.39	25.03
##	X3.Aminoisobutyrate	561.16		8.41	8.41	184.93
##		PIF_141	NETCR_025_V1	NETCR_025_V2	NETCR_016_V1	
##	X1.6.Anhydro.beta.D.glucose	15.80		29.96	16.95	292.95
##	X1.Methylnicotinamide	23.57		96.54	114.43	57.97
##	X2.Aminobutyrate	17.99		6.55	2.53	167.34
##	X2.Hydroxyisobutyrate	37.34		65.37	77.48	82.27
##	X2.Oxoglutarate	21.33		1053.63	2465.13	468.72
##	X3.Aminoisobutyrate	26.84		14.15	19.49	53.52
##		PIF_116	PIF_191	PIF_164	NETL_013_V1	PIF_188
##	X1.6.Anhydro.beta.D.glucose	29.67	18.92	127.74		34.81
##	X1.Methylnicotinamide	70.11	24.53	1032.77		12.30
##	X2.Aminobutyrate	5.58	3.29	8.58		5.87
##	X2.Hydroxyisobutyrate	18.73	10.49	66.02		15.18
					PIF_195	
					15.80	8.17

```

## X2.Oxoglutarate      5.53    9.68    38.09      16.78    7.24    5.64
## X3.Aminoisobutyrate   2.61    26.84    66.69      11.25    3.13    5.99
##                      NETCR_015_V1 PIF_102 NETL_010_V1 NETL_010_V2
## X1.6.Anhydro.beta.D.glucose  70.81    25.28      34.47    18.54
## X1.Methylnicotinamide   75.94   101.49      12.81     8.41
## X2.Aminobutyrate        22.65     8.33       3.78     3.78
## X2.Hydroxyisobutyrate   60.95    59.15       8.33     4.85
## X2.Oxoglutarate        230.44    88.23      14.30     8.08
## X3.Aminoisobutyrate     53.52    22.65      24.29    22.87
##                      NETL_001_V1 NETCR_015_V2 NETCR_005_V1 PIF_111
## X1.6.Anhydro.beta.D.glucose  37.34      33.78      22.42   146.94
## X1.Methylnicotinamide   55.15      53.52      55.15    10.07
## X2.Aminobutyrate         7.39      18.17      20.70     6.30
## X2.Hydroxyisobutyrate   36.23      46.53      38.47    27.94
## X2.Oxoglutarate        75.94      81.45     164.02    24.05
## X3.Aminoisobutyrate      9.87      44.70     206.44    14.88
##                      PIF_171 NETCR_008_V1 NETCR_008_V2 NETL_017_V1
## X1.6.Anhydro.beta.D.glucose  64.07      32.46     113.30    22.20
## X1.Methylnicotinamide      6.42      14.01      43.38    20.70
## X2.Aminobutyrate         28.79       2.97       4.66     7.85
## X2.Hydroxyisobutyrate   18.92       5.16      27.11    19.69
## X2.Oxoglutarate        85.63       8.08      22.42    38.47
## X3.Aminoisobutyrate     31.82       5.99      27.11     9.30
##                      NETL_017_V2 NETL_002_V1 NETL_002_V2 PIF_190
## X1.6.Anhydro.beta.D.glucose  46.53     192.48     528.48    28.79
## X1.Methylnicotinamide      9.78     108.85     225.88     9.21
## X2.Aminobutyrate          3.10       7.77      13.46     5.53
## X2.Hydroxyisobutyrate      9.30      46.06      93.69    17.64
## X2.Oxoglutarate         10.59      55.15     230.44    14.44
## X3.Aminoisobutyrate      13.20       7.03      10.80    15.49
##                      NETCR_009_V1 NETCR_009_V2 NETL_007_V1 PIF_112
## X1.6.Anhydro.beta.D.glucose 181.27      47.47      15.96    22.87
## X1.Methylnicotinamide      48.42       7.69      16.12    10.38
## X2.Aminobutyrate           8.94       4.06       1.93     1.28
## X2.Hydroxyisobutyrate     51.94       9.30      15.80     5.58
## X2.Oxoglutarate         982.40      65.37      25.28     8.50
## X3.Aminoisobutyrate     198.34      50.40      13.46    13.74
##                      NETCR_019_V2 NETL_012_V1 NETL_012_V2 NETL_003_V1
## X1.6.Anhydro.beta.D.glucose  35.16      16.95       9.39    37.71
## X1.Methylnicotinamide      52.46      15.80      14.01    18.17
## X2.Aminobutyrate          13.87      10.49       5.16    26.05
## X2.Hydroxyisobutyrate     44.26      22.42      23.57    15.03
## X2.Oxoglutarate          99.48      62.80      46.99    23.34
## X3.Aminoisobutyrate     208.51      10.91      13.33    33.45
##                      NETL_003_V2
## X1.6.Anhydro.beta.D.glucose  38.47
## X1.Methylnicotinamide      12.55
## X2.Aminobutyrate          15.03
## X2.Hydroxyisobutyrate     12.55
## X2.Oxoglutarate          22.20
## X3.Aminoisobutyrate       21.33

```

```
rowData(se)
```

```
## DataFrame with 63 rows and 0 columns
```



```
colData(se)
```

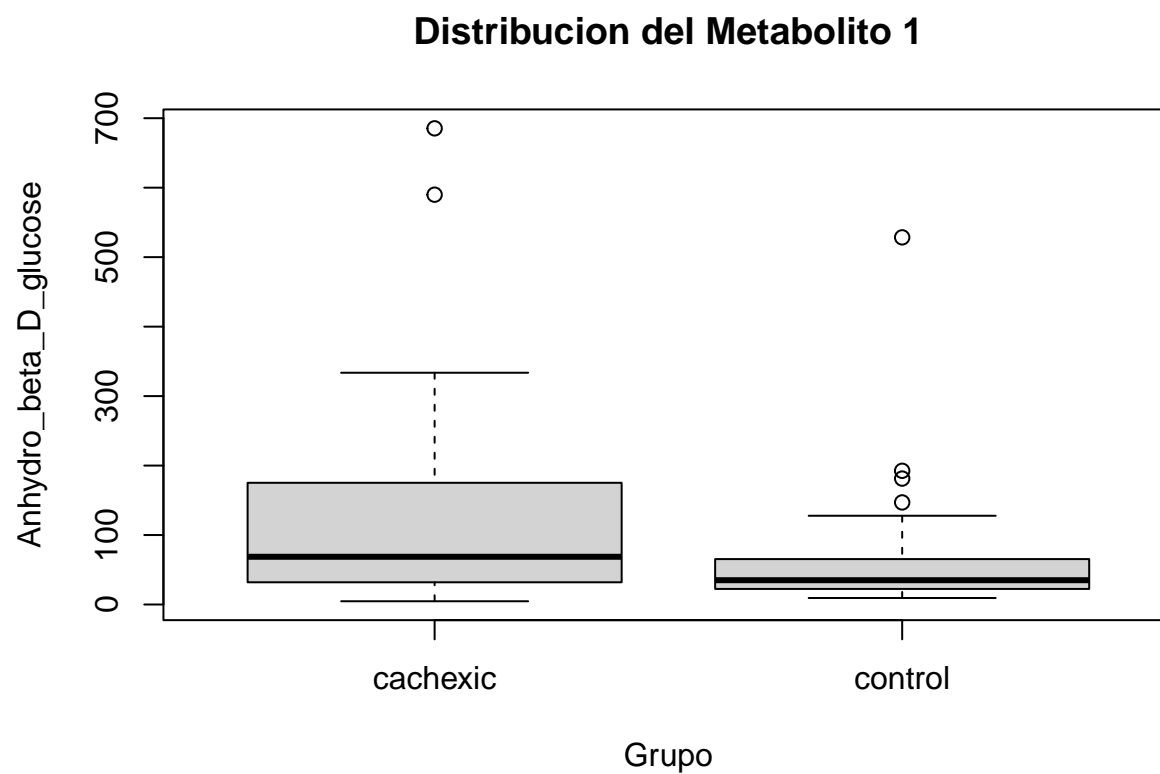
```
## DataFrame with 77 rows and 1 column
##           Grupo
##      <character>
## PIF_178      cachexic
## PIF_087      cachexic
## PIF_090      cachexic
## NETL_005_V1   cachexic
## PIF_115      cachexic
## ...          ...
## NETCR_019_V2  control
## NETL_012_V1   control
## NETL_012_V2   control
## NETL_003_V1   control
## NETL_003_V2   control
```

Exploración de los datos

Análisis univariante

Una vez extraídos los datos y vista su estructura, podemos hacer una exploración básica. Podemos ver su distribución mediante el uso de boxplots. Vamos a ver cómo se distribuyen los datos para el primer metabolito.

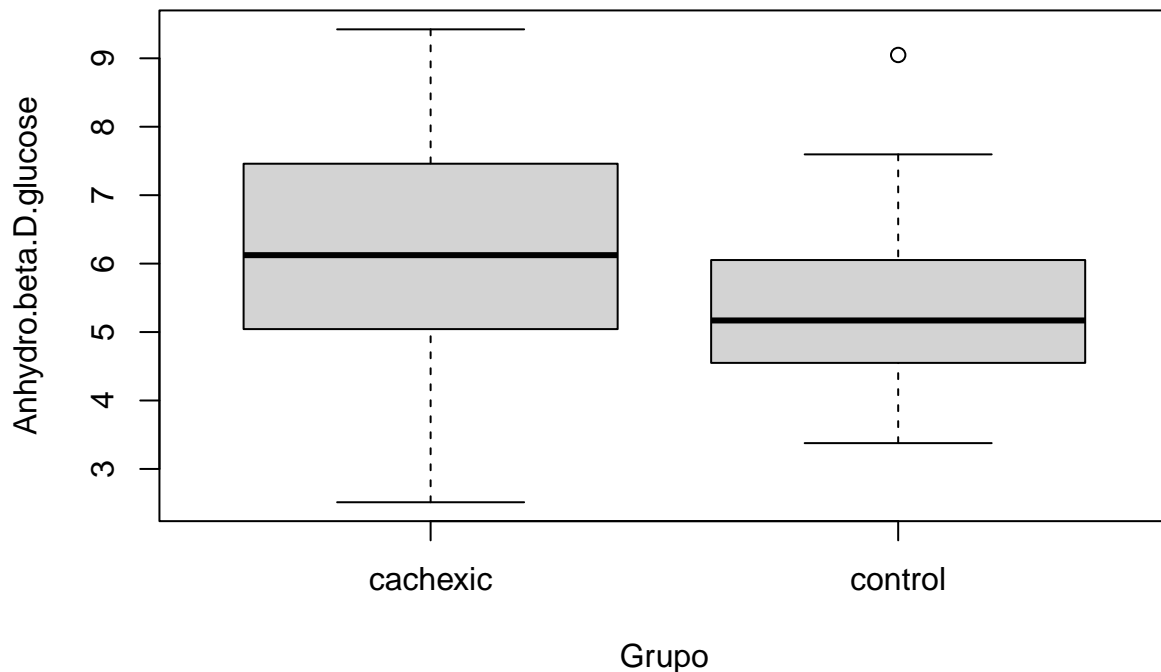
```
# Gráfico de caja para un metabolito en particular
boxplot(assay(se)[1, ] ~ colData(se)$Grupo,
        main = "Distribucion del Metabolito 1",
        xlab = "Grupo", ylab = "Anhydro_beta_D_glucose" )
```



Como los datos se ven muy asimétricos procedemos a tomar logaritmos

```
boxplot(log2(assay(se)[1, ] + 1)~colData(se)$Grupo,
        main = "Distribución del Metabolito 1",
        xlab = "Grupo", ylab = "Anhydro.beta.D.glucose")
```

Distribución del Metabolito 1



Vemos que es bastante mejor trabajar con los datos logarítmicos.

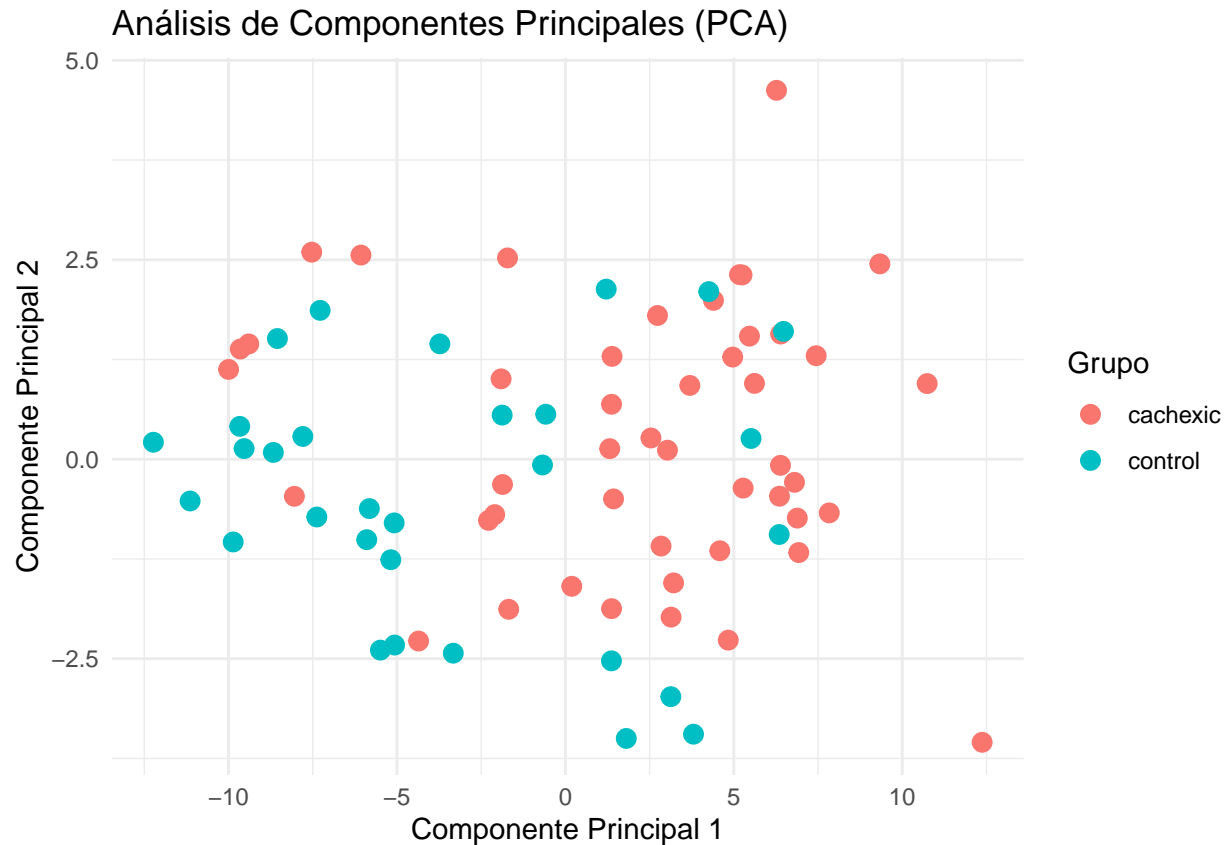
Análisis multivariante

Vamos a empezar realizando un análisis de componentes principales. Con este análisis podremos observar si existen algunos metabolitos que estén causando diferencias entre los dos grupos que tenemos de pacientes. Graficando los dos primeros componentes suele ser suficiente ya que toda la variación está explicada con ellos.

```
pca <- prcomp(t(log2(assay(se) + 1)), scale. = TRUE)
loads<- round(pca$sdev^2/sum(pca$sdev^2)*100,1)
```

Y graficar los dos primeros componentes

```
library(ggplot2)
df_pca <- data.frame(
  PC1 = pca$x[, 1],
  PC2 = pca$x[, 2],
  Grupo = colData(se)$Grupo
)
ggplot(df_pca, aes(x = PC1, y = PC2, colour = Grupo)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "Análisis de Componentes Principales (PCA)",
       x = "Componente Principal 1",
       y = "Componente Principal 2")
```



Vemos que hay ciertos pacientes que se ven alejados del grupo de pacientes control, lo cual puede indicar la actividad de metabolitos que estén causando esas diferencias.

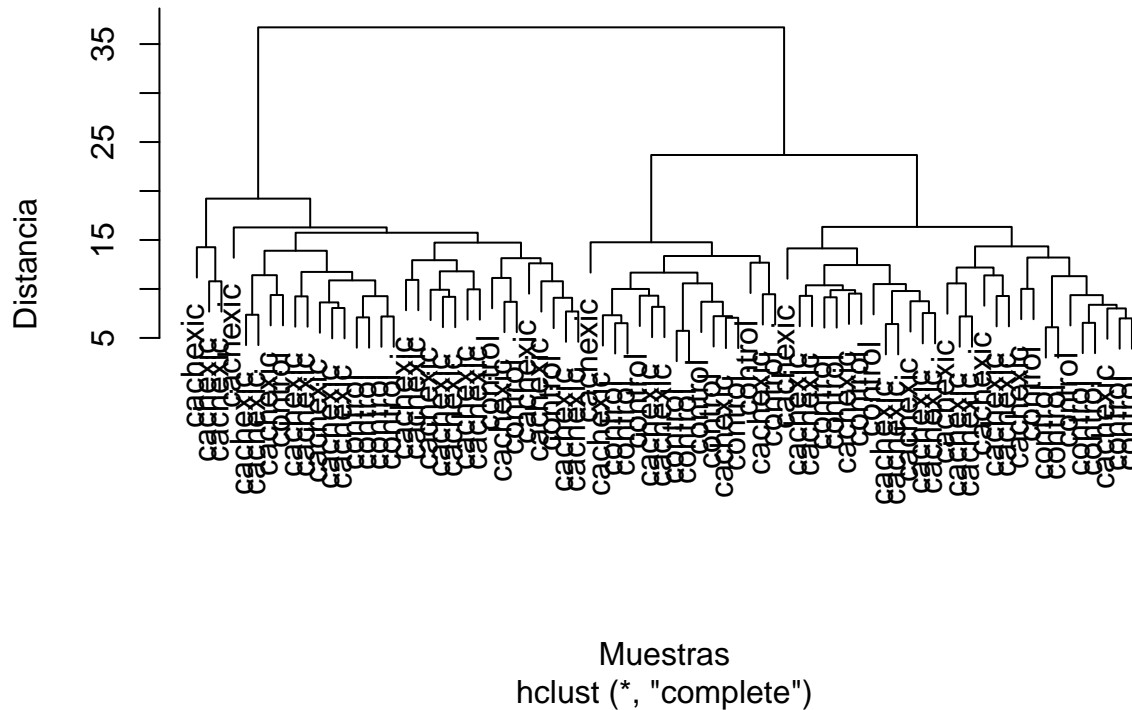
Ahora vamos a realizar un agrupamiento jerárquico por si podemos visualizar algún tipo de patrón

```
# Calcular la matriz de distancia Euclidiana
dist_matrix <- dist(t(log2(assay(se) + 1)), method = "euclidean")

# Agrupamiento jerárquico
hc <- hclust(dist_matrix, method = "complete")

# Dendrograma
plot(hc, labels = colData(se)$Grupo, main = "Agrupamiento Jerárquico",
     xlab = "Muestras", ylab = "Distancia")
```

Agrupamiento Jerárquico



```
clusters <- cutree(hc, k = 2) # seleccionamos un nivel de corte = 2
table(clusters)
```

```
## clusters
## 1 2
## 32 45
```

```
table(colData(se)$Grupo, clusters)
```

```
##          clusters
##          1  2
## cachexic 24 23
## control   8 22
```

Como el dendrograma está muy poblado es difícil ver patrones a simple vista. Al dividir en dos grupos vemos que hay las muestras se separan en 32 y 45. Y en el grupo de 32 vemos que 24 de ellos tienen la enfermedad. Esto al igual que el análisis de componentes principales muestra indicios de un perfil metabólico distinto entre individuos sanos y enfermos.

Discusión

En este trabajo hemos podido construir un *SummarizeExperiment* comprendiendo cómo se compone y cómo acceder a los datos a través de él. En cuanto a los datos, hemos podido observar que era mejor trabajar tomando logaritmos ya que en los boxplots que hemos realizado hemos podido observar asimetría que de no ser tenida en cuenta podría afectar a resultados posteriores o a la interpretación de los resultados. Por otro lado, gracias al análisis multivariante hemos podido ver indicios de diferencias metabólicas entre los dos grupos de pacientes, ya que en el análisis de componentes principales como en la agrupación jerárquica

hemos visto ciertos patrones diferentes entre los dos grupos de pacientes.

Conclusión

La programación orientada a objetos es una manera útil y organizada de tratar con grandes volúmenes de datos y proporciona muchas herramientas para analizarlos. Es importante hacer un análisis univariante de los datos para ver la dispersión y comprobar si es necesario hacer alguna transformación de estos. Gracias al análisis multivariante hemos podido ver indicios de perfiles metabólicos distintos entre los dos grupos de pacientes.

Bibliografía

- <https://github.com/alejandrogjuoc/Garcia-Jesus-Alejandro-PEC1>
- <https://bioconductor.org/packages/release/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html#constructing-a-summarizedexperiment>
- https://rest.xialab.ca/api/download/metaboanalyst/human_cachexia.csv