

PEC 1

**M0.157 - Anàlisi de
dades òmiques**

Informe de la pràctica

6 de Novembre de 2024
Jordi Llatser Torres

Objectius	3
Procés	3
Descàrrega de les dades	3
Creació Github	5
Codi R	6
Conclusions	11
Resultats	12

Objectius

L'objectiu d'aquesta pràctica és fer un anàlisi de dades òmiques, utilitzant el coneixement après en les activitats, mentre s'utilitzen dues eines molt comunes, com serien el cas del github i bioconductor. Aquest és l'informe demanat pel punt 4 de la pràctica.

Procés

Per començar aquesta pràctica hi ha dos punts d'inici, descarregar la base de dades, i preparar el repositori github on es penjarà totes les coses demanades en la pràctica, seguint l'enunciat.

Descàrrega de les dades

El pas de descarregar les dades ha estat senzill. Per fer-ho, primer he accedit al github on podem obtenir les dades, seguint l'enllaç del pdf.

The screenshot shows the GitHub interface for the repository 'nutrimetabolomics / metaboData'. The repository is public and has 1 branch (main) and 15 commits. The commit history table is as follows:

File	Commit Message	Time Ago
Datasets	Update repository	4 days ago
.gitignore	Initial commit	7 months ago
2024-metaboData.Rproj	first commit	7 months ago
Data_Catalog.xlsx	Added Cachexia dataset	5 months ago
LICENSE	Initial commit	7 months ago
README.html	Already (only) two datasets :-{	7 months ago
README.md	Added information on datasets in Data_Catalog.xls	5 months ago

Below the commit history, the README file is selected, showing the following content:

metaboData

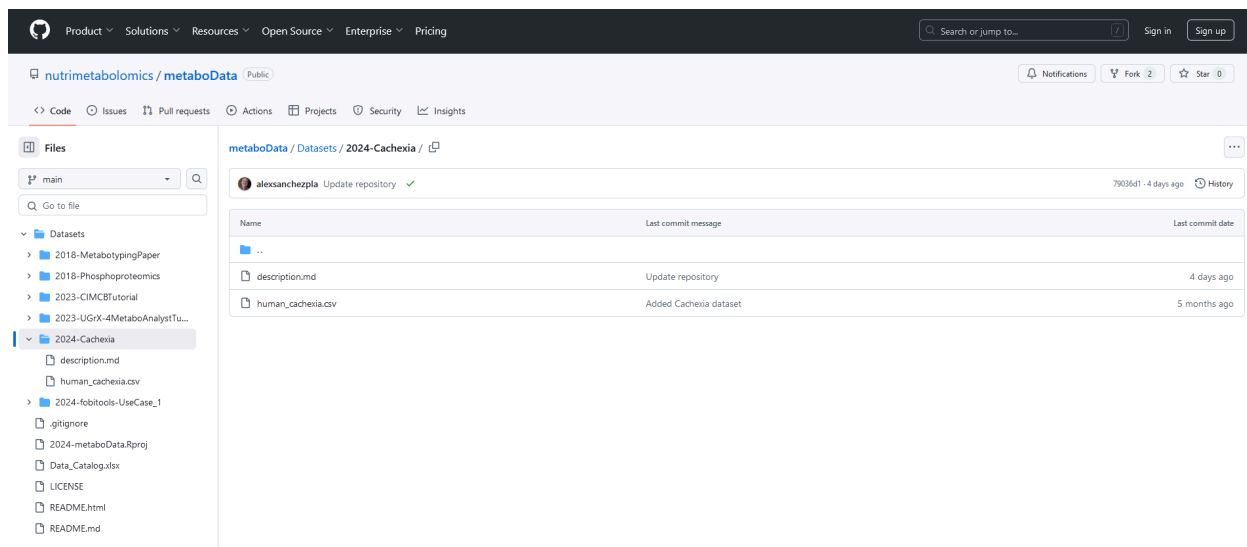
Introduction

A repository with a few public metabolomics datasets borrowed from different public open sources.

While we don't come out with a better option the repository will be "folder-based". That is:

- Each dataset is contained in a sub-folder of the "datasets" folder named with a short-descriptive name with no spaces and no special codes.

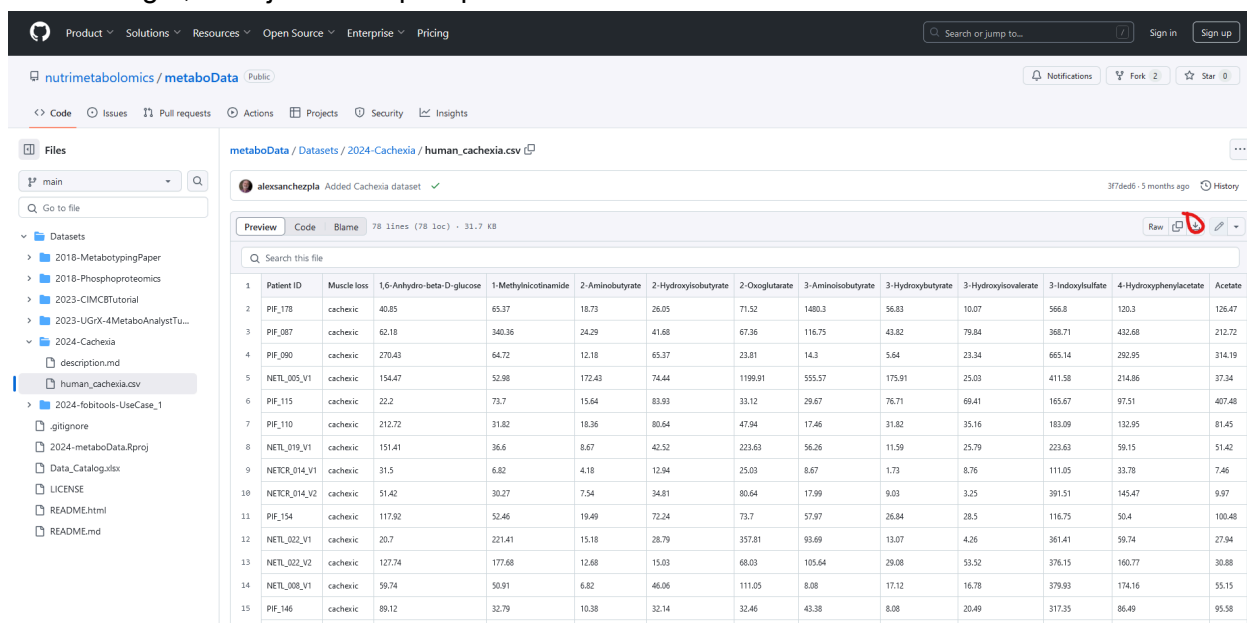
Un cop aquí dintre, per accedir als datasets s'ha de clicar a la carpeta anomenada Datasets, i fent que s'obri la carpeta, i a dintre es poden veure més carpetes que contenen les bases de dades. Jo he decidit accedir a la base de dades de Cachexia, i altre cop clicant a la carpeta corresponent, se m'ha obert permetent veure aquestes dades:



The screenshot shows the GitHub repository page for `nutrimetabolomics/metaboData`. The left sidebar displays the file structure, with the `2024-Cachexia` folder selected. The main content area shows the commit history for the `2024-Cachexia` folder, with the latest commit by `alexsanchezpla` titled "Update repository" and "Added Cachexia dataset".

Name	Last commit message	Last commit date
..		
description.md	Update repository	4 days ago
human_cachexia.csv	Added Cachexia dataset	5 months ago

Un cop aquí dintre, per descarregar les dades, he clicat dintre del fitxer .csv, que són les dades a descarregar, i allà ja hi ha l'opció per baixar el fitxer en l'ordinador de forma local.



The screenshot shows the GitHub repository page for `nutrimetabolomics/metaboData`, specifically the `human_cachexia.csv` file. The file is displayed in a table format, showing patient data and various metabolite levels. The table has 13 columns: Patient ID, Muscle loss, 1,6-Anhydro-beta-D-glucose, 1-Methylcarnitine, 2-Aminobutyrate, 2-Hydroxyisobutyrate, 2-Oxoglutarate, 3-Aminoisobutyrate, 3-Hydroxybutyrate, 3-Hydroxyisovalerate, 3-Indoxylsulfate, 4-Hydroxyphenylacetate, and Acetate. The table contains 15 rows of data, with the first row being the header.

Patient ID	Muscle loss	1,6-Anhydro-beta-D-glucose	1-Methylcarnitine	2-Aminobutyrate	2-Hydroxyisobutyrate	2-Oxoglutarate	3-Aminoisobutyrate	3-Hydroxybutyrate	3-Hydroxyisovalerate	3-Indoxylsulfate	4-Hydroxyphenylacetate	Acetate
PIF_178	cachexic	40.85	65.37	18.73	26.05	71.52	1480.3	56.83	10.07	566.8	120.3	126.47
PIF_087	cachexic	62.18	340.36	24.29	41.68	67.36	116.75	43.82	79.84	368.71	432.68	212.72
PIF_090	cachexic	270.43	64.72	12.18	65.37	23.81	14.3	5.64	23.34	665.14	250.95	314.19
NETL_005_V1	cachexic	154.47	52.98	172.43	74.44	1199.91	555.57	175.91	25.03	411.58	214.86	37.34
PIF_115	cachexic	22.2	73.7	15.64	83.93	33.12	29.67	76.71	69.41	165.67	97.51	407.48
PIF_110	cachexic	212.72	31.82	18.36	80.64	47.94	17.46	31.82	35.16	183.09	132.95	81.45
NETL_019_V1	cachexic	151.41	36.6	8.67	42.52	223.63	56.26	11.59	25.79	223.63	56.15	51.42
NETCR_014_V1	cachexic	31.5	6.82	4.18	12.94	25.03	8.67	1.73	8.76	111.05	33.78	7.46
NETCR_014_V2	cachexic	51.42	30.27	7.54	34.81	80.64	17.99	9.03	3.25	391.51	145.47	9.97
PIF_154	cachexic	117.82	52.46	19.49	72.24	73.7	57.97	26.84	28.5	116.75	50.4	100.48
NETL_002_V1	cachexic	20.7	221.41	15.18	28.79	357.81	93.69	13.07	4.26	361.41	58.74	27.94
NETL_002_V2	cachexic	127.74	177.68	12.68	15.03	68.03	105.64	29.08	53.52	376.15	160.77	30.88
NETL_008_V1	cachexic	59.74	50.91	6.82	46.06	111.05	8.08	17.12	16.78	379.93	174.16	55.15
PIF_146	cachexic	89.12	32.79	10.38	32.14	32.46	43.38	8.08	20.49	317.35	86.49	93.58

Creació Github

El següent pas és preparar el repositori de github on es penjaran els resultats. Per fer això, primer he entrat amb la meua conta de github, i des de dintre el propi dashboard, ja em sortia l'opció de crear un repositori, i des d'allà li he posat el nom demanat en l'enunciat. Un cop fet això, he seguit les comandes que sortien allà per crear el repositori en local i començar a penjar documents.

Primer amb la comanda

```
git init
```

per crear un repositori en local, i després amb la comanda

```
git add human_cachexia.csv
```

per afegir el fitxer amb la base de dades que havia baixat. Un cop fet això, he utilitzat les comandes següents per fer el commit, i vincular-ho amb el repositori que havia creat desde la pàgina web:

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/Jollito/test.git
```

```
git push -u origin main
```

Per fer més commits en el futur, però, he utilitzat Visual Studio Code, que ja tinc vinculat amb github, i és una eina més pràctica per fer els commits i els pushes de codi.

Codi R

Per treballar amb les dades, he utilitzat RStudio, primer creant un script d'R, on executaré el codi.

Les tasques a fer a partir d'aquest punt són

- Guardar les dades en un SummarizedExperiment
- Descarregar l'objecte contenidor en format binari d'R
- Tenir les dades en format text
- Tenir les metadades en format markdown
- Fer un anàlisi de les dades.

He decidit deixar l'anàlisi de les dades pel final, ja que és el pas més lliure, i començar per les altres tasques

Per poder guardar les dades en el format SummarizedExperiment, primer s'ha d'instal·lar la llibreria corresponent. Per fer això, s'utilitzen aquestes comandes de R:

```
1 if (!require("BiocManager", quietly = TRUE)){  
2   install.packages("BiocManager")  
3   BiocManager::install("SummarizedExperiment")  
4 }  
5  
6 library(SummarizedExperiment)
```

Amb això, s'aconsegueix instal·lar la llibreria BiocManager, que ens permet accedir a les eines de bioconductor, i així poder també baixar la llibreria SummarizedExperiment i activar-la. Un cop fet això, ja podem crear objectes SummarizedExperiment per poder fer els anàlisis corresponents.

Un cop fet això, fa falta llegir les dades i fer un primer anàlisi, per saber com haurem d'emmagatzemar les dades. Per llegir les dades, és tan senzill com utilitzar la funció `read.csv`, i així podem carregar les dades. Tot seguit, amb l'ajuda de `View`, podem veure les dades. Al analitzar-les, veiem que hi ha dos columnes amb dades no-numèriques, i la resta de columnes sí, fent referència a diferents metabòlits i compostos analitzats en l'estudi. I de mentres, les dades no-numèriques fan referència al id del pacient, i si tenen o no pèrdua muscular a causa de la caquèxia.

Amb aquest primer cop d'ull a les dades, i mirant en com es preparen les dades per guardar-les en un `summarized_experiment`, he decidit separar les dos primeres columnes referents a la informació de l'id del pacient i si el pacient té caquèxia o no, i les dades numèriques com a la informació a passar-li al Summarized Experiment. Per fer això, he separat les dades de la següent forma, traient també el nom de les columnes per guardar-ho com a informació.

```
# Preparar dades per SummarizedExperiment
row_data <- dades[, c("Patient.ID", "Muscle.loss")]
assay_data <- as.matrix(dades[, -c(1, 2)]) # Excloem les dues primeres columnes, ja estan a row_data
col_data <- data.frame(Compound = colnames(assay_data)) # Metadades de les columnes

# Creació de l'objecte SummarizedExperiment
se <- SummarizedExperiment(
  assays = list(counts = assay_data),
  rowData = row_data,
  colData = col_data
)
```

Referent a les metadades, sobre aquest fitxer no he trobat cap informació extra a afegir, i per això només hi ha la informació referent als components de les columnes, i la dels pacients i el seu estat mèdic, afegits al Summarized Experiment com a rowData i colData.

Per descarregar l'objecte del SummarizedExperiment en forma binària és molt senzill, utilitzant la comanda save i passant-li l'objecte com a paràmetre, es pot guardar:

```
save(se, file = "human_cachexia_se.Rda")
```

En el meu cas, les dades ja venen en format de text, en format csv (Comma separated values). Però per fer-ho igualment, he decidit tornar a descarregar les dades, encara que ara només les guardades en l'assays del objecte Summarized Experiment. I per poder fer això, hi ha la funció write.csv, que utilitzada de la següent forma permet descarregar les dades:

```
# Exportar dades del SummarizedExperiment
write.csv(as.data.frame(assays(se)$counts), "human_cachexia_data.csv", row.names = FALSE)
```

Per tenir les metadades en markdown, el que s'ha de fer és crear un fitxer markdown i escriure les dades, i per això no serà discutit aquí com s'ha fet.

Per tal de fer un bon anàlisi, primer hem d'observar les dades i veure quin tractament necessiten. En el cas d'aquestes dades, també m'interessa veure si hi ha la mateixa quantitat de mostres amb caquèxia i sense. Per fer això, he fet un summary de les dades, per veure com estan repartides, i també un petit conteig per veure com estan repartides les mostres:

```
> # Exploració de les dades
> table(rowData(se)$'Muscle.loss') # Comptar quants pacients són cachectics o no
```

cachexic	control
47	30

```
> summary(assays(se)$counts) # Resum de les concentracions de compostos
```

X1.6.Anhydro.beta.D.glucose	X1.Methylnicotinamide	X2.Aminobutyrate	X2.Hydroxyisobutyrate	X2.Oxoglutarate
Min. : 4.71	Min. : 6.42	Min. : 1.28	Min. : 4.85	Min. : 5.53
1st Qu.: 28.79	1st Qu.: 15.80	1st Qu.: 5.26	1st Qu.: 15.80	1st Qu.: 22.42
Median : 45.60	Median : 36.60	Median : 10.49	Median : 32.46	Median : 55.15
Mean : 105.63	Mean : 71.57	Mean : 18.16	Mean : 37.25	Mean : 145.09
3rd Qu.: 141.17	3rd Qu.: 73.70	3rd Qu.: 19.49	3rd Qu.: 54.60	3rd Qu.: 92.76
Max. : 685.40	Max. : 1032.77	Max. : 172.43	Max. : 93.69	Max. : 2465.13

X3.Aminoisobutyrate	X3.Hydroxybutyrate	X3.Hydroxyisovalerate	X3.Indoxylsulfate	X4.Hydroxyphenylacetate
Min. : 2.61	Min. : 1.70	Min. : 0.92	Min. : 27.66	Min. : 15.49
1st Qu.: 11.70	1st Qu.: 5.99	1st Qu.: 5.26	1st Qu.: 82.27	1st Qu.: 41.68
Median : 22.65	Median : 11.70	Median : 12.55	Median : 144.03	Median : 70.11
Mean : 76.76	Mean : 21.72	Mean : 21.65	Mean : 218.88	Mean : 112.02
3rd Qu.: 56.26	3rd Qu.: 29.96	3rd Qu.: 30.27	3rd Qu.: 333.62	3rd Qu.: 145.47
Max. : 1480.30	Max. : 175.91	Max. : 164.02	Max. : 1043.15	Max. : 796.32

En aquests resultats, es pot veure que hi ha més mostres de caquèxia que de control, però com estan repartides i amb % parts un, i % parts l'altre, no hauria d'haver-hi problemes per les mostres. Si ens fixem en la informació del summary (a la captura no surt tota la informació tornada, només una mostra), podem veure que els valors estan força repartits, però que hi ha a vegades valors màxims que poden afectar els resultats, i s'hauria de treure primer els outliers. Una de les tècniques que podem utilitzar és aplicar una transformació logarímic de les dades, que quedaria així:

```
> assays(se)$log_counts <- log(assays(se)$counts + 1)
> summary(assays(se)$log_counts)
```

X1.6.Anhydro.beta.D.glucose	X1.Methylnicotinamide	X2.Aminobutyrate	X2.Hydroxyisobutyrate	X2.Oxoglutarate
Min. :1.742	Min. :2.004	Min. :0.8242	Min. :1.766	Min. :1.876
1st Qu.:3.394	1st Qu.:2.821	1st Qu.:1.8342	1st Qu.:2.821	1st Qu.:3.154
Median :3.842	Median :3.627	Median :2.4415	Median :3.510	Median :4.028
Mean :4.113	Mean :3.657	Mean :2.5040	Mean :3.407	Mean :3.983
3rd Qu.:4.957	3rd Qu.:4.313	3rd Qu.:3.0199	3rd Qu.:4.018	3rd Qu.:4.541
Max. :6.531	Max. :6.941	Max. :5.1558	Max. :4.551	Max. :7.810

X3.Aminoisobutyrate	X3.Hydroxybutyrate	X3.Hydroxyisovalerate	X3.Indoxylsulfate	X4.Hydroxyphenylacetate
Min. :1.284	Min. :0.9933	Min. :0.6523	Min. :3.356	Min. :2.803
1st Qu.:2.542	1st Qu.:1.9445	1st Qu.:1.8342	1st Qu.:4.422	1st Qu.:3.754
Median :3.163	Median :2.5416	Median :2.6064	Median :4.977	Median :4.264
Mean :3.381	Mean :2.6634	Mean :2.6417	Mean :5.029	Mean :4.356
3rd Qu.:4.048	3rd Qu.:3.4327	3rd Qu.:3.4427	3rd Qu.:5.813	3rd Qu.:4.987
Max. :7.301	Max. :5.1756	Max. :5.1061	Max. :6.951	Max. :6.681

I després d'aplicar un escalat, tenim les dades a punt per treballar:

```
> assays(se)$exploration_counts <- scale(assays(se)$log_counts)
> summary(assays(se)$exploration_counts)
```

X1.6.Anhydro.beta.D.glucose	X1.Methylnicotinamide	X2.Aminobutyrate	X2.Hydroxyisobutyrate	X2.Oxoglutarate
Min. :-2.2678	Min. :-1.60719	Min. :-1.93485	Min. :-2.2069	Min. :-1.65071
1st Qu.: -0.6878	1st Qu.: -0.81255	1st Qu.: -0.77152	1st Qu.: -0.7879	1st Qu.: -0.64972
Median : -0.2598	Median : -0.02916	Median : -0.07203	Median : 0.1388	Median : 0.03562
Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000
3rd Qu.: 0.8070	3rd Qu.: 0.63837	3rd Qu.: 0.59425	3rd Qu.: 0.8218	3rd Qu.: 0.43745
Max. : 2.3129	Max. : 3.19334	Max. : 3.05432	Max. : 1.5380	Max. : 3.00005

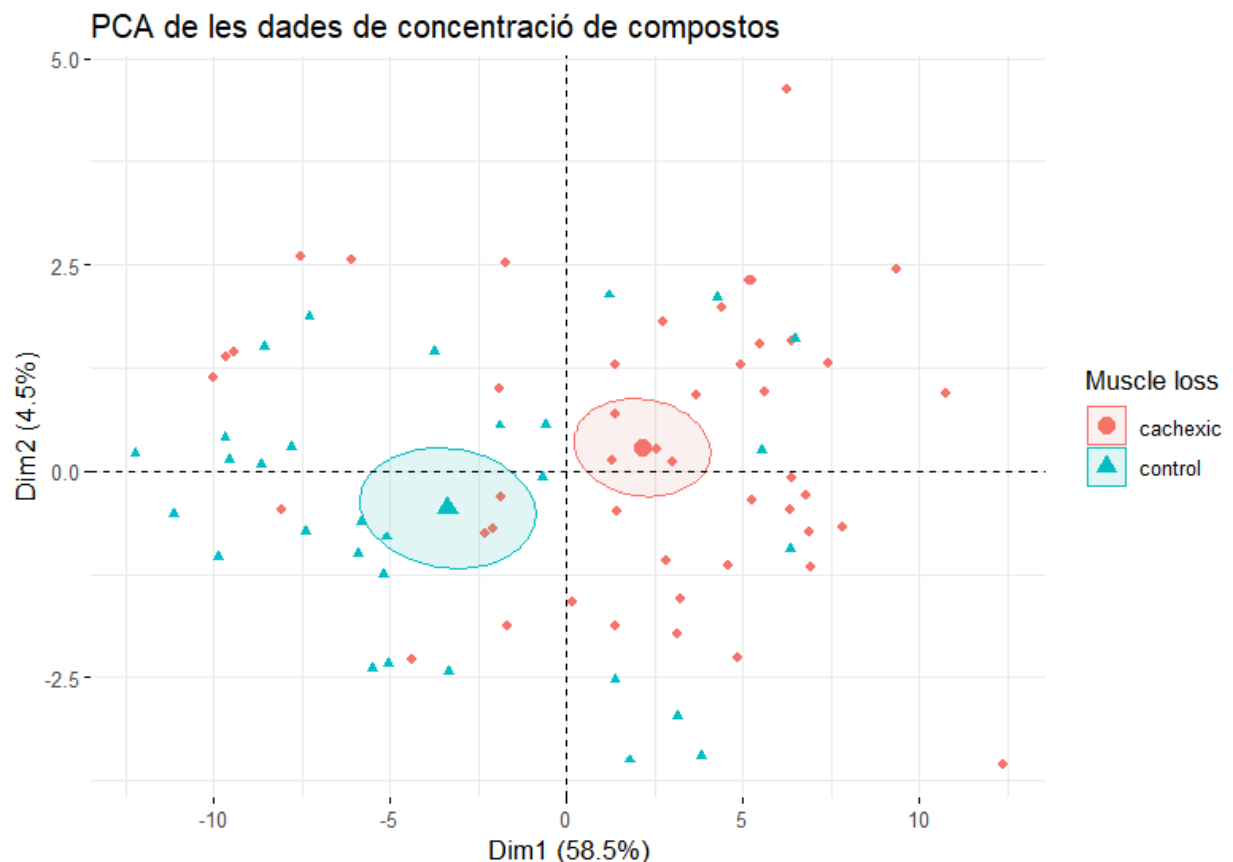
X3.Aminoisobutyrate	X3.Hydroxybutyrate	X3.Hydroxyisovalerate	X3.Indoxylsulfate	X4.Hydroxyphenylacetate
Min. :-1.7064	Min. :-1.7612	Min. :-2.00456	Min. :-1.91297	Min. :-1.8599
1st Qu.: -0.6831	1st Qu.: -0.7581	1st Qu.: -0.81370	1st Qu.: -0.69388	1st Qu.: -0.7210
Median : -0.1772	Median : -0.1285	Median : -0.03562	Median : -0.05969	Median : -0.1097
Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000
3rd Qu.: 0.5421	3rd Qu.: 0.8112	3rd Qu.: 0.80702	3rd Qu.: 0.89590	3rd Qu.: 0.7557
Max. : 3.1886	Max. : 2.6491	Max. : 2.48310	Max. : 2.19658	Max. : 2.7849

Acetate	Acetone	Adipate	Alanine	Asparagine	Betaine
Min. :-2.03090	Min. :-1.5368	Min. :-1.6920	Min. :-2.2222	Min. :-2.08279	Min. :-2.6680
1st Qu.: -0.76932	1st Qu.: -0.6125	1st Qu.: -0.6400	1st Qu.: -0.7690	1st Qu.: -0.87186	1st Qu.: -0.6164
Median : 0.03148	Median : -0.1313	Median : -0.1756	Median : 0.1085	Median : -0.05181	Median : 0.1203
Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000	Mean : 0.00000
3rd Qu.: 0.74903	3rd Qu.: 0.4141	3rd Qu.: 0.4268	3rd Qu.: 0.8059	3rd Qu.: 0.81735	3rd Qu.: 0.7463
Max. : 2.20084	Max. : 4.9278	Max. : 3.2912	Max. : 1.9613	Max. : 2.12823	Max. : 1.7843

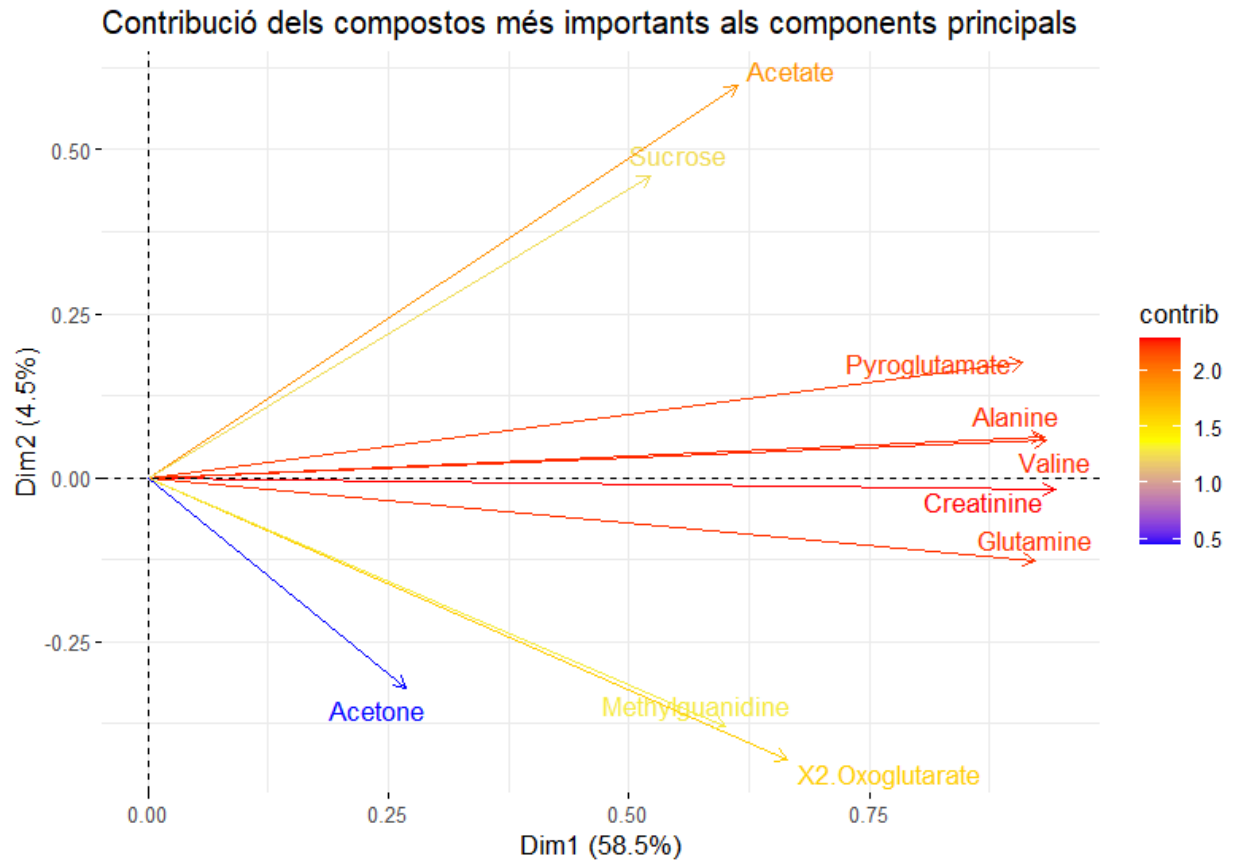
Un dels possibles estudis que podem fer amb les dades, és aplicar el Principal Component Analysis, conegut com a PCA. Per fer aquesta anàlisi, podem fer ús de la llibreria FactoMineR, que té una funció encarregada d'aplicar aquesta anàlisi. Aquesta funció torna 63 components, un per cada columna de dades que teníem, i si comparem els dos primers amb un gràfic

(utilitzant una funció de factoextra) veiem aquest resultat:

```
dades_pca <- princomp(assays(se)$exploration_counts)
fviz_pca_ind(dades_pca,
  geom.ind = "point",
  col.ind = rowData(se)$'Muscle.loss', # Color per condició de pèrdua muscular
  addEllipses = TRUE,
  ellipse.type = "confidence",
  legend.title = "Muscle loss") +
labs(title = "PCA de les dades de concentració de compostos")
```



Ens podem fixar que els components principals no donen massa confiança per classificar les variables, fixant-nos amb les el·lipses de la confiança, encara que sí que podem veure una tendència en la separació dels grups. Un altre detall que podem veure, és que el primer component principal explica un 58,5% de les dades, mentre que el segon només explica un 4,5% de les dades, indicant que l'únic grup rellevant per indicar una separació és el primer. Per poder veure quines variables són les més rellevants, he agafat les 5 variables que més contribueixen als dos primers components principals, obtenint aquest gràfic, on podem veure com contribueixen:



I podem veure com les variables que contribueixen més al component principal, contribueixen molt més que les que contribueixen al segon component, on per exemple el 5è element que més contribueix, l'acetona, està indicat que no contribueix significativament.

Aquestes dades ens poden servir per calcular un model de regressió logarítmic per exemple, que utilitzant les 5 variables que més contribueixen del primer component principal, obtenim el següent resultat:

```
Call:
glm(formula = condition ~ ., family = binomial, data = data_model)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9590  -0.9478   0.3305   0.8165   1.5969

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.260e+00  5.016e-01  -2.511   0.012 *
Creatinine   -8.852e-05  1.290e-04  -0.686   0.493
Valine        5.452e-02  3.362e-02   1.622   0.105
Alanine      -3.514e-03  4.011e-03  -0.876   0.381
Glutamine     2.328e-03  2.899e-03   0.803   0.422
Pyroglutamate 5.633e-03  4.290e-03   1.313   0.189
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 102.96  on 76  degrees of freedom
Residual deviance:  81.37  on 71  degrees of freedom
AIC: 93.37
```

Es pot observar que cap de les variables indica que té significança, sent això un mal resultat, i per tant s'hauria de provar amb altres variables, però si ho provem amb les variables del segon component principal, observem en aquest cas que si hi ha una variable significant en l'acetat:

```
Call:
glm(formula = condition ~ ., family = binomial, data = data_model)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.7340	-1.0290	0.4298	0.9640	1.5042

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.038944	0.661491	-1.571	0.116
Acetate	0.015695	0.007096	2.212	0.027 *
Sucrose	0.002039	0.002329	0.875	0.381
X2.Oxoglutarate	0.001129	0.001500	0.752	0.452
Methylguanidine	0.008943	0.020522	0.436	0.663
Acetone	0.024260	0.047246	0.513	0.608

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 102.960 on 76 degrees of freedom
Residual deviance: 87.209 on 71 degrees of freedom
AIC: 99.209

Number of Fisher Scoring iterations: 6

Amb un estudi d'aquest estil, i triant millor les variables, es pot acabar definint un model que ajudés a predir millor si un pacient té o no caquexia.

Un cop l'anàlisi de les dades ja s'ha fet, és quan he posat d'executar les funcions de guardar l'objecte, per tal de tenir la versió final obtinguda durant la pràctica.

Conclusions

Com s'ha pogut observar, amb l'anàlisi PCA dona la impressió que es pot fer una classificació de les dades, però com s'ha observat amb el model lineal fet a continuació, els resultats obtinguts encara necessitarien més treball perquè fossin significatius, demostrant les limitacions d'aquesta tècnica.

Resultats

Els resultats d'aquesta pràctica, així com l'informe, es poden trobar en el github d'aquest enllaç:

<https://github.com/Jollto/Llatser-Torres-Jordi-PEC1>