

Análisis de datos ómicos

PEC1

Lorca Orloff, Pablo

Índice

1	Resumen	1
2	Objetivos	1
3	Métodos	1
3.1	Obtención set de datos	1
3.2	Obtención set de datos	2
3.3	Análisis de datos	2
4	Resultados	2
4.1	Cargar data	2
4.2	Exploración de la data	2
4.3	Generación de SummarizedExperiment	3
4.4	Análisis de la data	3
5	Discusión	5
6	Conclusiones	5
7	Referencias	6
8	Anexo	7

1 Resumen

a

2 Objetivos

Los objetivos de este trabajo son los siguientes:

1. Analizar de manera exploratoria un conjunto de datos ómicos.
2. Crear y utilizar un objeto de la clase `SummarizedExperiment`.

3 Métodos

3.1 Obtención set de datos

Los datos utilizados en este trabajo se obtuvieron del repositorio de *Github* de *nutrimetabolomics* Nutrimetabolomics (2025). Se escogió el set de datos referente a la enfermedad caquexia (2024-Cachexia).

3.2 Obtención set de datos

Para almacenar y contener la data y metadata a trabajar, se generó un objeto SummarizedExperiment (Morgan et al. 2024).

3.3 Análisis de datos

Para un análisis estadístico de los datos, se visualizaron a través de gráficos de cajas e histogramas, además de realizar una prueba de Shapiro-Wilks. La correlación entre los metabolitos se analizó mediante la función cor.

4 Resultados

4.1 Cargar data

Se procede a cargar la data con de human_cachexia y la metadata.

```
#Cargar archivos data y metadata
data <- read.csv('human_cachexia.csv', check.names = FALSE)
data_info <- openxlsx::read.xlsx("Data_Catalog.xlsx")
```

4.2 Exploración de la data

Se procede a realizar una primera exploración de los datos.

```
# Dimensión
dim(data)
```

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

```
# Nombre variables
colnames(data)
```

```
## [1] "Patient ID" "Muscle loss" "1,6-Anhydro-beta-D-glucose"
## [4] "1-Methylnicotinamide" "2-Aminobutyrate" "2-Hydroxyisobutyrate"
## [7] "2-Oxoglutarate" "3-Aminoisobutyrate" "3-Hydroxybutyrate"
## [10] "3-Hydroxyisovalerate" "3-Indoxylsulfate" "4-Hydroxyphenylacetate"
## [13] "Acetate" "Acetone" "Adipate"
## [16] "Alanine" "Asparagine" "Betaine"
## [19] "Carnitine" "Citrate" "Creatine"
## [22] "Creatinine" "Dimethylamine" "Ethanolamine"
## [25] "Formate" "Fucose" "Fumarate"
## [28] "Glucose" "Glutamine" "Glycine"
## [31] "Glycolate" "Guanidoacetate" "Hippurate"
## [34] "Histidine" "Hypoxanthine" "Isoleucine"
## [37] "Lactate" "Leucine" "Lysine"
## [40] "Methylamine" "Methylguanidine" "N,N-Dimethylglycine"
## [43] "O-Acetylcarnitine" "Pantothenate" "Pyroglutamate"
## [46] "Pyruvate" "Quinolate" "Serine"
## [49] "Succinate" "Sucrose" "Tartrate"
```

```
## [52] "Taurine"                "Threonine"                "Trigonelline"
## [55] "Trimethylamine N-oxide" "Tryptophan"               "Tyrosine"
## [58] "Uracil"                 "Valine"                   "Xylose"
## [61] "cis-Aconitane"          "myo-Inositol"             "trans-Aconitane"
## [64] "pi-Methylhistidine"     "tau-Methylhistidine"
```

```
# Presencia datos faltantes
any(is.na(data))
```

```
## [1] FALSE
```

```
#Proporción Caquexia/Control
table(data$`Muscle loss`)
```

```
##
## cachexic   control
##        47        30
```

La data consta de 77 observaciones y 65 variables sin datos faltantes. De las 65 variables, la primera corresponde al identificador del paciente, la segunda corresponde si el paciente presenta caquexia o pertenece al grupo control, y luego los 63 metabolitos medidos.

4.3 Generación de SummarizedExperiment

Se generó el objeto SummarizedExperiment considerando las primeras dos columnas, Patient ID y Muscle loss, como información descriptiva (col_data), por otra parte, los valores de los metabolitos se consideraron para la matriz de expresión (assays).

```
# Matriz de expresión - Sin 'Patient ID' y 'Muscle loss'
exprs <- as.matrix(data[, -(1:2)])

# Extraer componentes
assay_data <- t(as.matrix(data[, -c(1, 2)])) # Matriz transpuesta de metabolitos
row_data <- Dataframe(metabolite = colnames(data)[-c(1, 2)]) # Nombres metabolitos
col_data <- data.frame(`Muscle loss` = data$`Muscle loss`) # Info descriptiva
rownames(col_data) <- data$`Patient ID` # IDs pacientes - nombres de fila
metadata_list <- list(DataInfo = data_info[6,]) # Metadata

# Objeto SummarizedExperiment
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(metabolites = assay_data),
  rowData = row_data,
  colData = col_data,
  metadata = metadata_list
)
```

4.4 Análisis de la data

4.4.1 Valores atípicos

Al agrupar y analizar los datos según su condición Muscle.loss se observa que, en promedio, los metabolitos de los pacientes con caquexia son más altos que los pacientes control. Adicionalmente, todas las muestras presentan datos estadísticamente anómalos (Ver Figura Anexo 1).

4.4.2 Distribución

La distribución que presentan los datos de cada metabolito según su condición `Muscle.loss`, por lo general, no muestran presentar una distribución normal (Ver Figura Anexo 2), lo cual se corrobora al realizar un test de Shapiro-Wilks (Ver Anexo 3), en donde en ningún caso hay evidencia para confirmar normalidad.

4.4.3 Correlación

```
# Matriz de correlación
m_cor <- cor(t(assay(se)))

# Elimina parte inferior
m_cor[lower.tri(m_cor)] <- NA

# Límite cor 0.7 - Excluye diagonal
sum(abs(m_cor) > 0.7 & abs(m_cor) != 1, na.rm = TRUE)
```

```
## [1] 108
```

```
# Límite cor 0.3 - Excluye diagonal
sum(abs(m_cor) > 0.3 & abs(m_cor) != 1, na.rm = TRUE)
```

```
## [1] 1140
```

Al evaluar las posibles correlaciones entre metabolitos, se encontró que 1140 parejas de metabolitos podrían presentar algún grado de correlación lineal significativa, y 140 presentarían una correlación lineal fuerte.

4.4.4 Análisis de componentes principales

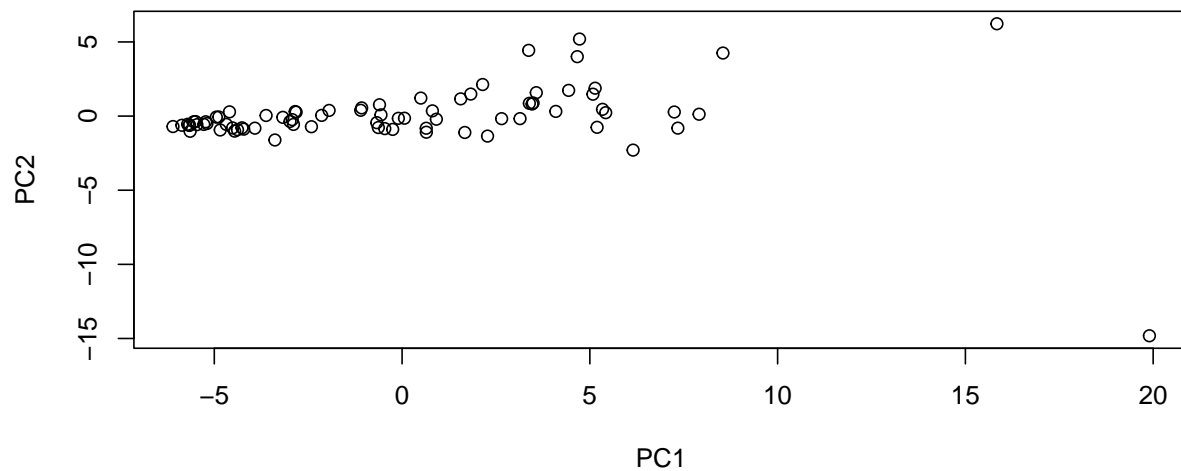
```
# PCA
pca <- prcomp(t(assay(se)), scale. = TRUE)

# Ver resultados
summary(pca)
```

```
## Importance of components:
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8      PC9
## Standard deviation  5.0467 2.2701 1.83311 1.74728 1.65906 1.6130 1.47304 1.36403 1.24275
## Proportion of Variance 0.4043 0.0818 0.05334 0.04846 0.04369 0.0413 0.03444 0.02953 0.02451
## Cumulative Proportion 0.4043 0.4861 0.53941 0.58787 0.63156 0.6729 0.70730 0.73683 0.76135
##              PC10     PC11     PC12     PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation  1.20650 1.1584 1.05503 1.03620 0.9914 0.96773 0.89551 0.86788 0.83041
## Proportion of Variance 0.02311 0.0213 0.01767 0.01704 0.0156 0.01487 0.01273 0.01196 0.01095
## Cumulative Proportion 0.78445 0.8057 0.82342 0.84046 0.8561 0.87093 0.88366 0.89562 0.90656
##              PC19     PC20     PC21     PC22     PC23     PC24     PC25     PC26     PC27
## Standard deviation  0.8133 0.73918 0.72112 0.71053 0.64606 0.63389 0.5830 0.5442 0.50539
## Proportion of Variance 0.0105 0.00867 0.00825 0.00801 0.00663 0.00638 0.0054 0.0047 0.00405
## Cumulative Proportion 0.9171 0.92573 0.93399 0.94200 0.94863 0.95500 0.9604 0.9651 0.96916
```

```
##          PC28    PC29    PC30    PC31    PC32    PC33    PC34    PC35    PC36
## Standard deviation 0.48743 0.42674 0.42427 0.41483 0.38653 0.35092 0.32424 0.31646 0.2867
## Proportion of Variance 0.00377 0.00289 0.00286 0.00273 0.00237 0.00195 0.00167 0.00159 0.0013
## Cumulative Proportion 0.97293 0.97582 0.97867 0.98141 0.98378 0.98573 0.98740 0.98899 0.9903
##          PC37    PC38    PC39    PC40    PC41    PC42    PC43    PC44    PC45
## Standard deviation 0.28435 0.26060 0.25353 0.24800 0.21896 0.19537 0.18914 0.1767 0.16864
## Proportion of Variance 0.00128 0.00108 0.00102 0.00098 0.00076 0.00061 0.00057 0.0005 0.00045
## Cumulative Proportion 0.99158 0.99266 0.99368 0.99465 0.99541 0.99602 0.99659 0.9971 0.99753
##          PC46    PC47    PC48    PC49    PC50    PC51    PC52    PC53    PC54
## Standard deviation 0.1580 0.15287 0.1380 0.13101 0.10759 0.10374 0.09853 0.08760 0.08258
## Proportion of Variance 0.0004 0.00037 0.0003 0.00027 0.00018 0.00017 0.00015 0.00012 0.00011
## Cumulative Proportion 0.9979 0.99830 0.9986 0.99888 0.99906 0.99923 0.99939 0.99951 0.99962
##          PC55    PC56    PC57    PC58    PC59    PC60    PC61    PC62    PC63
## Standard deviation 0.08049 0.06927 0.05937 0.05673 0.05088 0.04001 0.02972 0.02789 0.01876
## Proportion of Variance 0.00010 0.00008 0.00006 0.00005 0.00004 0.00003 0.00001 0.00001 0.00001
## Cumulative Proportion 0.99972 0.99979 0.99985 0.99990 0.99994 0.99997 0.99998 0.99999 1.00000
```

```
# Gráfico de los primeros dos componentes
plot(pca$x[, 1:2])
```



5 Discusión

a

6 Conclusiones

a

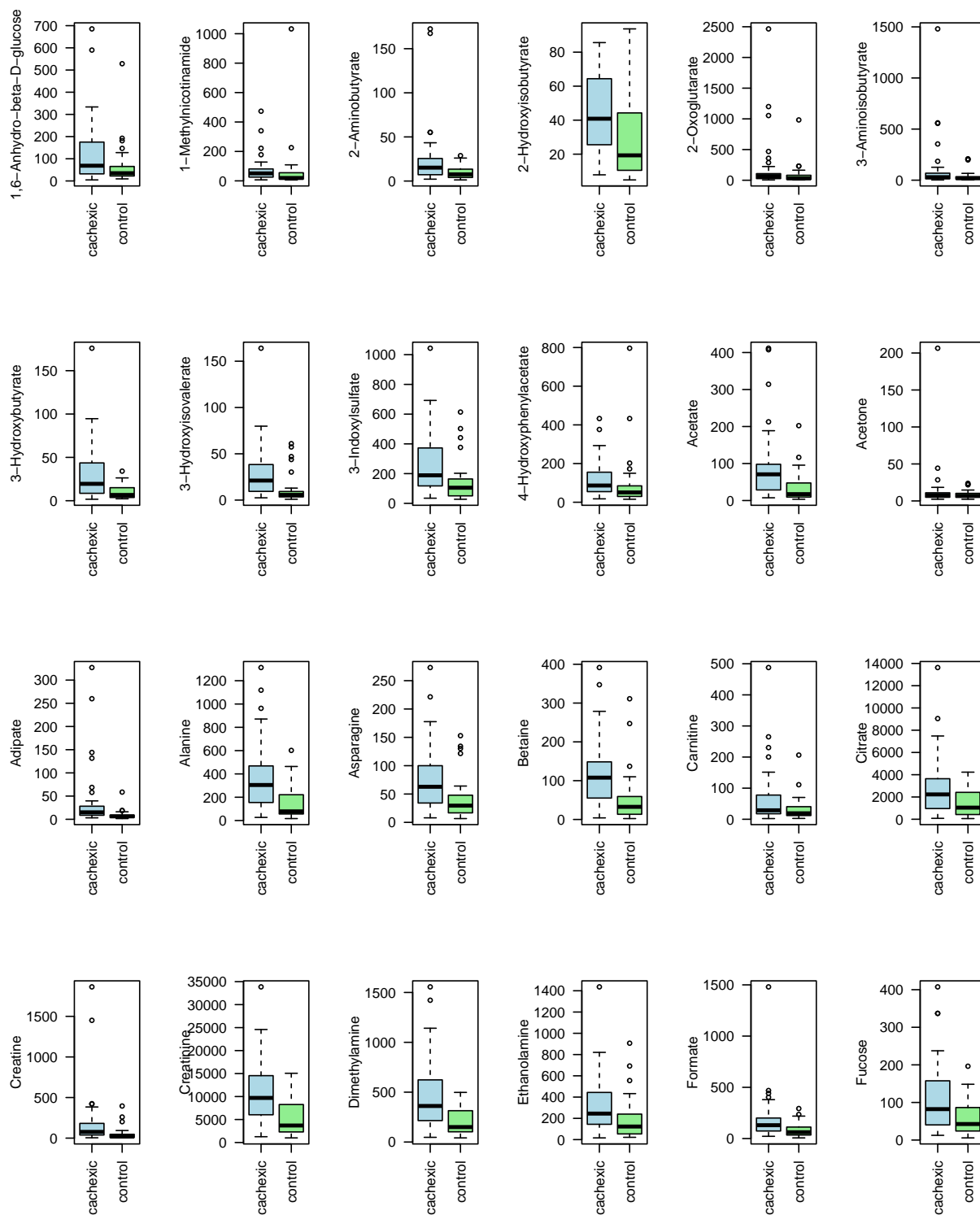
7 Referencias

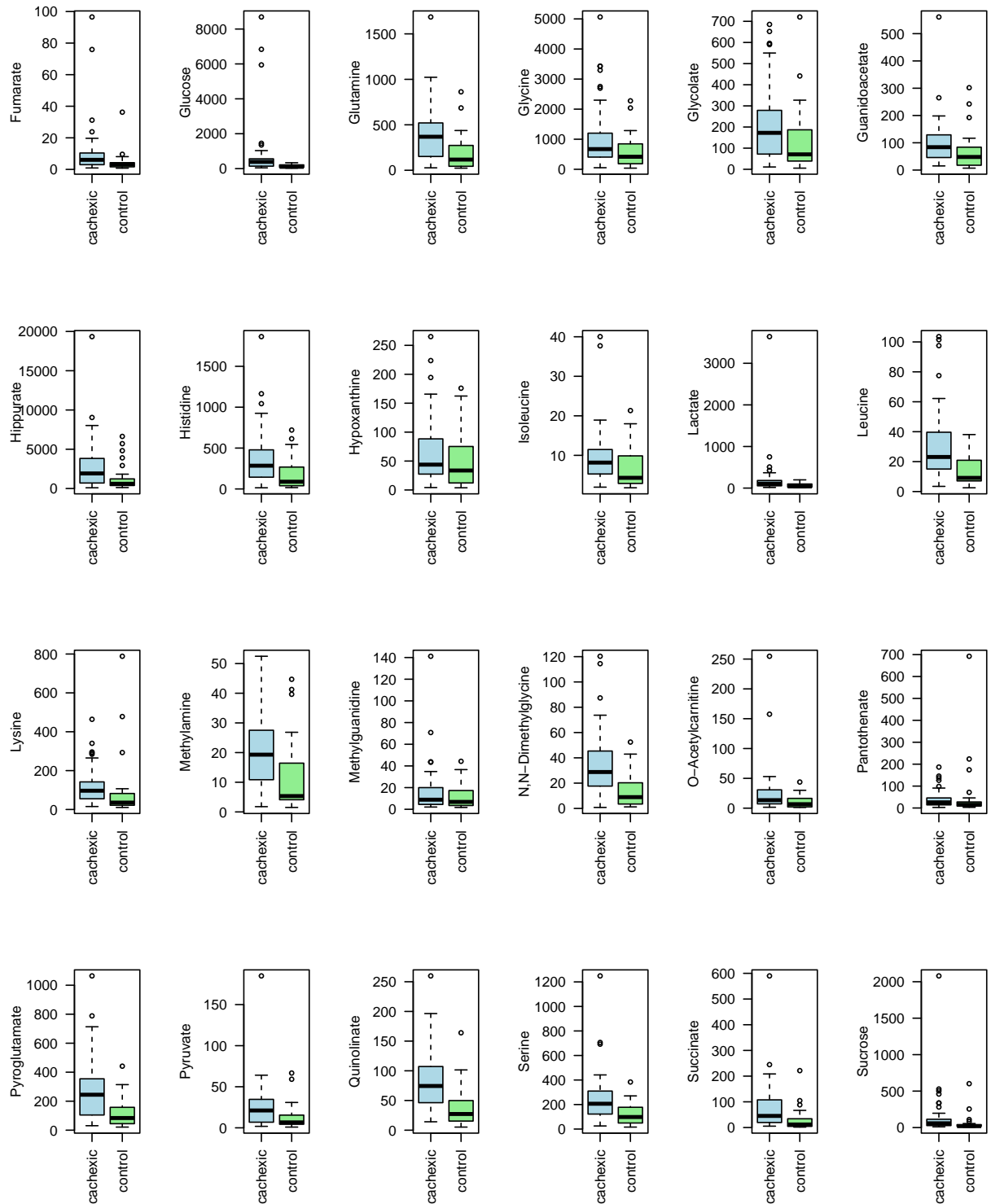
Morgan, Martin, Valerie Obenchain, Jim Hester, y Hervé Pagès. 2024. «SummarizedExperiment: SummarizedExperiment container». <https://doi.org/10.18129/B9.bioc.SummarizedExperiment>.

Nutrimetabolomics. 2025. «metaboData». <https://github.com/nutrimetabolomics/metaboData>.

8 Anexo

Anexo 1





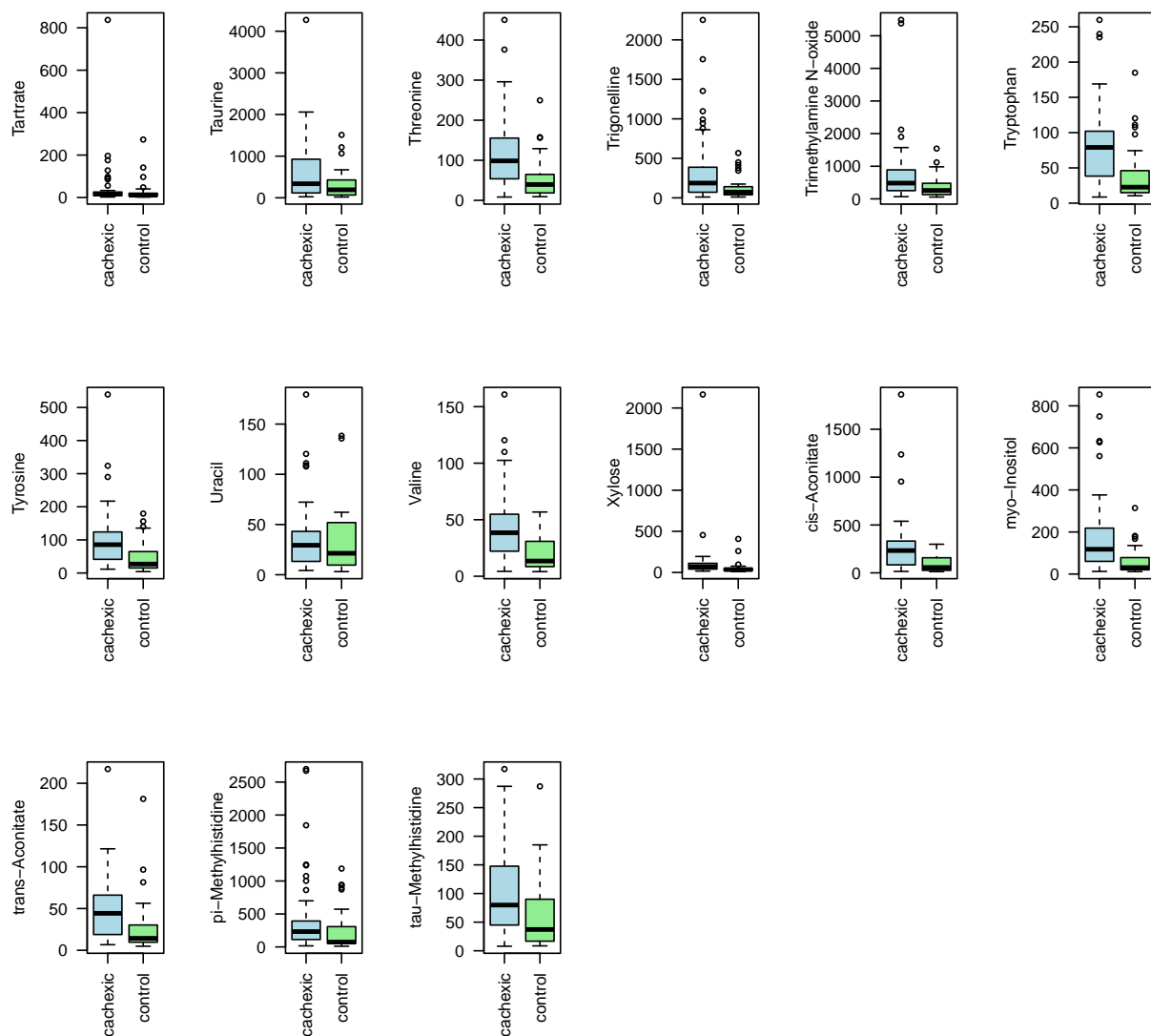
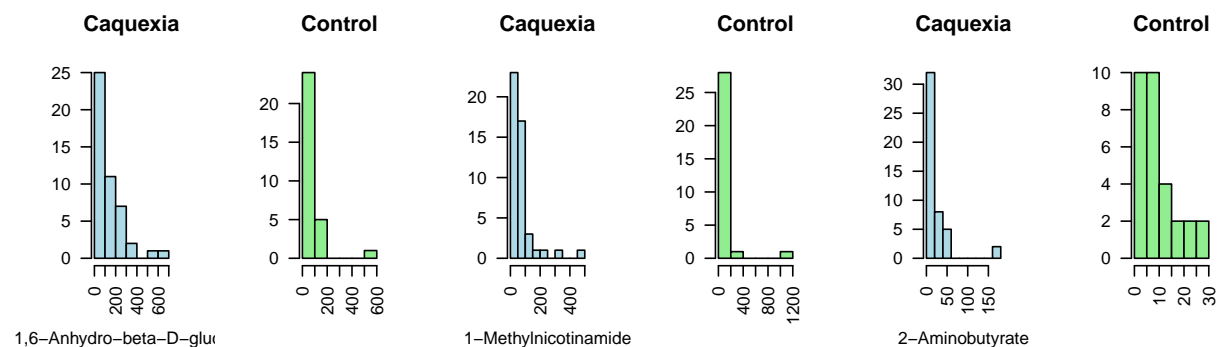
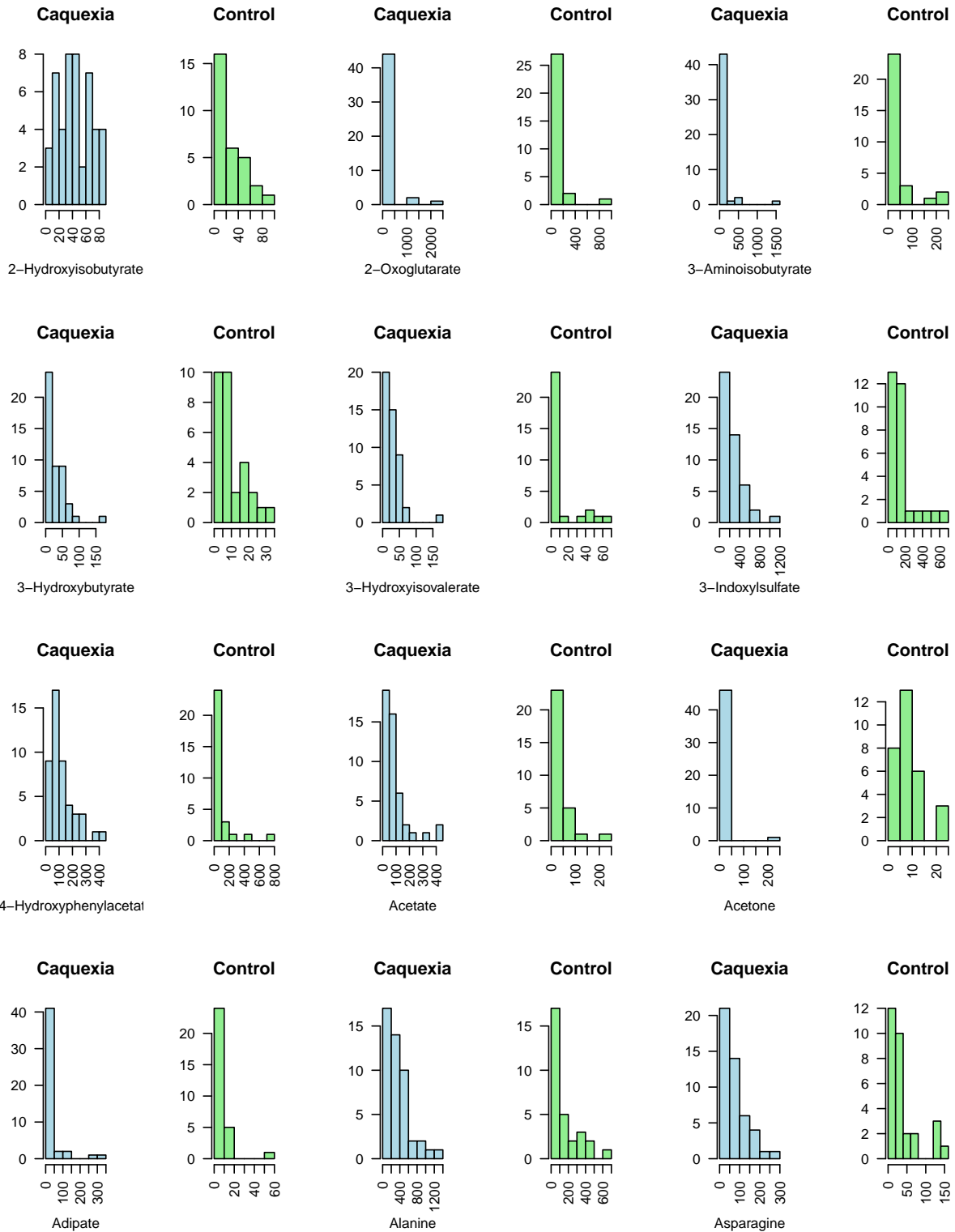
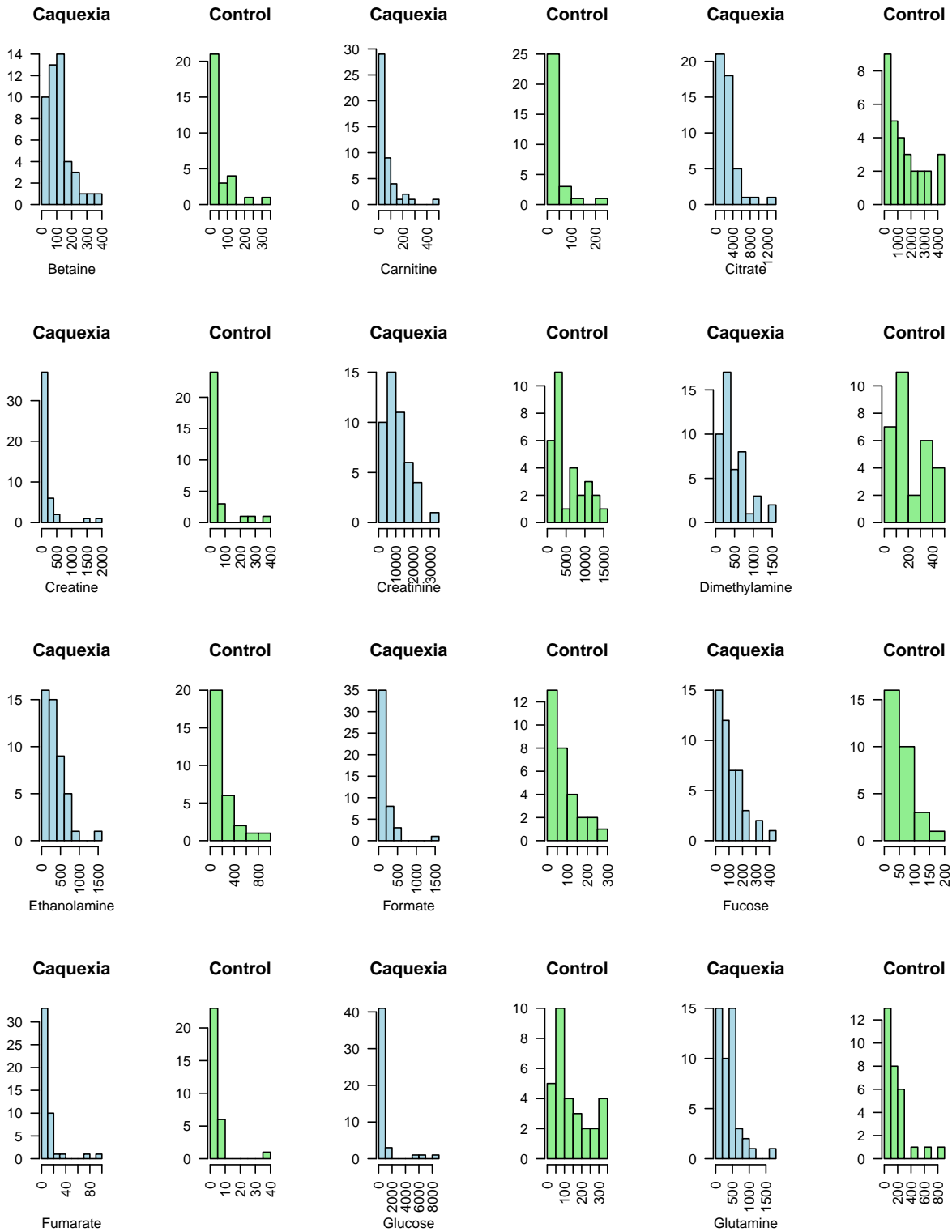


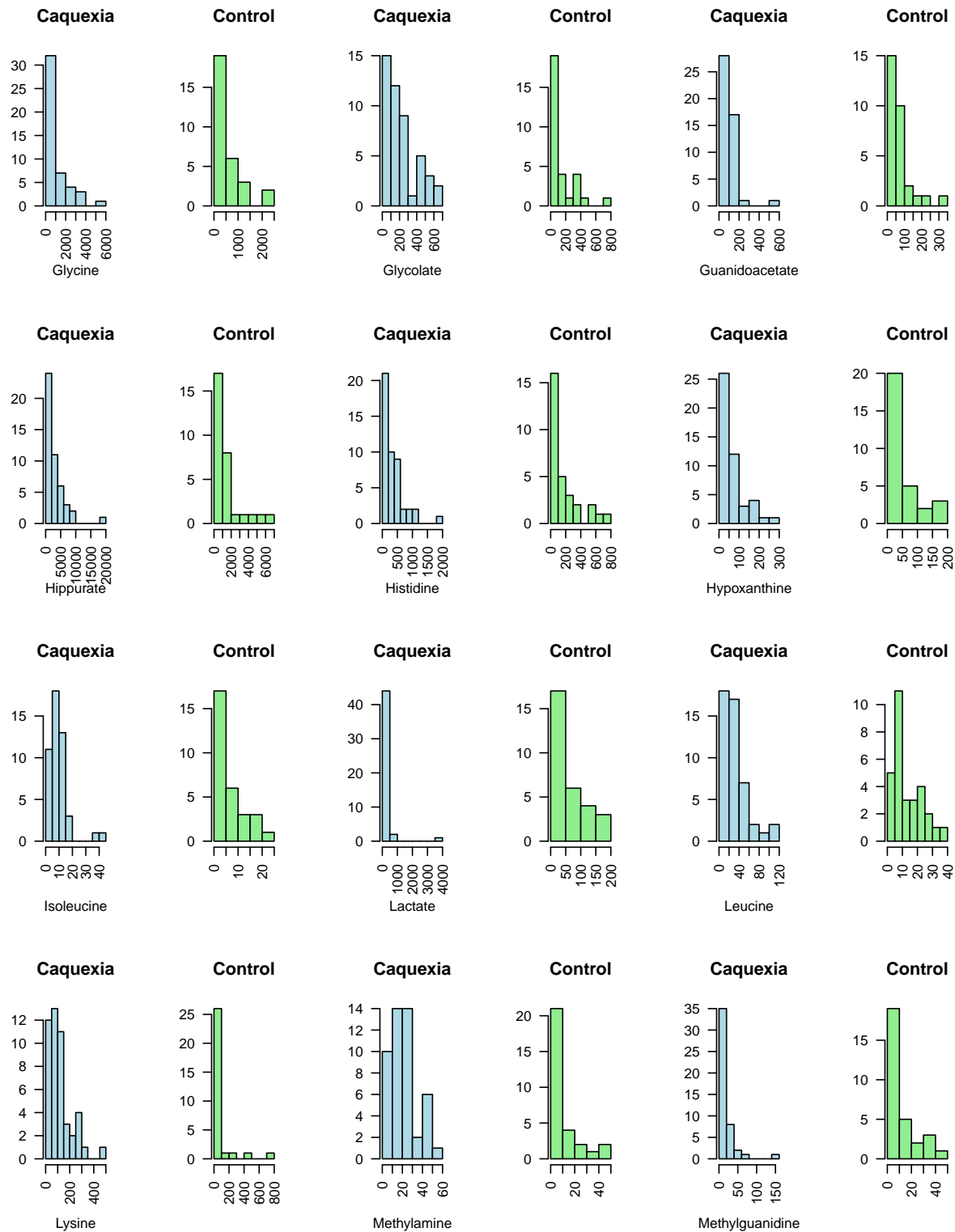
Figura Anexo 1: Boxplot de cada metabolito separado según condición `Muscle_loss`. En celeste el grupo que presenta caquexia y en verde claro el grupo control.

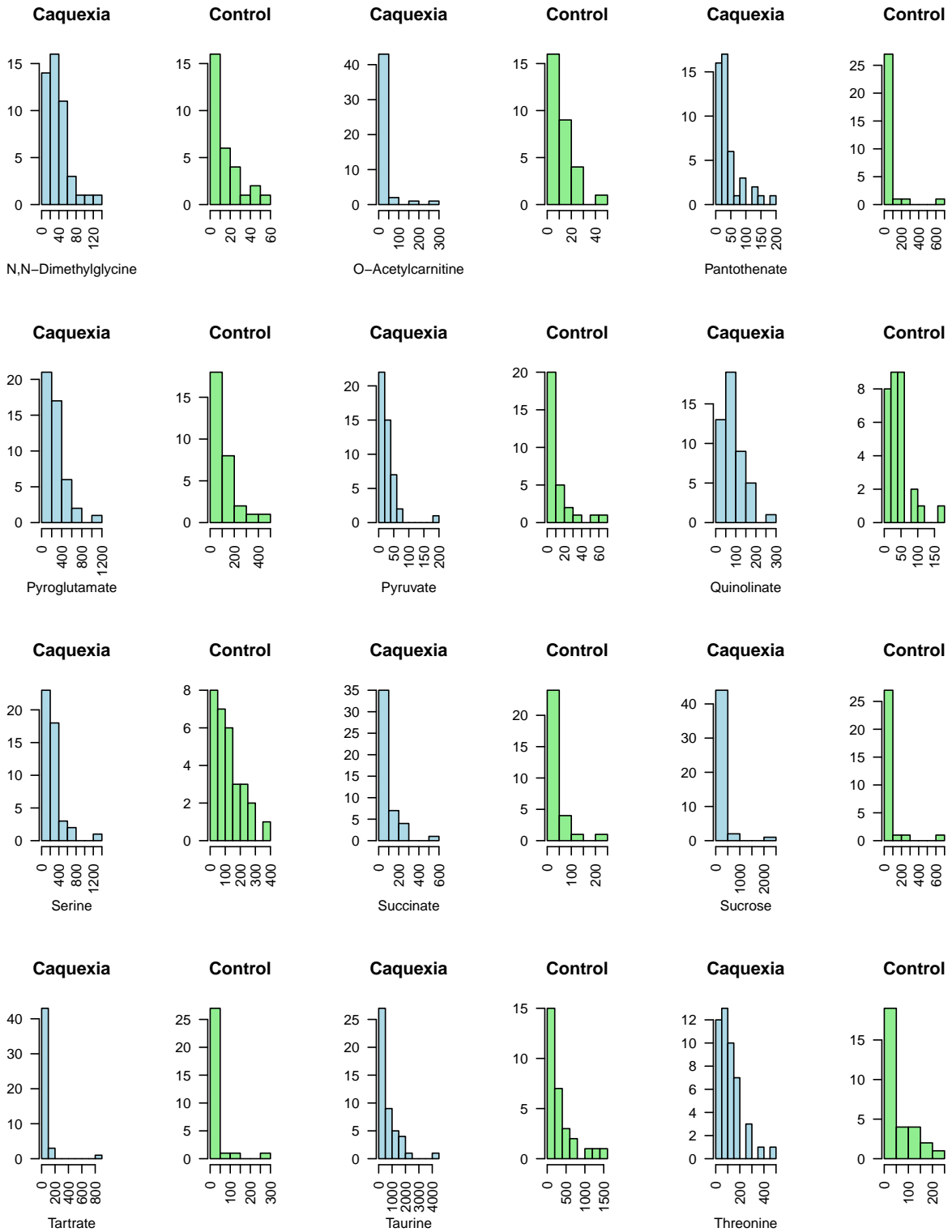
Anexo 2











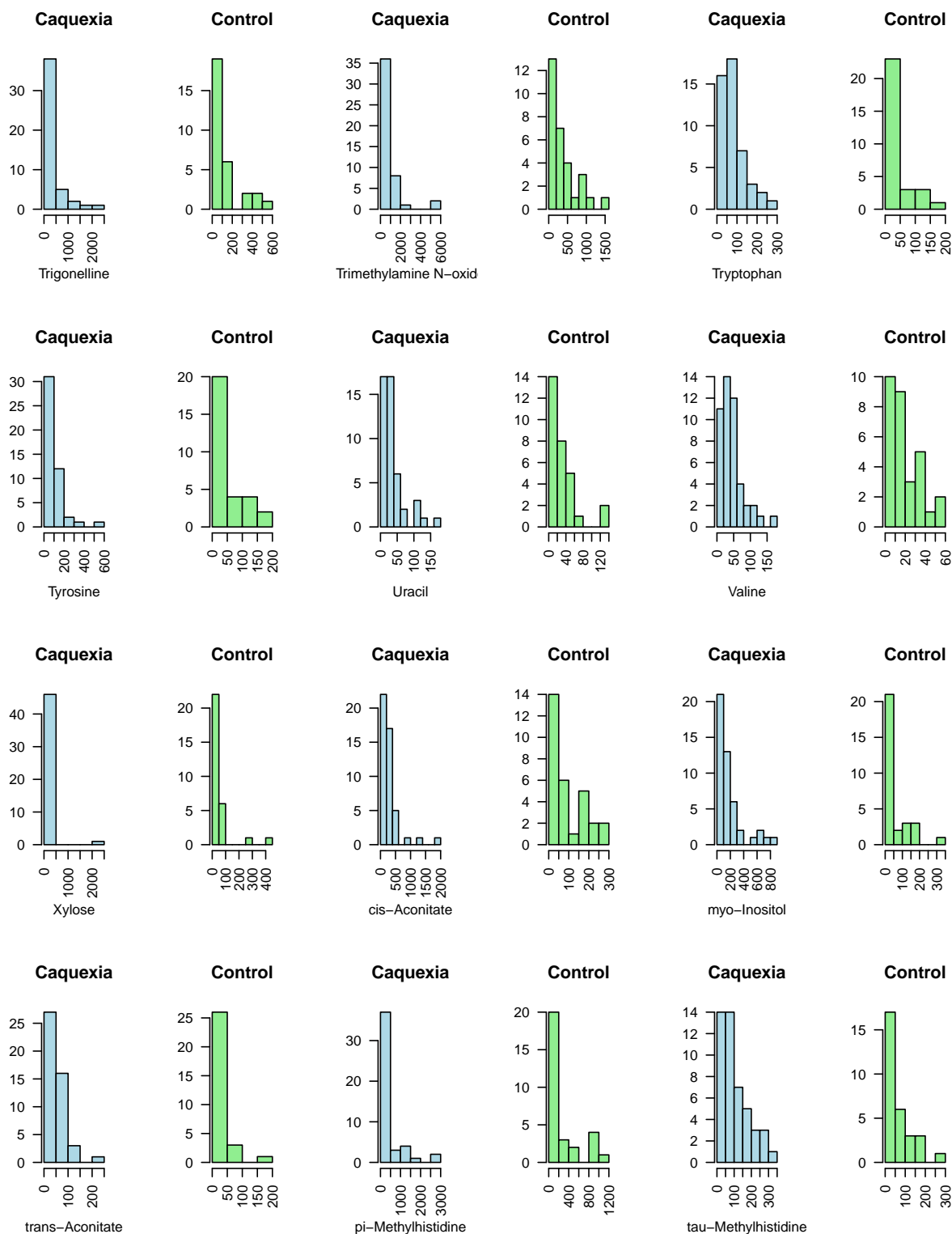


Figura Anexo 2: Distribución de cada metabolito separado según condición Muscle_loss. En celeste el grupo que presenta caquexia y en verde claro el grupo control.

Anexo 3

```

# Evaluar normalidad

metabolitos <- character()
v_caq_p <- character()
v_con_p <- character()
v_caq_w <- character()
v_con_w <- character()

for (i in 1:nrow(assay(se))) {
  metabolito <- rownames(assay(se))[i]

  caq <- as.numeric(assay(se)[i, colData(se)$Muscle.loss == "cachexic"])
  conl <- as.numeric(assay(se)[i, colData(se)$Muscle.loss == "control"])

  # Test normalidad
  test_caq <- shapiro.test(caq)
  test_con <- shapiro.test(control)

  # valor p < 0.05
  p_caq <- ifelse(test_caq$p.value > 0.05, "normal", "no normal")
  p_con <- ifelse(test_con$p.value > 0.05, "normal", "no normal")

  # valor W > 0.95
  w_caq <- ifelse(test_caq$statistic > 0.95, "Posible normal", "no normal")
  w_con <- ifelse(test_con$statistic > 0.95, "Posible normal", "no normal")

  metabolitos <- c(metabolitos, metabolito)

  v_caq_p <- c(v_caq_p, p_caq)
  v_con_p <- c(v_con_p, p_con)

  v_caq_w <- c(v_caq_w, w_caq)
  v_con_w <- c(v_con_w, w_con)
}

df_shapiro_p <- data.frame(
  Metabolito = metabolitos,
  Cachexia = v_caq_p,
  Control = v_con_p,
  stringsAsFactors = FALSE
)

df_shapiro_w <- data.frame(
  Metabolito = metabolitos,
  Cachexia = v_caq_w,
  Control = v_con_w,
  stringsAsFactors = FALSE
)

table(df_shapiro_p$Cachexia)

##
## no normal

```

```
##          63
```

```
table(df_shapiro_p$Control)
```

```
##  
## no normal  
##          63
```

```
table(df_shapiro_w$Cachexia)
```

```
##  
## no normal  
##          63
```

```
table(df_shapiro_w$Control)
```

```
##  
## no normal  
##          63
```