Gallardo_ADO_PEC1

Lorena Gallardo

1/5/2020

Contents

Enlace al GitHub:https://github.com/LorenaGaMo/Gallardo_ADO_PEC1.git

ABSTRACT

Este trabajo se basa en el artículo publicado por Hoopfer ED (n.d.) , donde se estudian los mecanismos celulares de la poda de axones en *Drosophila* (MB) durante la metamorfosis. Concretamente se utilizan 18 muestras del díptero divididas en tres gropos: en fase prepupa, recién pupado y a las 5 horas de haberlo hecho. Se compara los diferentes niveles de expresión de los genes después de normalizar y filtrar los datos.

OBJETIVOS

El objetivo del estudio es estudiar las diferencias de expresión génica de la poda de axones dependiendo del estado de desarrollo en que se encuentra el díptero.

El objetivo de este trabajo es poner en práctica los conocimientos adquiridos para realizar un análisis de datos de microarray. Se trata de un estudio de un factor con tres niveles diferentes (prepupa, pupa, pupa de 5 horas) y el objetivo es seleccionar los genes que se estén expresando de manera diferencial según la edad del individuo.

MATERIAL Y MÉTODOS

Software y naturaleza de los datos

El software utilizado para realizar este estudio es el siguiente: R[1] (versión 3.6.3) con el interfaz RStudio[2] y los paquetes de Bioconductor Affy[3] (versión 3.10). El paquete de R ('limma()') para el ajuste del modelo lineal y la selección de genes. Para el * Gene Enrichment Analysis* se ha utilizado la aplicación g:Profiler,[4] debido a que con el paquete *ReactomePA* de R he tenido diversos problemas.

Este trabajo se basa en el artículo de Hoopfer ED (n.d.) y los datos públicos extraidos de de la base de datos Gene Expression Omnibus con su correspondiente códiga GSE10012.[5]

Preparación de los datos

Antes de empezar con el analisi propiamente se debe preparar el entorno de trabajo con un sistema de archivos.el hecho de tener la información bien estructurada facilita el proceso del analisis. Es recomendable tener instaladas las librerias necesarias ya que el proceso puede ser largo.

```
> setwd(".")
> dir.create("data")
> dir.create("results")
> dir.create("figuras")
```

Lectura datos crudos

Se debe preparar un fixero *targets.csv* de manera manual con la información obtenida en la base de datos (Table 1).

[1] https://cran.r-project.org/index.html
[2] https://www.rstudio.com/
[3] https://www.bioconductor.org/
[4] https://biit.cs.ut.ee/gprofiler/gost
[5] https://www.ncbi.nlm.nih.gov/geo/geo2r/?acc=GSE10012

```
> targets <- read.csv2("./data/targets.csv", header = TRUE, sep = ",")
> knitr::kable(
+   targets, booktabs = TRUE,
+   caption = 'Contenido del fichero *targets.csv* ')
```

Table 1: Contenido del fichero *targets.csv*

ï..FileName	Age	Grupo	ShortName
GSM252982	Prepupa	Pre	Pre82
GSM252983	Prepupa	Pre	Pre83
GSM252984	Prepupa	Pre	Pre84
GSM252985	Prepupa	Pre	Pre85
GSM252986	Prepupa	Pre	Pre86
GSM252987	Prepupa	Pre	Pre87
GSM252988	Prepupa	Pre	Pre88
GSM252989	Pupa 0h	P0	P0_89
GSM252990	Pupa 0h	P0	P0_90
GSM252991	Pupa 0h	P0	P0_91
GSM252992	Pupa 0h	P0	P0_92
GSM252993	Pupa 0h	P0	P0_93
GSM252994	Pupa 0h	P0	P0_94
GSM252995	Pupa 0h	P0	P0_95
GSM252996	Pupa 0h	P5	P5_96
GSM252997	Pupa 0h	P5	P5_97
GSM252998	Pupa 0h	P5	P5_98

Para leer los ficheros *.CEL* propiamente se crea un nuevo data frame *rawData*, donde previamente de debe crar un nuevo objeto *my.targets* associado a los archivos *.CEL*.

Control de calidad y exploración de los datos

Se realiza un control de calidad de los datos con el paquete 'arrayQualityMetrics()' el cual realiza diferentes test. El resumen del análisis de calidad lo encontramos dentro de la carpeta "resunts" en el fichero "index.htlm"(Figura 1). Las columnas 1, 2, y 3 nos indican los diferentes criterios de detección de *outlier* respectivamente: mediante distancias entre arrays, mediante boxplot y mediante MA plots. Podemos observar que en las columnas 1 y 3 tenemos señaladas diversas muestras con una "X", esto nos indica que los posibles problemas a la ahora de normalizar los datos podrían derivarse de estas muestras ya que nos indica que esas muestras se pueden considerar que tienen *outliers* en su distibución, en la categoria de la casilla donde estan señaladas.

```
> arrayQualityMetrics(rawData, outdir = file.path("./results","rawData_quality"), force = T)
```

```
> knitr::include_graphics("figuras/ResumQM.png")
```

El control de calidad de los datos crudos también se puede realizar con un análisis gráfico. Para este trabajo se han realizado un boxplot (Fig.2), un cluster (Fig.3) y un análisis de componentes principales (Fig.4).En el caso del análisis de las componentes principales se ha utilizado la función definida por Gonzalo Sanz and Sánchez-Pla (2019) y especificada en el apendice adjunto.

En el boxplot podemos observar que la intensidad de las muestras es algo irregular tanto en el tamaño de las cajas como en su media. Tendremos que ver si al normalizar los datos se mejora esta variabilidad.

	array	sampleNames	*1	*2	*3	Age	Grupo	ShortName
☐	1	Pre82				Prepupa	Pre	Pre82
☐	2	Pre83			x	Prepupa	Pre	Pre83
☐	3	Pre84				Prepupa	Pre	Pre84
☐	4	Pre85			x	Prepupa	Pre	Pre85
☐	5	Pre86				Prepupa	Pre	Pre86
☐	6	Pre87	x		x	Prepupa	Pre	Pre87
☐	7	Pre88				Prepupa	Pre	Pre88
☐	8	P0_89	x		x	Pupa 0h	P0	P0_89
☐	9	P0_90			x	Pupa 0h	P0	P0_90
☐	10	P0_91			x	Pupa 0h	P0	P0_91
☐	11	P0_92				Pupa 0h	P0	P0_92
☐	12	P0_93			x	Pupa 0h	P0	P0_93
☐	13	P0_94			x	Pupa 0h	P0	P0_94
☐	14	P0_95			x	Pupa 0h	P0	P0_95
☐	15	P5_96			x	Pupa 0h	P5	P5_96
☐	16	P5_97			x	Pupa 0h	P5	P5_97
☐	17	P5_98			x	Pupa 0h	P5	P5_98

Figure 1: Fig.1 Tabla resumen de la calidad de los datos del fichero index.htlm

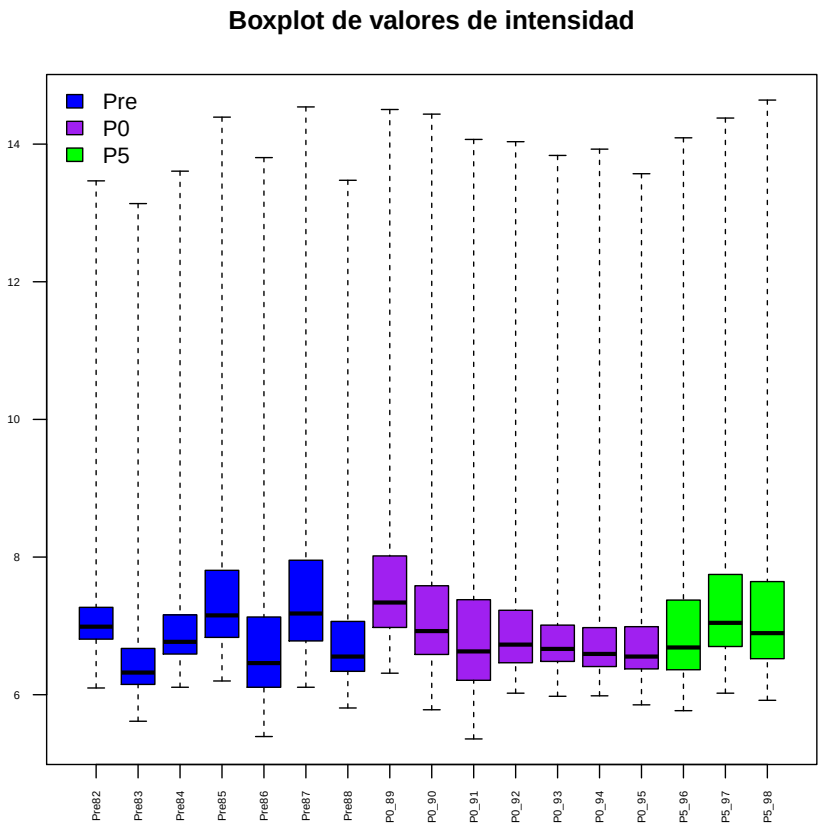


Figure 2: Boxplot de los datos crudos

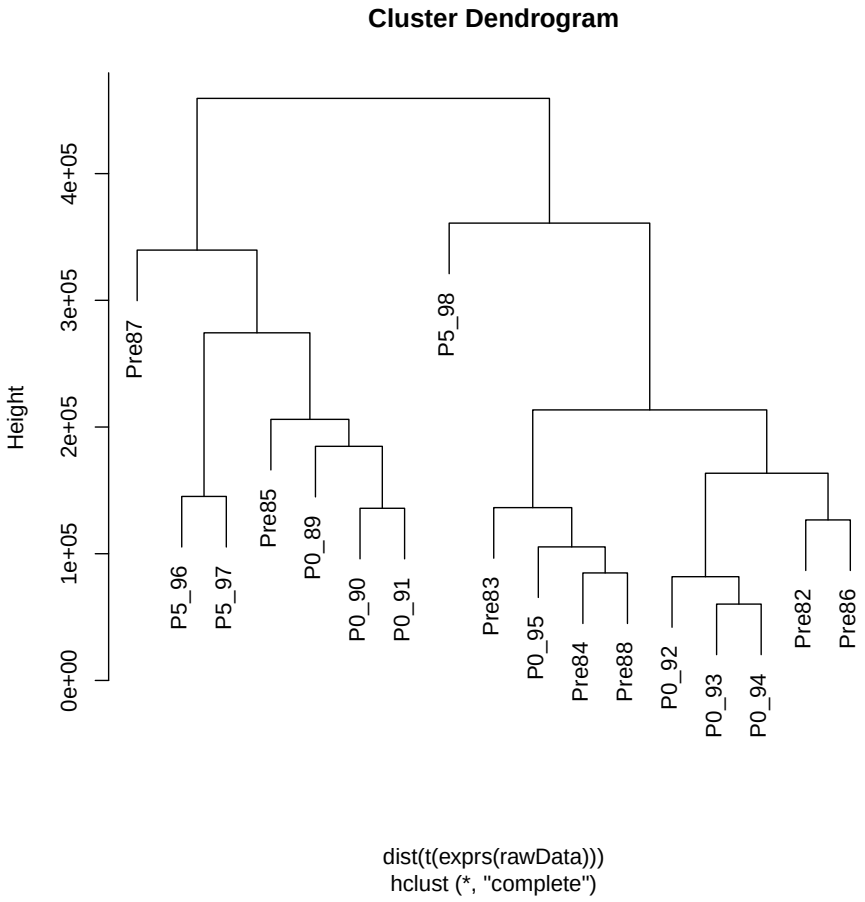


Figure 3: Cluster de los datos crudos

Con el cluster de los datos crudos podemos observar que no existe una jerarquía entre los diferentes grupos, todas las muestras están mezcladas. Como pasa con el boxplot miraremos de mejóralo con la normalización de los datos.

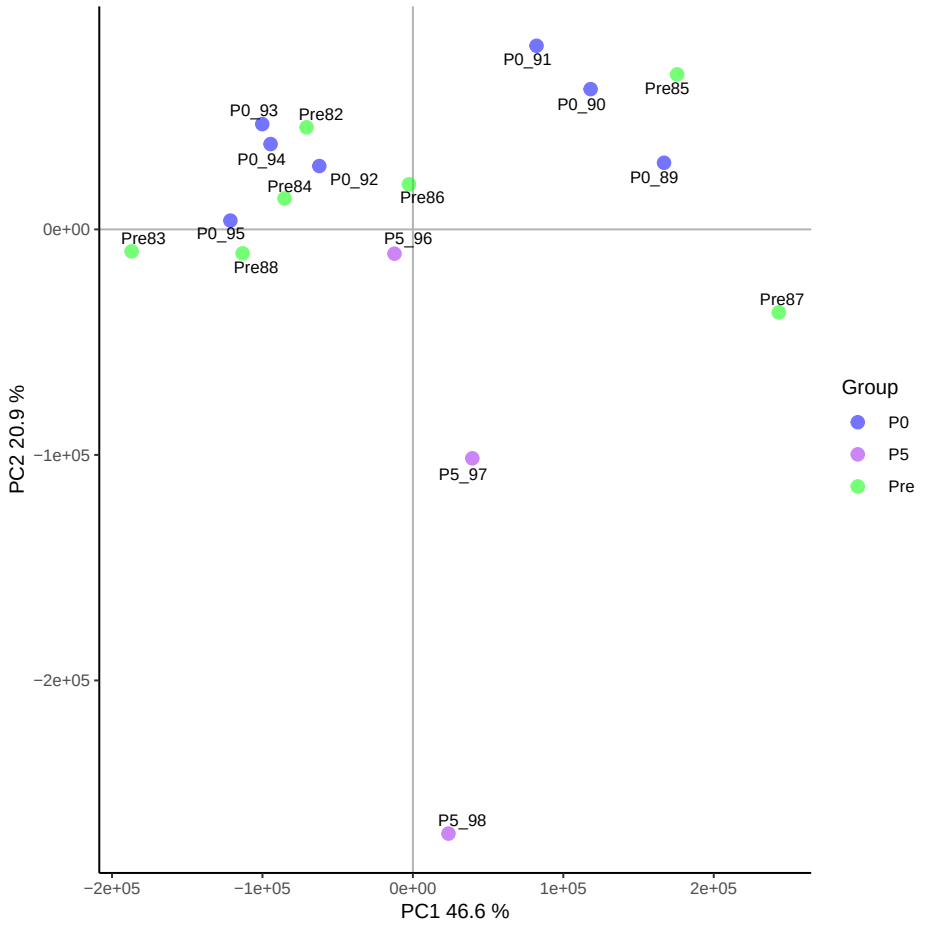


Figure 4: Anàlisis PCA de los datos crudos

En el gráfico de las componentes principales vemos que la primera componente se explica el 46.6% de la variabilidad y el 20.9% con la segunda componente. Pero no se observan grupos definidos i diferencias entre las muestras.

Normalización

Con el paso de normalización de los datos pretendemos mejorar los análisis anteriores con el objetivo de poder comparar las diferentes muestras entre sí y eliminar la variabilidad, debida a los efectos de la técnica o el análisis y no a la producida por las diferencias biológicas, con la transformación de los datos. Se ha utilizado un método de normalización RMA, uno de los métodos más utilizados en la normalización de datos procedentes de array de Affymetrix, descrito por Irizarry (2003). Para la realización de esta se ha creado un nuevo objeto con la función 'rma()'.

```
Background correcting
Normalizing
Calculating Expression
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 14010 features, 17 samples
  element names: exprs
protocolData
  rowNames: Pre82 Pre83 ...  P5_98 (17 total)
  varLabels: exprs dates
  varMetadata: labelDescription channel
phenoData
  rowNames: Pre82 Pre83 ...  P5_98 (17 total)
  varLabels: Age Grupo ShortName
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation: pd.drosgenome1
```

Control de calidad y exploración datos normalizados

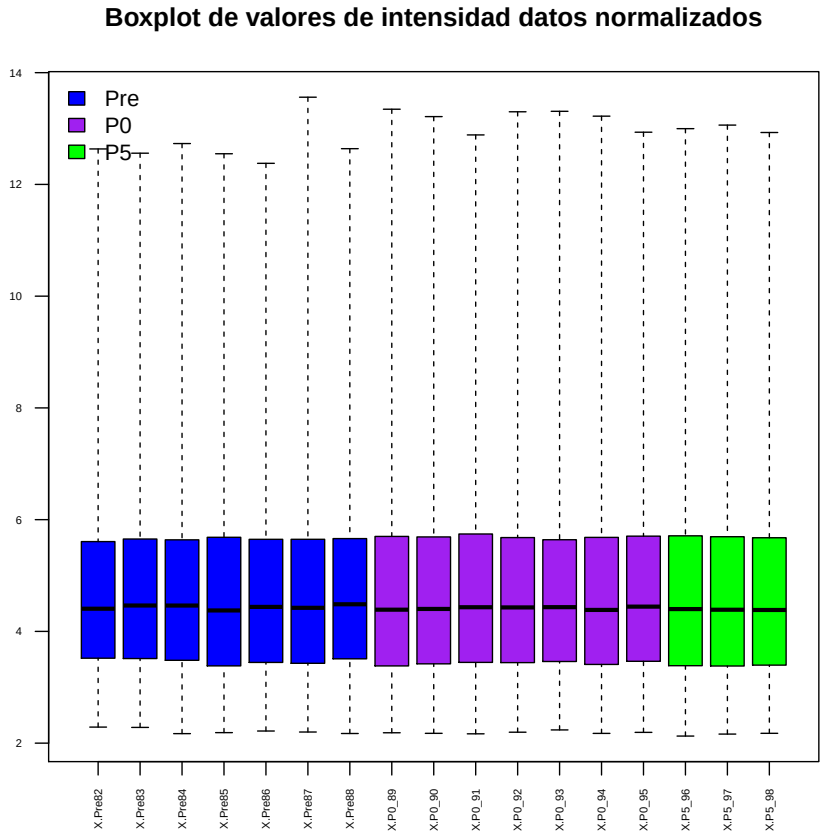
Se vuelve a realizar un control de calidad de los datos pero esta vez ya transformados. En los nevos resultados se puede observar una mejora considerable de los datos, la unica muestra que presenta problemas es la Pre82 cn dos *outliers*.

```
> arrayQualityMetrics(eset_rma, outdir = file.path("./results", "rawDataNorm_quality"), force = T)
```

```
> knitr::include_graphics("figuras/ResumQM_norm.png")
```

	array	sampleNames	*1	*2	*3	Age	Grupo	ShortName
☐	1	Pre82	x	x		Prepupa	Pre	Pre82
☐	2	Pre83				Prepupa	Pre	Pre83
☐	3	Pre84				Prepupa	Pre	Pre84
☐	4	Pre85				Prepupa	Pre	Pre85
☐	5	Pre86				Prepupa	Pre	Pre86
☐	6	Pre87				Prepupa	Pre	Pre87
☐	7	Pre88				Prepupa	Pre	Pre88
☐	8	P0_89				Pupa 0h	P0	P0_89
☐	9	P0_90				Pupa 0h	P0	P0_90
☐	10	P0_91				Pupa 0h	P0	P0_91
☐	11	P0_92				Pupa 0h	P0	P0_92
☐	12	P0_93				Pupa 0h	P0	P0_93
☐	13	P0_94				Pupa 0h	P0	P0_94
☐	14	P0_95				Pupa 0h	P0	P0_95
☐	15	P5_96				Pupa 0h	P5	P5_96
☐	16	P5_97				Pupa 0h	P5	P5_97
☐	17	P5_98				Pupa 0h	P5	P5_98

Figure 5: Tabla resumen análisa datos normalizazados



El boxplot corrobora lo que hemos visto con el análisis anterior. La intensidad de las muestras es mucho más regular tanto en el tamaño de las cajas como en su media, aunque se pueden apreciar pequeñas diferencias.

Esta vez sí que podemos observar una jerarquía definida en el clúster de los datos normalizados. Se diferencian claramente las prepupas de las pupas y dentro del estas últimas se observa que las pupas de cero horas también de diferencian de las de 5 h.

En análisis de componentes principales de los datos normalizados volvemos a ver los tres grupos definidos y diferenciados como lo podíamos ver en el clúster. Esta vez la variabilidad entre ellos esta explicada en un 50.2% por la primera componente y en un 10.1% por la segunda.

Selección de genes

Para ver qué porcentaje de genes presentan una variabilidad superior a la variabilidad intrínseca entre muestras, ya que estos serían los genes que están expresados diferencialmente. Esto se puede observar gráficamente con la representación de la desviación estándar de todos los genes. Se puede apreciar que hay un gran número de genes que se están expresando diferencialmente, aunque no llega a ser el 5% de los genes. Los genes que se consideran con más variabilidad son aquellos con una desviación estándar por encima de los percentiles 90% y 95% , representados en el grafico por las líneas verticales.

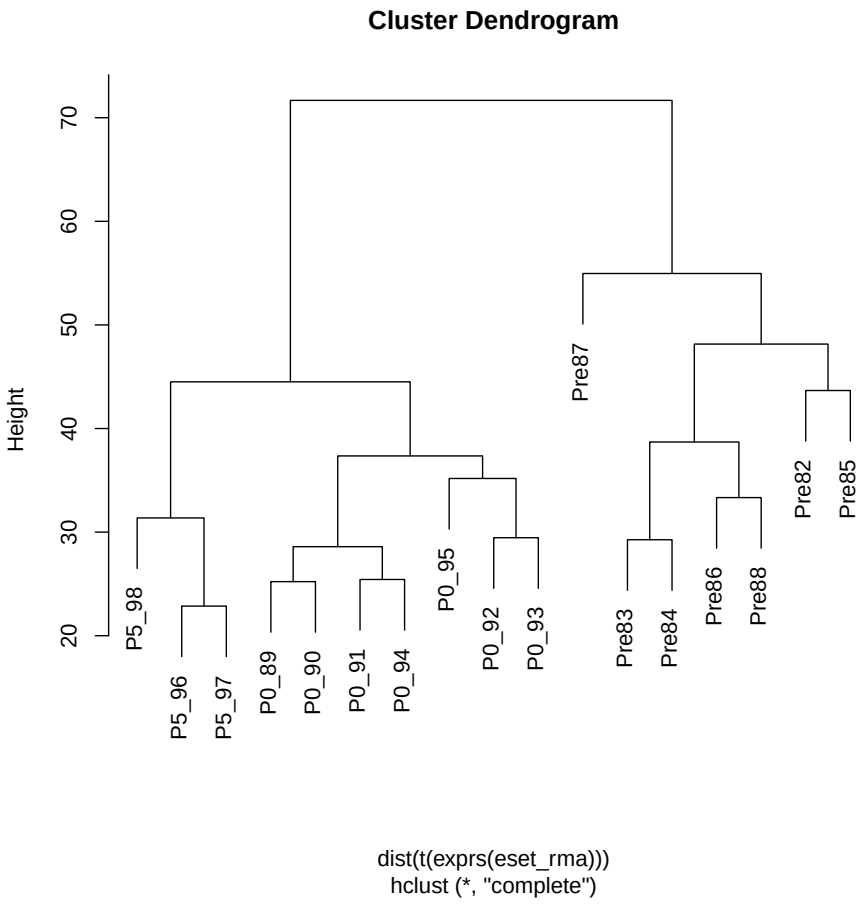


Figure 6: Cluster datos normalizados

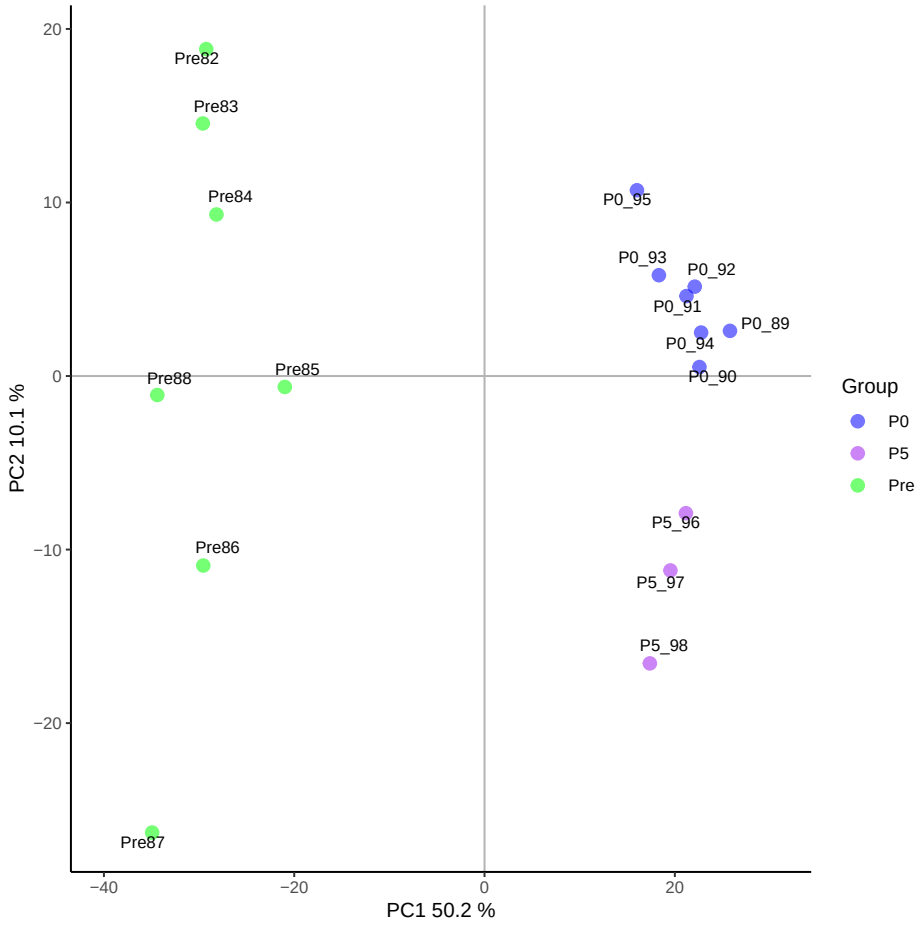


Figure 7: Anàlisis PCA de los datos normalizadoss

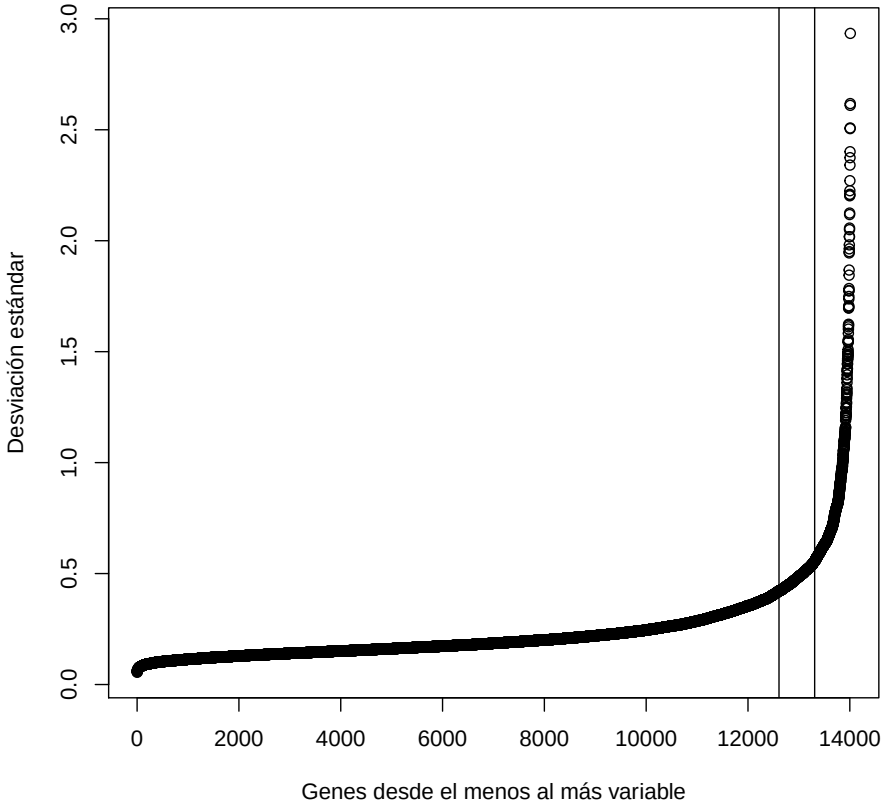


Figure 8: Distrubución de la variabilidad genica

Anotaciones

Las anotaciones son el paso previo al filtraje de los datos normalizados para poder eliminar los genes sin identificador asociado. Para esto se debe cargar la librería *drosgenome1.db* para este estudio (cada estudio tiene su paquete de anotaciones asociado) y así anidar el identificador del gen con la sonda.

```
> library(drosgenome1.db)
> annotation(eset_rma)<-"drosgenome1.db"
```

Filtraje de genes

Una vez tenemos las anotaciones correspondientes realizadas se procede al filtraje de los genes con poca variabilidad o variabilidad aleatoria. Este filtraje se puede realizar con el paquete *genefiltre* con la función *nsfilter()*. Con este filtraje hemos conseguido eliminar 1023 genes.

```
$numDupsRemoved
[1] 1483

$numLowVar
[1] 8628

$numRemoved.ENTREZID
[1] 1023
```

Control de calidad datos filtrados

Una vez filtrados los datos normalizados se vuelve a realizar un control de calidad de los datos como en las veces anteriores. Esta vez podemos comprobar que se ha vuelto a mejorar el resultado y la muestra que presentaba problemas de *outliers* ya no los presenta.

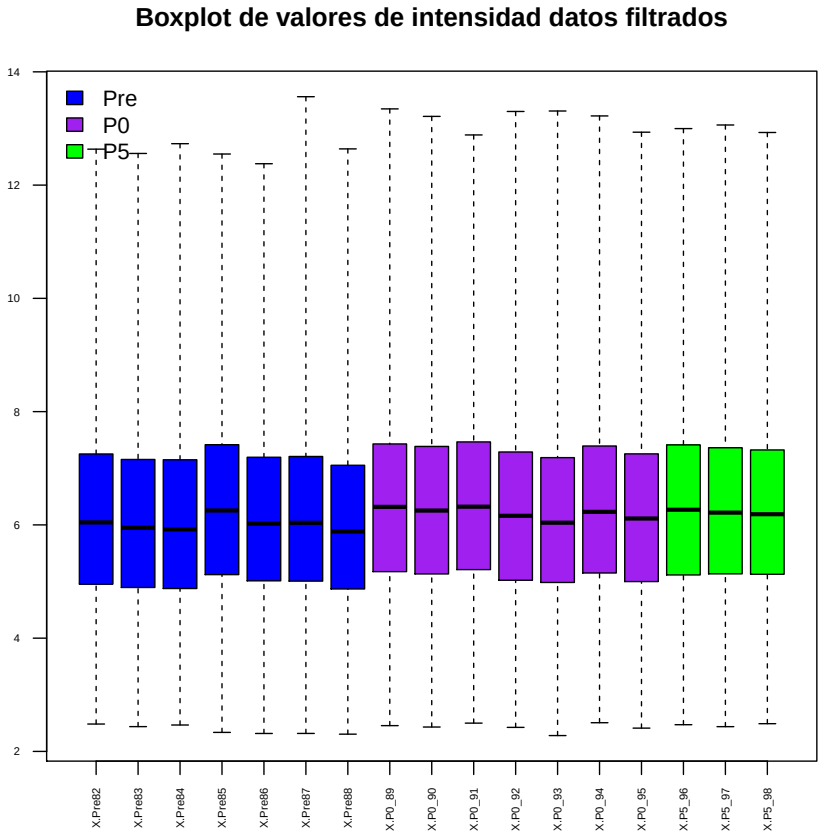
Gráficamente el boxplot no presenta ninguna variación apreciable al anterior boxplot de datos normalizados. En el clúster podemos ver una reorganización interna de los grupos pero estos se mantienes igual de diferenciados entre sí. En el PCA los tres grupos de diferencian aún más y se crea un subgrupo en las muestras de prepupa. El porcentaje de la variabilidad explicada por la primera componente vuelve a augmentar (66%) y la de la segunda a disminuir(8.3%).

```
> arrayQualityMetrics(eset_filt, outdir = file.path("./results", "Norm_data_filt"), force= T)
```

```
> knitr::include_graphics("figuras/ResumQM_filt.png")
```

	array	sampleNames	*1	*2	*3	Age	Grupo	ShortName
☐	1	Pre82				Prepupa	Pre	Pre82
☐	2	Pre83				Prepupa	Pre	Pre83
☐	3	Pre84				Prepupa	Pre	Pre84
☐	4	Pre85				Prepupa	Pre	Pre85
☐	5	Pre86				Prepupa	Pre	Pre86
☐	6	Pre87				Prepupa	Pre	Pre87
☐	7	Pre88				Prepupa	Pre	Pre88
☐	8	P0_89				Pupa 0h	P0	P0_89
☐	9	P0_90				Pupa 0h	P0	P0_90
☐	10	P0_91				Pupa 0h	P0	P0_91
☐	11	P0_92				Pupa 0h	P0	P0_92
☐	12	P0_93				Pupa 0h	P0	P0_93
☐	13	P0_94				Pupa 0h	P0	P0_94
☐	14	P0_95				Pupa 0h	P0	P0_95
☐	15	P5_96				Pupa 0h	P5	P5_96
☐	16	P5_97				Pupa 0h	P5	P5_97
☐	17	P5_98				Pupa 0h	P5	P5_98

Figure 9: Resumen de la calidad de los datos filtrados



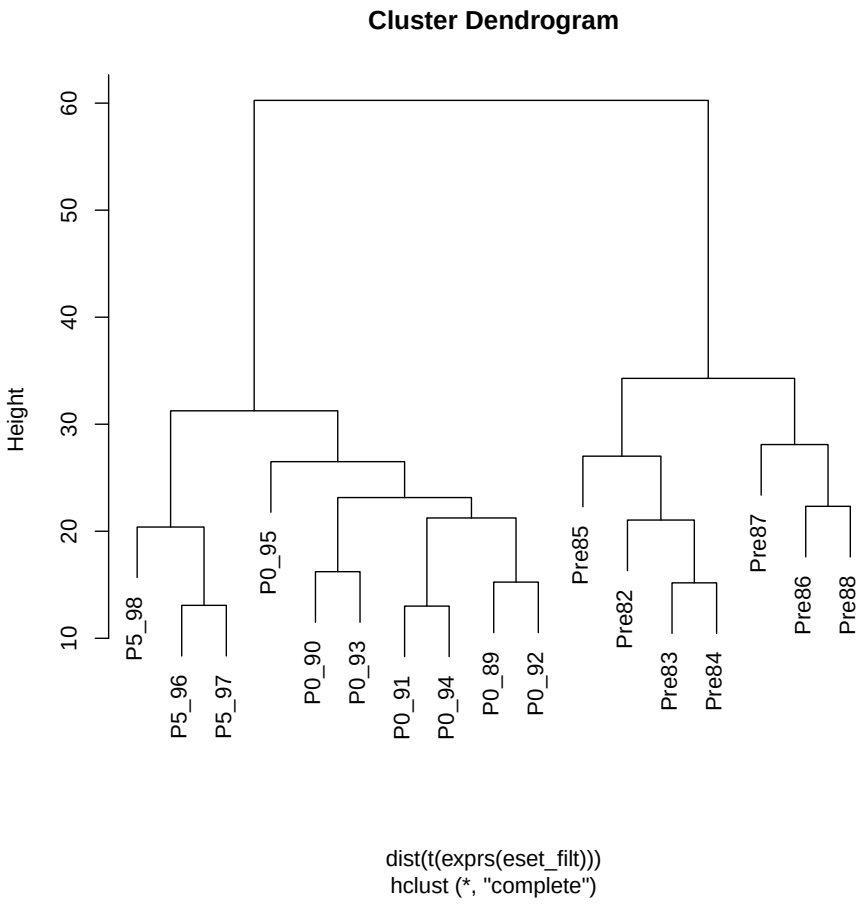


Figure 10: Cluster datos filtrados

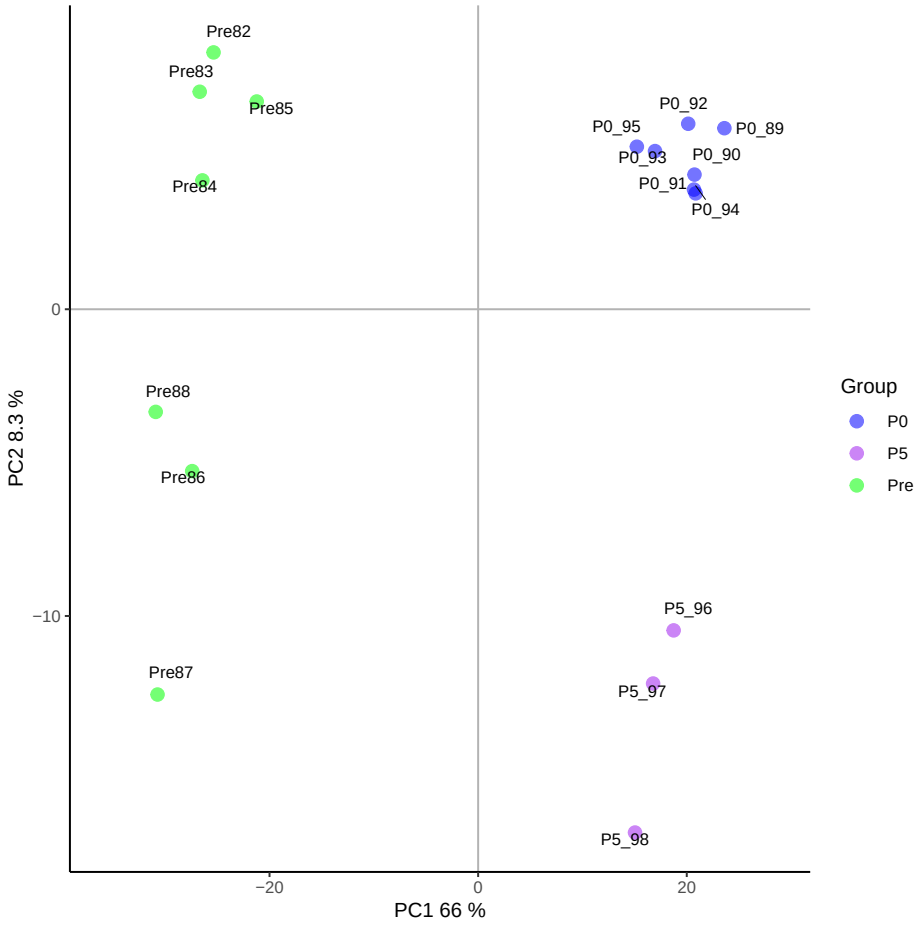


Figure 11: Anàlisis PCA de los datos filtrados

Diseño matriz experimental

El siguiente paso en el estudio es la creación o diseño de la matriz experimental que se realiza con la función *designMat()* . La matriz resultante es una matriz de 1 y 0 donde el valor 1 quiere decir que esa muestra (que las encontramos dispuestas en filas) corresponde a un determinado grupo (dispuestos en las columnas).

```
       P0 P5 Pre
 Pre82  0  0   1
 Pre83  0  0   1
 Pre84  0  0   1
 Pre85  0  0   1
 Pre86  0  0   1
 Pre87  0  0   1
 Pre88  0  0   1
 P0_89  1  0   0
 P0_90  1  0   0
 P0_91  1  0   0
 P0_92  1  0   0
 P0_93  1  0   0
 P0_94  1  0   0
 P0_95  1  0   0
 P5_96  0  1   0
 P5_97  0  1   0
 P5_98  0  1   0
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$Grupo
[1] "contr.treatment"
```

A continuación seguimos con la matriz de contrstes para la comparación 2 a 2 de los grupos creados

```
      Contrasts
Levels PP0 PP5 PP05
   P0   -1   0    1
   P5    0  -1   -1
   Pre   1   1    0
```

Selección de genes diferencialmente expresados y anotación

Una vez ya tenemos definidas la matriz de diseño y la de contraste con el paquete *limma*podemos proceder a la selección de genes expresados diferencialmente y a su anotación.

```
> fit<-lmFit(eset_filt, designMat)
> fit.main<-contrasts.fit(fit, cont.matrix)
> fit.main<-eBayes(fit.main)
> head(fit.main)
```

```
An object of class "MArrayLM"
$coefficients
           Contrasts
                   PP0        PP5        PP05
```

```
  151494_at -0.2070358 -0.4289567 -0.22192096
  153135_at -0.7667038 -0.8267176 -0.06001381
  154994_at -1.1560770 -1.3125978 -0.15652079
  152298_at -0.4667538 -0.6124326 -0.14567880
  143798_at  2.5541842  4.2764838  1.72229962
  152911_at  0.6639670  1.2701890  0.60622199

$rank
[1] 3

$assign
[1] 1 1 1

$qr
$qr
              P0        P5        Pre
 Pre82 -2.645751  0.000000  0.0000000
 Pre83  0.000000 -1.732051  0.0000000
 Pre84  0.000000  0.000000 -2.6457513
 Pre85  0.000000  0.000000  0.3779645
 Pre86  0.000000  0.000000  0.3779645
12 more rows ...

$qraux
[1] 1.000000 1.000000 1.377964

$pivot
[1] 1 2 3

$tol
[1] 1e-07

$rank
[1] 3


$df.residual
[1] 14 14 14 14 14 14

$sigma
151494_at 153135_at 154994_at 152298_at 143798_at 152911_at
0.2473773 0.2048103 0.1919025 0.2559847 0.2803217 0.5607609

$cov.coefficients
         Contrasts
Contrasts        PP0       PP5       PP05
     PP0   0.2857143 0.1428571 -0.1428571
     PP5   0.1428571 0.4761905  0.3333333
     PP05 -0.1428571 0.3333333  0.4761905

$stdev.unscaled
           Contrasts
                  PP0       PP5      PP05
  151494_at 0.5345225 0.6900656 0.6900656
```

```
  153135_at 0.5345225 0.6900656 0.6900656
  154994_at 0.5345225 0.6900656 0.6900656
  152298_at 0.5345225 0.6900656 0.6900656
  143798_at 0.5345225 0.6900656 0.6900656
  152911_at 0.5345225 0.6900656 0.6900656

$Amean
151494_at 153135_at 154994_at 152298_at 143798_at 152911_at
11.193236  5.370479  6.019606  7.305581  6.809179  6.127531

$method
[1] "ls"

$design
       P0 P5 Pre
 Pre82  0  0   1
 Pre83  0  0   1
 Pre84  0  0   1
 Pre85  0  0   1
 Pre86  0  0   1
12 more rows ...

$contrasts
      Contrasts
Levels PP0 PP5 PP05
   P0   -1   0    1
   P5    0  -1   -1
   Pre   1   1    0

$df.prior
[1] 5.217904

$s2.prior
[1] 0.04836891

$var.prior
[1] 192.18243 220.45899  51.63242

$proportion
[1] 0.01

$s2.post
 151494_at  153135_at  154994_at  152298_at  143798_at  152911_at
0.05771294 0.04369081 0.03996046 0.06086921 0.07037751 0.24220762

$t
           Contrasts
                  PP0       PP5       PP05
  151494_at  -1.612289 -2.587537 -1.3386634
  153135_at  -6.862251 -5.731546 -0.4160695
  154994_at -10.819459 -9.515379 -1.1346618
  152298_at  -3.539346 -3.597237 -0.8556717
  143798_at  18.012305 23.360360  9.4080888
  152911_at   2.523984  3.740108  1.7850379
```

```
$df.total
[1] 19.2179 19.2179 19.2179 19.2179 19.2179 19.2179

$p.value
           Contrasts
                     PP0          PP5         PP05
  151494_at 1.232006e-01 1.795316e-02 1.963023e-01
  153135_at 1.420509e-06 1.526614e-05 6.819687e-01
  154994_at 1.287067e-09 1.048367e-08 2.704720e-01
  152298_at 2.160290e-03 1.892729e-03 4.027221e-01
  143798_at 1.703125e-13 1.420904e-15 1.256619e-08
  152911_at 2.055433e-02 1.364760e-03 9.004908e-02

$lods
           Contrasts
                  PP0       PP5      PP05
  151494_at -6.571029 -4.650923 -6.050125
  153135_at  4.631505  2.373579 -6.852933
  154994_at 11.859547  9.847255 -6.293435
  152298_at -2.787673 -2.475397 -6.568292
  143798_at 21.050642 25.911322 10.066153
  152911_at -4.962280 -2.152309 -5.407255

$F
[1]   3.566657  29.103067  75.210254   9.179426 320.129095   7.674547

$F.p.value
[1] 4.815750e-02 1.530920e-06 8.159783e-10 1.591277e-03 1.760035e-15
[6] 3.549307e-03
```

```
> results <- decideTests(fit.main)
> summary(results)
```

```
        PP0  PP5 PP05
Down   1438 1268  271
NotSig  483  813 2401
Up      955  795  204
```

```
> topTab_PP0 <- topTable(fit.main, number = nrow(fit.main),coef= 1, adjust="fdr")
>
> topTab_PP5 <- topTable(fit.main, number = nrow(fit.main),coef= 2, adjust="fdr")
>
> topTab_PP05 <- topTable(fit.main, number = nrow(fit.main), coef= 3, adjust="fdr")
>
> head(topTab_PP0)
```

```
              logFC  AveExpr         t      P.Value    adj.P.Val        B
149304_at -5.444965 6.630441 -57.94541 4.844901e-23 1.393394e-19 42.06919
152504_at  4.008227 8.217189  43.29650 1.258222e-20 1.740472e-17 37.18664
141361_at -4.403292 7.196863 -42.47019 1.815513e-20 1.740472e-17 36.84954
154473_at -4.858746 7.139983 -32.88628 2.320730e-18 1.668605e-15 32.25730
152515_at  2.606889 8.393754  32.07716 3.715300e-18 2.137041e-15 31.80061
143276_at  3.020383 9.775735  31.50812 5.208655e-18 2.496682e-15 31.47169
```

```
> head(topTab_PP5)
```

```
             logFC  AveExpr         t      P.Value    adj.P.Val        B
149304_at -4.128635 6.630441 -34.03345 1.213530e-18 3.490112e-15 32.70173
152504_at  3.868532 8.217189  32.36849 3.132152e-18 4.504035e-15 31.81657
142351_at -3.778902 5.598634 -31.52377 5.160103e-18 4.946819e-15 31.34672
141361_at -4.139400 7.196863 -30.92572 7.405827e-18 5.324790e-15 31.00513
154473_at -5.018644 7.139983 -26.31192 1.542669e-16 8.873430e-14 28.08836
152515_at  2.622263 8.393754  24.99339 4.034041e-16 1.745033e-13 27.14989
```

```
> head(topTab_PP05)
```

```
               logFC  AveExpr         t      P.Value    adj.P.Val        B
142351_at   -2.739460 5.598634 -22.85269 2.137331e-15 6.146964e-12 24.56302
149770_at    3.073472 8.040702  15.61446 2.242227e-12 3.224323e-09 18.39766
151630_at    1.707209 8.614367  13.63427 2.471620e-11 2.369460e-08 16.13605
145560_at    1.952235 6.952139  12.58030 9.998356e-11 7.188818e-08 14.79621
151727_s_at  1.840085 5.216161  11.85085 2.780847e-10 1.599543e-07 13.80648
141536_at    1.713911 8.574241  11.56053 4.235522e-10 2.030227e-07 13.39738
```

```
    PROBEID   SYMBOL ENTREZID                                      GENENAME
1 141200_at    Hmg-2    37407                 High mobility group protein 2
2 141204_at GCS2beta    35056                    Glucosidase 2 beta subunit
3 141205_at    Naa60    39142                 N(alpha)-acetyltransferase 60
4 141217_at     Dak1    43165                                          Dak1
5 141218_at   Hrb27C    33968 Heterogeneous nuclear ribonucleoprotein at 27C
6 141219_at   CG7099    34753                       uncharacterized protein
        logFC  AveExpr           t      P.Value    adj.P.Val          B
1 -0.433568424 5.309338 -4.55473219 2.109591e-04 3.891714e-04 -0.4666475
2 -0.009620201 6.409605 -0.06801962 9.464727e-01 9.487820e-01 -7.8490369
3 -0.495252140 6.075200 -3.61069618 1.835382e-03 2.767991e-03 -2.6269061
4  1.138547596 6.471827  9.46492205 1.141421e-08 8.969201e-08  9.6046441
5  0.511630924 3.104276  4.40141481 2.994030e-04 5.302235e-04 -0.8190249
6 -0.458122212 5.289664 -4.70309625 1.505203e-04 2.901450e-04 -0.1261150
```

```
    PROBEID   SYMBOL ENTREZID                                      GENENAME
1 141200_at    Hmg-2    37407                 High mobility group protein 2
2 141204_at GCS2beta    35056                    Glucosidase 2 beta subunit
3 141205_at    Naa60    39142                 N(alpha)-acetyltransferase 60
4 141217_at     Dak1    43165                                          Dak1
5 141218_at   Hrb27C    33968 Heterogeneous nuclear ribonucleoprotein at 27C
6 141219_at   CG7099    34753                       uncharacterized protein
       logFC  AveExpr           t   P.Value adj.P.Val         B
1 -0.02163003 5.309338 -0.17601029 0.8621273 0.9391963 -6.926620
2 -0.27251975 6.409605 -1.49253207 0.1517950 0.3555068 -5.844756
3  0.11005461 6.075200  0.62151049 0.5415669 0.7356988 -6.743438
4 -0.36302444 6.471827 -2.33763843 0.0303663 0.1240228 -4.439218
5  0.01102236 3.104276  0.07344896 0.9422080 0.9743941 -6.939942
6  0.05037758 5.289664  0.40060418 0.6931302 0.8437494 -6.859458
```

```
    PROBEID   SYMBOL ENTREZID                                      GENENAME
```

```
1 141200_at    Hmg-2    37407                  High mobility group protein 2
2 141204_at GCS2beta    35056                     Glucosidase 2 beta subunit
3 141205_at    Naa60    39142                  N(alpha)-acetyltransferase 60
4 141217_at     Dak1    43165                                           Dak1
5 141218_at   Hrb27C    33968 Heterogeneous nuclear ribonucleoprotein at 27C
6 141219_at   CG7099    34753                       uncharacterized protein
       logFC  AveExpr         t      P.Value    adj.P.Val          B
1 -0.4551985 5.309338 -3.704091 1.482168e-03 0.0032220077 -2.2339595
2 -0.2821400 6.409605 -1.545220 1.386016e-01 0.1705812137 -6.4838646
3 -0.3851975 6.075200 -2.175323 4.228584e-02 0.0580219769 -5.4451720
4  0.7755232 6.471827  4.993859 7.802709e-05 0.0002804452  0.7163636
5  0.5226533 3.104276  3.482770 2.457768e-03 0.0049351896 -2.7325050
6 -0.4077446 5.289664 -3.242399 4.239304e-03 0.0077900726 -3.2658975
```

RESULTADOS Y DISCUSIÓN

Visualización de la expresión diferencial

Una visualitzación de los resultados es posible con los graficos tipo volcan. Estos gráficos muestran los genes con p-valores muy altos y por tanto con probabilidades elevadas de expresarse diferencialmente. Los genes más alejados del centro del gráfico son los más significativos, esto es, los primeros de la lista de selección de genes. Esto es debido a que el efecto biológico se representa en el eje de abscisas, al representar los cambios de expresión en escala logarítmica. En el eje ordenadas se representa el efecto estadístico al representar el logaritmo negativo de los p-valores.

```
> geneSymbols <- select(drosgenome1.db, rownames(fit.main), c("SYMBOL"))
> SYMBOLS <- geneSymbols$SYMBOL
```

```
> volcanoplot(fit.main, coef = 1, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
```

```
> volcanoplot(fit.main, coef = 2, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
```

```
> volcanoplot(fit.main, coef = 3, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
```

Comparaciónes multiples y visualización

A continuación, se muestran el número de genes que se expresan diferencialmente para las tres comparaciones realizadas en forma de table y de manera gráfica con un diagrama de Venn.

En el diagrama de Venn podemos observar que 14 de los genes significativos son comunes en los tres grupos, de los 614 regulados “Down” y de los 451 regulados “up”, pero no se puede saber si son genes regulados “up” o “down”.

Los heatmaps son otra forma de visualizar estas comparaciones . En este tipo de gráficos se utiliza una paleta de colores para diferenciar los niveles de expresión.

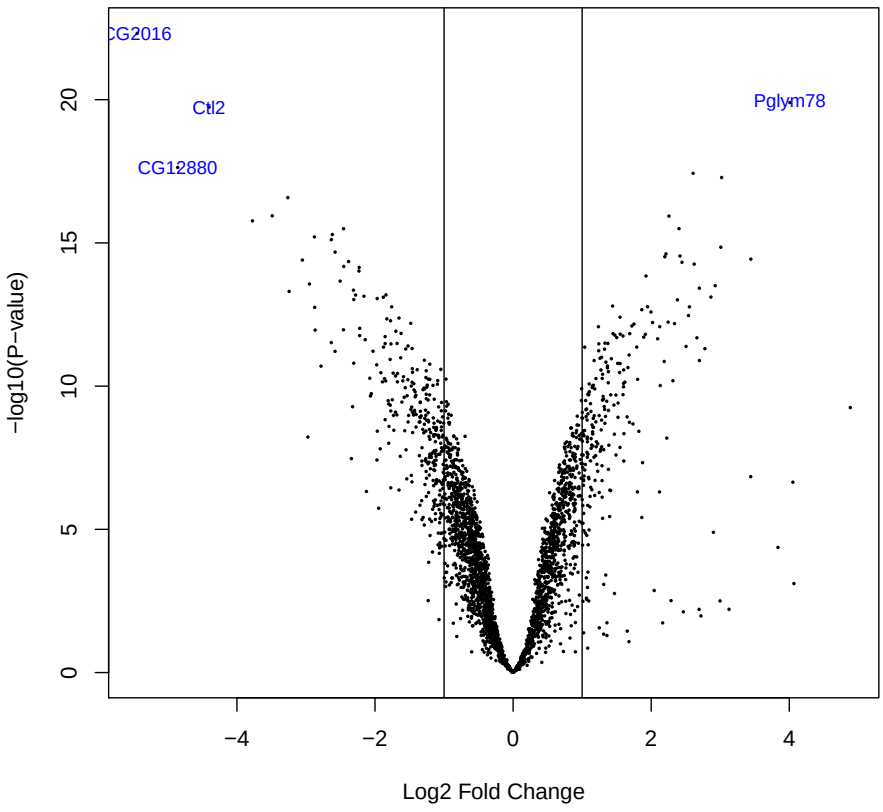


Figure 12: Genes expresados diferencialmente para PP0

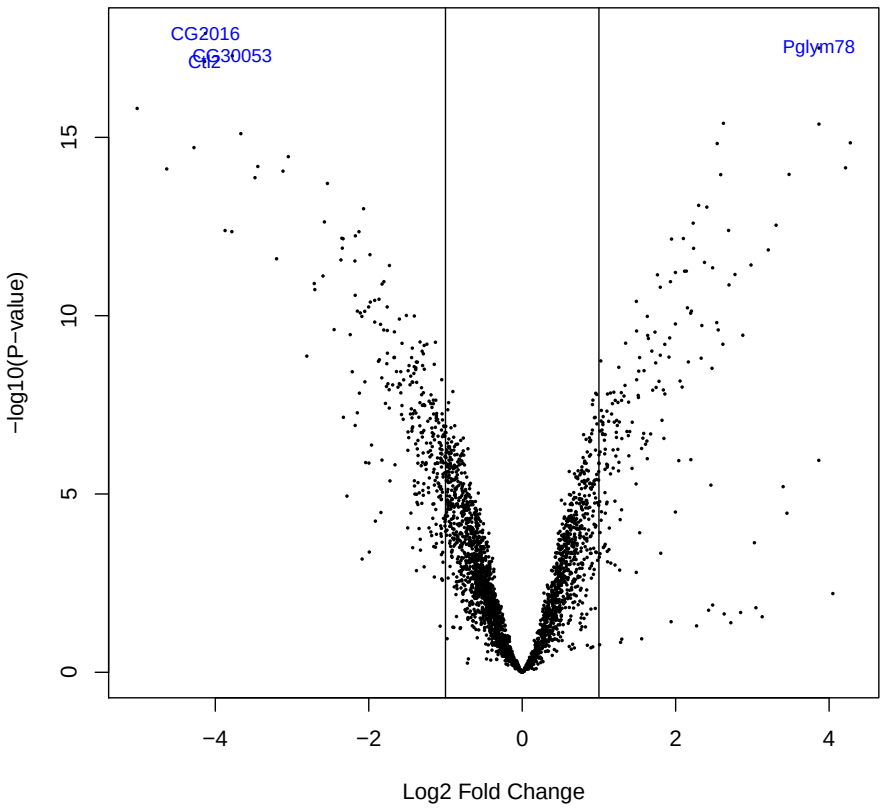


Figure 13: Genes expresados diferencialmente para PP5

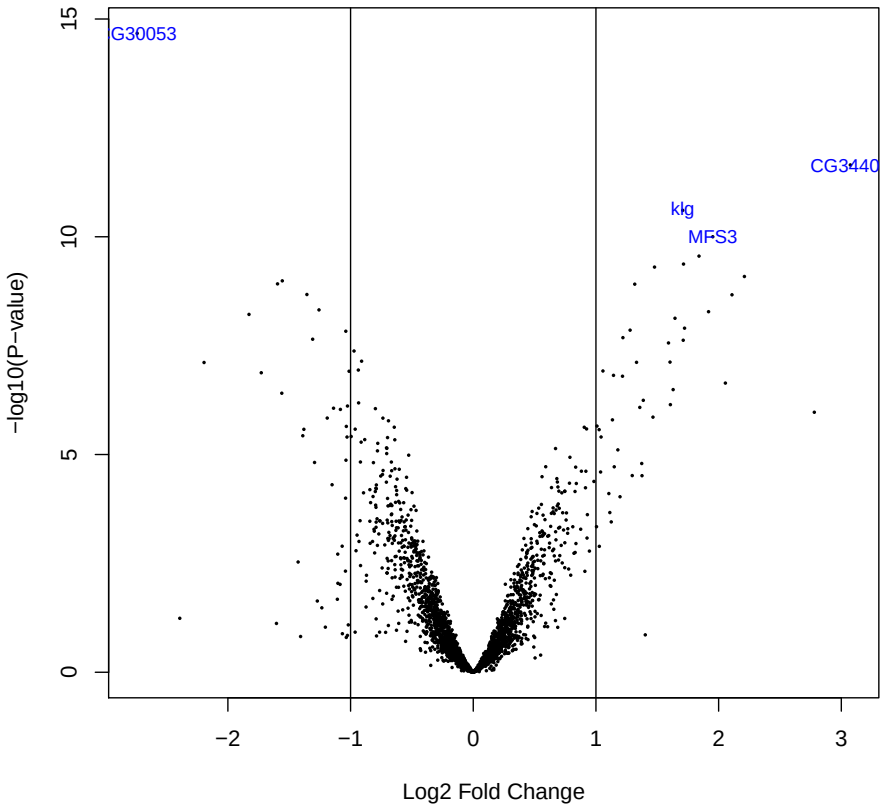


Figure 14: Genes expresados diferencialmente para P05

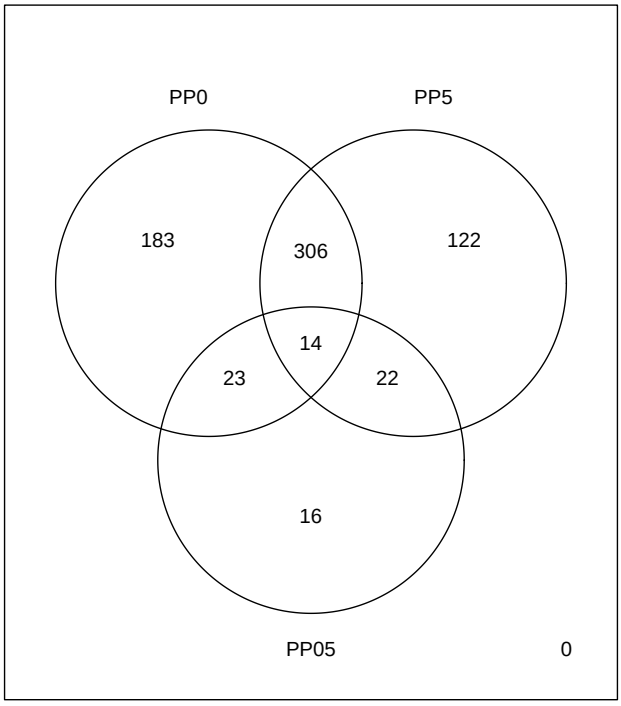


Figure 15: Diagrama de Venn de los genes en común

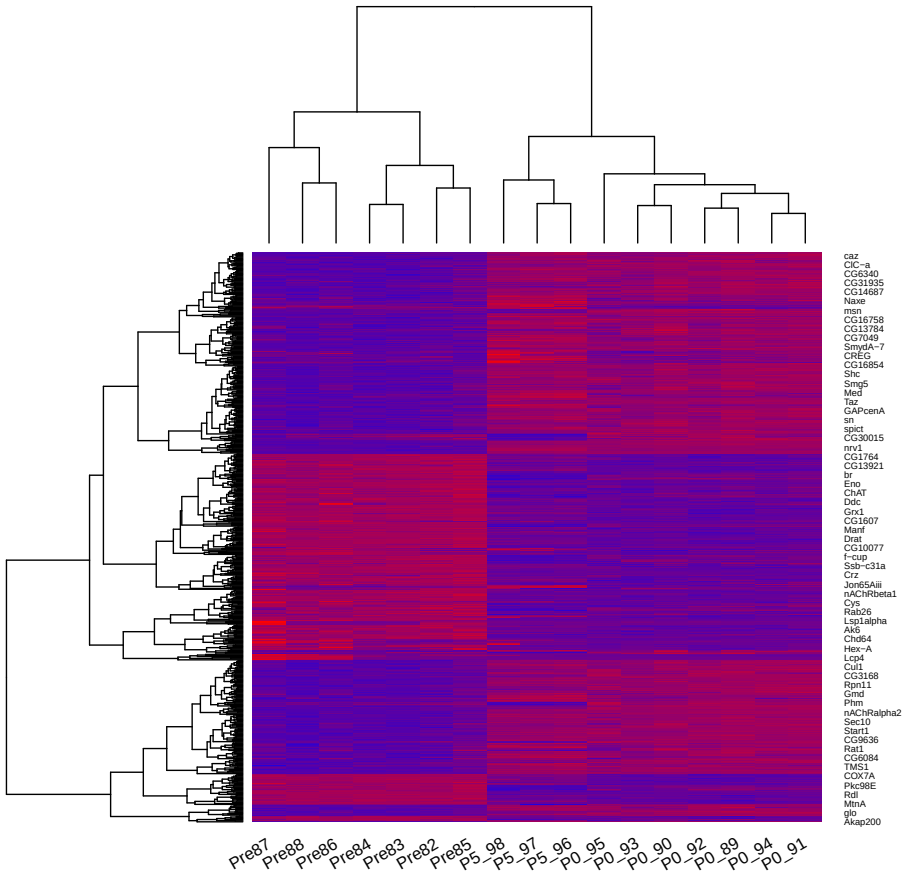


Figure 16: Genes expresados diferencialmente

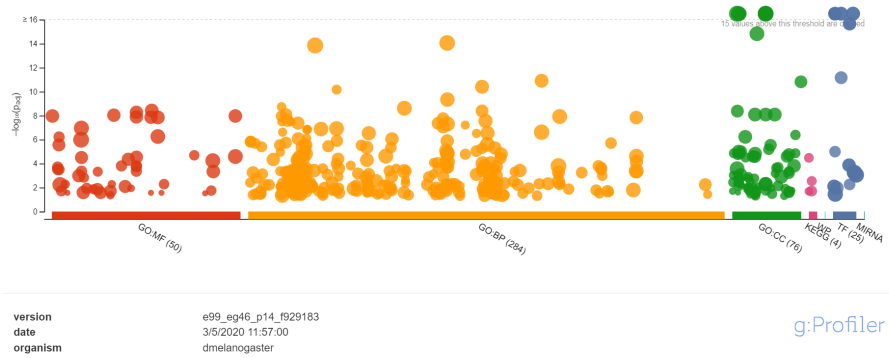
Significación biológica

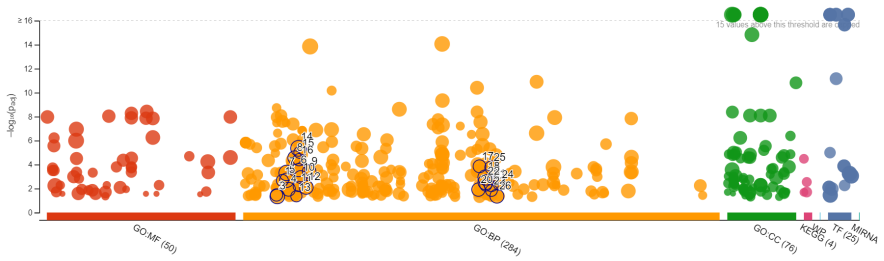
```
 PP0  PP5 PP05
2565 2310  766
```

```
> listOfData <- listOfSelected[1:3]
> comparisonsNames <- names(listOfData)
> universe <- mapped_genes
> for (i in 1:length(listOfData)) {
+   genesIn <- listOfData[[i]]
+   comparison <- comparisonsNames[i]
+   enrich.result <- enrichPathway(gene = genesIn,
+                                  pvalueCutoff = 0.05,
+                                  readable = T,
+                                  pAdjustMethod = "BH",
+                                  organism = "fly",
+                                  universe = universe)
+
+  }
> print(genesIn)
> print(genesIn[1001:2310])
> print(genesIn[2001:2310])
> str(genesIn)
> #write.csv2(as.data.frame(enrich.result), file.path("./results/enrich_results.cvs"))
> #comparison<- comparisonsNames[1]
> #cnetplot(enrich.result, categorySize="geneNum", showCategory = 3,vertex.label.cex = 0.2)
```

Debido a errores en la función de "Gene Enrichment Analysis" se ha realizado el estudio de significación biologica con la herremienta g:Profiler y estos han sido los resultados:

La herramienta que se ha utilizado es g: GOSt que realiza análisis de enriquecimiento estadístico para encontrar una significación biológica. El primer resultado visible en g: GOSt es un diagrama interactivo de Manhattan que ilustra los resultados del análisis de enriquecimiento. El eje x representa términos funcionales que están agrupados y codificados por colores por fuentes de datos (la función molecular de GO es roja; los componentes celulares GO en de color verde, los procesos celulares Go en color naranja; rutas biológicas en rosa; motivos reguladores de DNA en azul y las fuentes que no se incluyeron en el análisis se muestran en gris). El eje y muestra los valores p de enriquecimiento ajustados en la escala negativa log10. Estadísticamente, los valores p más pequeños que 10 ˆ -16 son altamente significativos y son los que tiene en cuenta la herramienta. De nuestra lista se han obviado 15 términos. Podemos observar que los términos más abundantes(mayor número de genes) se encuentran en la sección de procesos celulares donde si nos fijamos hay dos zonas donde los términos son muy cercanos lo que significa que pertenecen a la misma subrama y que son estadísticamente significativos (destacados con círculos negros y detallados en la lista subyacente).





ID	Source	Term ID	Term Name	p_{adj} (query_1)
1	GO:BP	GO:0006511	ubiquitin-dependent protein catabolic process	2.122×10^{-3}
2	GO:BP	GO:0006082	organic acid metabolic process	3.435×10^{-2}
3	GO:BP	GO:0006139	nucleobase-containing compound metabolic proc...	4.585×10^{-2}
4	GO:BP	GO:0006950	response to stress	1.148×10^{-2}
5	GO:BP	GO:0006810	transport	2.032×10^{-3}
6	GO:BP	GO:0008104	protein localization	2.922×10^{-4}
7	GO:BP	GO:0006807	nitrogen compound metabolic process	4.991×10^{-4}
8	GO:BP	GO:0007399	nervous system development	2.643×10^{-5}
9	GO:BP	GO:0009888	tissue development	3.800×10^{-4}
10	GO:BP	GO:0009057	macromolecule catabolic process	1.090×10^{-3}
11	GO:BP	GO:0008063	Toll signaling pathway	6.635×10^{-3}
12	GO:BP	GO:0009653	anatomical structure morphogenesis	7.852×10^{-3}
13	GO:BP	GO:0007528	neuromuscular junction development	4.421×10^{-2}
14	GO:BP	GO:0008152	metabolic process	4.397×10^{-6}
15	GO:BP	GO:0008380	RNA splicing	9.660×10^{-6}
16	GO:BP	GO:0008219	cell death	3.938×10^{-5}
17	GO:BP	GO:0048666	neuron development	1.393×10^{-4}
18	GO:BP	GO:0050789	regulation of biological process	1.194×10^{-3}
19	GO:BP	GO:0050808	synapse organization	5.217×10^{-3}
20	GO:BP	GO:0048583	regulation of response to stimulus	1.222×10^{-2}
21	GO:BP	GO:0051234	establishment of localization	4.525×10^{-3}
22	GO:BP	GO:0050794	regulation of cellular process	2.915×10^{-3}
23	GO:BP	GO:0051240	positive regulation of multicellular organismal pro...	1.455×10^{-2}
24	GO:BP	GO:0051960	regulation of nervous system development	4.377×10^{-3}
25	GO:BP	GO:0051239	regulation of multicellular organismal process	1.735×10^{-4}
26	GO:BP	GO:0051704	multi-organism process	4.315×10^{-2}

version	e99_eg46_p14_f929183
date	3/5/2020 12:35:37
organism	dmelanogaster

g:Profiler

Resumen de resultados

```
> Result_Files <- dir("./results/")
> knitr::kable(
+   Result_Files, booktabs = TRUE,
+   caption = "Listado de ficheros de resultados",
+   col.names="Ficheros"
+ )
```

Table 2: Listado de ficheros de resultados

Ficheros
Annotated_P05.csv
Annotated_PP0.csv
Annotated_PP5.csv
datos_norm.csv
enrich_results.cvs
gprofiler_dmelanogaster.ENSG.zip
gprofiler_dmelanogaster.name.zip
heatmap.csv
Norm_data_filt
Norm_data_filt.csv
rawData_quality
rawDataNorm_quality

APENDICE

```
> library(knitr)
> knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
+                       comment = NA, prompt = TRUE, tidy = FALSE,
+                       fig.width = 7, fig.height = 7, fig_caption = TRUE,
+                       cache=FALSE)
>
> setwd(".")
> dir.create("data")
> dir.create("results")
> dir.create("figuras")
>
> library(ggplot2)
> library(ggrepel)
> library(oligo)
> library(Biobase)
> library(GEOquery)
> library(arrayQualityMetrics)
> library(pvca)
> library(genefilter)
> library(limma)
> library(gplots)
> library(ReactomePA)
> library(BiocGenerics)
> library(BiocParallel)
> library(BiocManager)
> library(tinytex)
> library(genefilter)
>
> targets <- read.csv2("./data/targets.csv", header = TRUE, sep = ",")
> knitr::kable(
+   targets, booktabs = TRUE,
+   caption = 'Contenido del fichero *targets.csv* ')
> celFiles <- list.celfiles("./data", full.names = TRUE)
>
> my.targets <-read.AnnotatedDataFrame(file.path("./data","targets.csv"),
+                                      header = TRUE, row.names = 1,
+                                      sep=",")
> rawData <- read.celfiles(celFiles, phenoData = my.targets)
>
> my.targets@data$ShortName->rownames(pData(rawData))
>
> colnames(rawData)<-rownames(pData(rawData))
>
> head(rawData)
>
> arrayQualityMetrics(rawData, outdir = file.path("./results","rawData_quality"), force = T)
> knitr::include_graphics("figuras/ResumQM.png")
> labels <- c("Pre","P0","P5")
>
> boxplot(rawData, cex.axis = 0.5, las=2, which="all", main="Boxplot de valores de intensidad",
+         col = c(rep("blue", 7), rep("purple", 7), rep("green", 3)))
```

```
> legend("topleft", labels,fill= c("blue","purple","green"), bty="n")
> plot(hclust(dist(t(exprs(rawData)))))
> plotPCA <- function(datos, labels, factor, scale,
+                     colores, size = 1.5, glineas = 0.25) {
+   data <- prcomp(t(datos), scale = scale)
+   #plot adjustments
+   dataDF <- data.frame(data$x)
+   Group <- factor
+   loads <- round(data$sdev^2/sum(data$sdev^2)*100, 1)
+   #main plot
+   p1 <- ggplot(dataDF, aes(x=PC1, y =PC2)) +
+     theme_classic() +
+     geom_hline(yintercept = 0, color = "gray70") +
+     geom_vline(xintercept = 0, color = "gray70") +
+     geom_point(aes(color = Group), alpha = 0.55, size = 3) +
+     coord_cartesian(xlim = c(min(data$x[,1])-5, max(data$x[,1])+5)) +
+     scale_fill_discrete(name = "Group")
+   #avoiding labels superposition
+   p1 + geom_text_repel(aes(y = PC2 + 0.25, label = labels), segment.size = 0.25, size = size) +
+     labs(x = c(paste("PC1", loads[1], "%")), y = c(paste("PC2", loads[2], "%"))) +
+     theme(plot.title = element_text(hjust = 0.5)) +
+     scale_colour_manual(values = colores)
+ }
> plotPCA(exprs(rawData), labels = targets$ShortName, factor = targets$Grupo,
+         scale = F, size = 3, colores = c("blue","purple","green"))
> eset_rma<- rma(rawData)
> eset_rma
> write.csv2(exprs(eset_rma), file = "./results/datos_norm.csv")
> arrayQualityMetrics(eset_rma, outdir = file.path("./results", "rawDataNorm_quality"), force = T)
> knitr::include_graphics("figuras/ResumQM_norm.png")
>
> labels <- c("Pre","P0","P5")
>
> boxplot(eset_rma, cex.axis = 0.5, las=2, which="all", main="Boxplot de valores de intensidad datos no
+         col = c(rep("blue", 7), rep("purple", 7), rep("green", 3)))
> legend("topleft", labels,fill= c("blue","purple","green"), bty="n")
>
>
>
> plot(hclust(dist(t(exprs(eset_rma)))))
>
>
>
> plotPCA(exprs(eset_rma), labels = targets$ShortName, factor = targets$Grupo,
+         scale = F, size = 3, colores = c("blue","purple","green"))
>
>
>
>
> ## Selección de genes
>
> sds <- apply(exprs(eset_rma), 1, sd)
> sds0 <- sort(sds)
```

```
> plot(1:length(sds0), sds0, xlab="Genes desde el menos al más variable", ylab="Desviación estándar")
> abline(v=length(sds)*c(0.9,0.95))
>
>
> ## Anotaciones
>
>
>
>
> library(drosgenome1.db)
> annotation(eset_rma)<-"drosgenome1.db"
>
>
>
> ## Filtraje de genes
>
>
>
>
> filtered <- nsFilter(eset_rma,require.entrez = TRUE, remove.dupEntrez = TRUE,
+                      var.filter = TRUE, var.func = IQR, var.cutoff = 0.75,
+                      filterByQuantile = T, feature.exclude = "^AFFX")
>
> print(filtered$filter.log)
> eset_filt<- filtered$eset
> write.csv2(exprs(eset_filt), file="./results/Norm_data_filt.csv")
>
>
> ## Control de calidad dato filtrados
>
> arrayQualityMetrics(eset_filt, outdir = file.path("./results", "Norm_data_filt"), force= T)
>
> knitr::include_graphics("figuras/ResumQM_filt.png")
>
> labels <- c("Pre","P0","P5")
>
> boxplot(eset_filt, cex.axis = 0.5, las=2, which="all", main="Boxplot de valores de intensidad datos fi
+         col = c(rep("blue", 7), rep("purple", 7), rep("green", 3)))
> legend("topleft", labels,fill= c("blue","purple","green","yellow"), bty="n")
>
> plot(hclust(dist(t(exprs(eset_filt)))))
>
> plotPCA(exprs(eset_filt), labels = targets$ShortName, factor = targets$Grupo,
+         scale = F, size = 3, colores = c("blue","purple","green"))
>
> ## Diseño matriz experimental
>
> designMat<- model.matrix(~0+Grupo, pData(eset_filt))
>
> colnames(designMat) <- c("P0","P5","Pre")
> print(designMat)
>
> cont.matrix<- makeContrasts(PP0= Pre-P0, PP5= Pre-P5, PP05=P0-P5, levels = designMat)
```

```
> cont.matrix
>
> ## Selección de genes diferencialmente expresados y anotación
>
> fit<-lmFit(eset_filt, designMat)
> fit.main<-contrasts.fit(fit, cont.matrix)
> fit.main<-eBayes(fit.main)
> head(fit.main)
> results <- decideTests(fit.main)
> summary(results)
>
> topTab_PP0 <- topTable(fit.main, number = nrow(fit.main),coef= 1, adjust="fdr")
>
> topTab_PP5 <- topTable(fit.main, number = nrow(fit.main),coef= 2, adjust="fdr")
>
> topTab_PP05 <- topTable(fit.main, number = nrow(fit.main), coef= 3, adjust="fdr")
>
> head(topTab_PP0)
> head(topTab_PP5)
> head(topTab_PP05)
>
> annotatedTopTable <- function(topTab, anotPackage)
+ {
+   topTab <- cbind(PROBEID=rownames(topTab), topTab)
+   myProbes <- rownames(topTab)
+   thePackage <- eval(parse(text = anotPackage))
+   geneAnots <- select(thePackage, myProbes, c("SYMBOL", "ENTREZID", "GENENAME"))
+   annotatedTopTab <- merge(x=geneAnots, y=topTab, by.x="PROBEID", by.y="PROBEID")
+   return(annotatedTopTab)
+ }
>
> topAnnotated_PP0 <- annotatedTopTable(topTab_PP0,
+                                       anotPackage = "drosgenome1.db")
> topAnnotated_PP5 <- annotatedTopTable(topTab_PP5,
+                                         anotPackage = "drosgenome1.db")
> topAnnotated_P05 <- annotatedTopTable(topTab_PP05,
+                                      anotPackage = "drosgenome1.db")
> head(topAnnotated_PP0)
> head(topAnnotated_P05)
> head(topAnnotated_PP5)
>
> write.csv(topAnnotated_PP0, file="./results/Annotated_PP0.csv")
> write.csv(topAnnotated_PP5, file="./results/Annotated_PP5.csv")
> write.csv(topAnnotated_P05, file="./results/Annotated_P05.csv")
>
>
> # RESULTADOS
>
> ## Visualización de la expresión diferencial
>
> geneSymbols <- select(drosgenome1.db, rownames(fit.main), c("SYMBOL"))
> SYMBOLS <- geneSymbols$SYMBOL
>
```

```
> volcanoplot(fit.main, coef = 1, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
>
>
>
> volcanoplot(fit.main, coef = 2, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
>
>
> volcanoplot(fit.main, coef = 3, highlight = 4, names = SYMBOLS)
> abline(v=c(-1,1))
>
> ## Comparaciónes multiples y visualización
>
>
> res <- decideTests(fit.main, method = "separate", adjust.method = "fdr", p.value = 0.1, lfc = 1)
> sum.res.rows <- apply(abs(res),1,sum)
> res.selected <- res[sum.res.rows!=0,]
> print(summary(res))
>
>
> vennDiagram(res.selected[, 1:3], cex = 0.9)
>
> probesInHeatmap <- rownames(res.selected)
> HMdata <- exprs(eset_filt)[rownames(exprs(eset_filt)) %in% probesInHeatmap,]
> geneSymbols <- select(drosgenome1.db, rownames(HMdata), c("SYMBOL"))
> SYMBOLS <- geneSymbols$SYMBOL
> rownames(HMdata) <- SYMBOLS
> my_palette <- colorRampPalette(c("blue", "red"))(n =299)
> write.csv2(HMdata, file=file.path("./results/heatmap.csv"))
>
>
> heatmap.2(HMdata, Rowv = T, Colv = T,
+           scale = "row", col = my_palette, sepcolor = "white",
+           sepwidth = c(0.05,0.05), cexRow = 0.5, cexCol = 0.9,
+           key = F, density.info = "histogram",
+           tracecol = NULL, dendrogram = "both", srtCol = 30)
>
> ## Significación biológica
>
>
> listOfTables <- list(PP0 = topTab_PP0,
+                      PP5 = topTab_PP5,
+                      PP05 = topTab_PP05)
> listOfSelected <- list()
> for(i in 1:length(listOfTables)){
+   topTab <- listOfTables[[i]]
+   whichGenes <- topTab["adj.P.Val"]<0.15
+   selectedIDs <- rownames(topTab)[whichGenes]
+   EntrezIDs <- select(drosgenome1.db, selectedIDs, c("ENTREZID"))
+   EntrezIDs <- EntrezIDs$ENTREZID
+   listOfSelected[[i]] <- EntrezIDs
+   names(listOfSelected)[i] <- names(listOfTables)[i]
```

```
+ }
> sapply(listOfSelected, length)
>
>
> library(org.Hs.eg.db)
> mapped_genes2GO <- mappedkeys(org.Hs.egGO)
> mapped_genes2KEGG <- mappedkeys(org.Hs.egPATH)
> mapped_genes <- union(mapped_genes2GO, mapped_genes2KEGG)
```

REFERENCIAS

Gonzalo Sanz, Ricardo, and Alex Sánchez-Pla. 2019. "Statistical Analysis of Microarray Data." In *Microarray Bioinformatics*, edited by Verónica Bolón-Canedo and Amparo Alonso-Betanzos, 87–121. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4939-9442-7_5.

Hoopfer ED, Watts RJ, Penton A. n.d. "Genomic Analysis of Drosophila Neuronal Remodeling: A Role for the Rna-Binding Protein Boule as a Negative Regulator of Axon Pruning." *J Neurosci* 28(24)::6092–103.

Irizarry, B, R. A.; Hobbs. 2003. "Exploration, Normalization, and Summaries of High Density Oligonucleotide Array Probe Level Data." *Biostatistics* 4 (2): 249–64. https://doi.org/10.1093/biostatistics/4.2.249.