Genómica Funcional y Análisis de Microarrays

PEC 1- Primera Prueba de Evaluación Continua

Author: Ramón Tamarit Agusti

Análisis estadístico de microarrays del experimento:

Molecular basis of age-associated cytokine dysregulation in LPS-stimulated macrophages

R. Lakshman Chelvarajan, Yushu Liu,‡ Diana Popa, Marilyn L. Getchell,
Thomas V. Getchell,,¶ Arnold J. Stromberg, and Subbarao Bondada.

Resumen

Los humanos y roedores adultos son susceptibles a la infección por las bacterias de estreptococo pneumoniae, como consecuencia de la incapacidad de generar anticuerpos para los polisacáridos capsulares. Esto es debido en parte a la:

- *producción reducida de Cytokines proinflamatorias y al*
- *incemento de producción de interleukina por los macrófagos (un tipo de células intestinales) de ratones.*

Para comprobar esta hipótesis se realiza un estudio de microarrays en poblaciones de macrofagos jóvenes y envejecidos (de ratón), paralelamente a una estimulación con Lipopolisacaridos (LPS).

El estudio original de los autores mediante anova de 2x2 factores con un nivel de significación de $p<0.01$, demuestra que 853 genes están regulados por LPS (169 en jóvenes, 184 en adultos y 500 en ambos).

Introducción

El objetivo de este estudio particular, no es reproducir el análisis original de los autores del artículo, sino realizar un análisis sencillo aplicando técnicas comunes de microarrays. Igualmente se intenta comparar los resultados obtenidos mediante distintas metodologías a efectos de comprobar si existen diferencias apreciables entre los resultados.

El estudio esta dividido en dos partes,

a) Estudio sencillo con un único factor, efecto del LPS.

b) Estudio complejo de dos factores , Edad x Tratamiento

Los objetivos biológicos de los análisis estadísticos son,

- Estudiar la expresión diferencial por la acción de LPS,
- Estudiar la expresión diferencial por efecto de la Edad,
- Comprobar si existe un efecto de interacción LPS-Edad.

Descripción del experimento

Microarrays

Los microarrays son de la marca Affymetrix, en concreto el modelo utilizado es el Affymetrix GeneChip Mouse Genome 430 2.0(1). El fichero de anotaciones se puede descargar de la pagina de Affymetrix.

Diseño experimental

Se hibridizan 12 microarrays, que se identifican mediante los nombres y condiciones de la Tabla 1. Los factores son Edad y Trat (tratamiento con LPS). Los niveles de cada factor son dos:

Aged – Young: Envejecido o joven,

LPS – MED: Tratado con LPS o tratado en medio sin LPS.

ExperimentNames	FileName	Edad	Trat
Aged_LPS_80L	Aged_LPS_80L.CEL	Aged	LPS
Aged_LPS_86L	Aged_LPS_86L.CEL	Aged	LPS
Aged_LPS_88L	Aged_LPS_88L.CEL	Aged	LPS
Aged_Medium_81m	Aged_Medium_81m.CEL	Aged	MED
Aged_Medium_82m	Aged_Medium_82m.CEL	Aged	MED
Aged_Medium_84m	Aged_Medium_84m.CEL	Aged	MED
Young_LPS_75L	Young_LPS_75L.CEL	Young	LPS
Young_LPS_76L	Young_LPS_76L.CEL	Young	LPS
Young_LPS_77L	Young_LPS_77L.CEL	Young	LPS
Young_Medium_71m	Young_Medium_71m.CEL	Young	MED
Young_Medium_72m	Young_Medium_72m.CEL	Young	MED
Young_Medium_73m	Young_Medium_73m.CEL	Young	MED

Tabla1: Contenido del fichero “LPSTargets.txt” utilizado en alguno de los cálculos.

El diseño experimental es típicamente factorial con 2x2 factores, dos factores con dos niveles, con dos experimentos por cada combinación (Tabla 2)

		Factor Trat	
		LPS	MED
Factor Edad	Aged	2	2
	Young	2	2

Tabla2: Diseño factorial 2x2 del experimento.

Diseño experimental reducido

El diseño experimental de la Tabla 2, se puede reducir a un solo factor con dos niveles y cuatro experimentos por nivel (tabla 3). Este diseño experimental se usará para comprobar el efecto del LPS con independencia de la Edad.

Factor Trat	
LPS	MED
4	4

Tabla 3: Diseño factorial 1x2 del experimento.

Metodología, programas y flujo de análisis

En todos los procedimientos utilizados se sigue el siguiente flujo de operaciones:

1) Importación y carga de los datos: Se realiza a partir de los ficheros .CEL descargados desde la pagina del experimento (2). Dado que todo el software y rutinas utilizadas se basan en el lenguaje R, la carga se realiza mediante el comando ReadAffy(). En las pruebas preliminares con MeV y para el control de calidad se ha utilizado el programa RMAExpress para trasformar los ficheros al formato RMA. En el caso de las interfaces web utilizadas , RACE, GEPAS y CARMA (3,4,5) usan los comandos de R.
2) Control de calidad y visualización de los datos: Se aplican distintas técnicas para comprobar la calidad de las hibridaciones, en concreto:
    a. RMAexpress: facilita la obtención de los gráficos de pesos e imágenes de los chips.
    b. affyQCreport y affyPLM
3) Normalización: Se emplea en todos los casos el método rma con las siguientes opciones:
    a. Background Adjusting.
    b. Normalizing Using Quantile Normalization.
    c. Summarizing Using Median Polish.

    En este punto se puede guardar el fichero de resultante de la normalización como una matriz de expresión para posteriormente realizar los pasos siguientes. En la figura podemos ver la estructura del mismo:

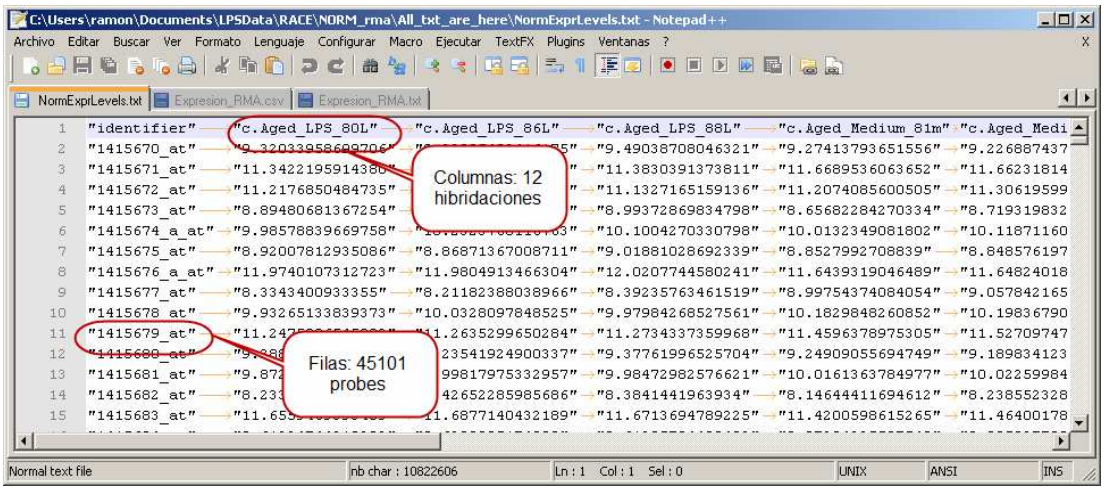


Figura1. Estructura del fichero resultante de la normalización. Matriz de expresión.

4) Filtrado de los datos: cuando es posible se realiza un filtrado los datos para eliminar
    a. los de baja calidad,
    b. los no significativos,
    c. los “probes” de control de affymetrix

Hay que tener en cuenta que no todas las interfaces permiten un control absoluto del filtrado, y en algunas de las interfaces no se puede partir directamente de un fichero de expresión filtrado. En general el filtrado de los datos, no va a influir de forma determinante en los resultados, pero sí en los tiempos de cálculo y en los pesos de los ficheros obtenidos.

Los probes de control son fácilmente detectables por el identificador, ver figura 2.

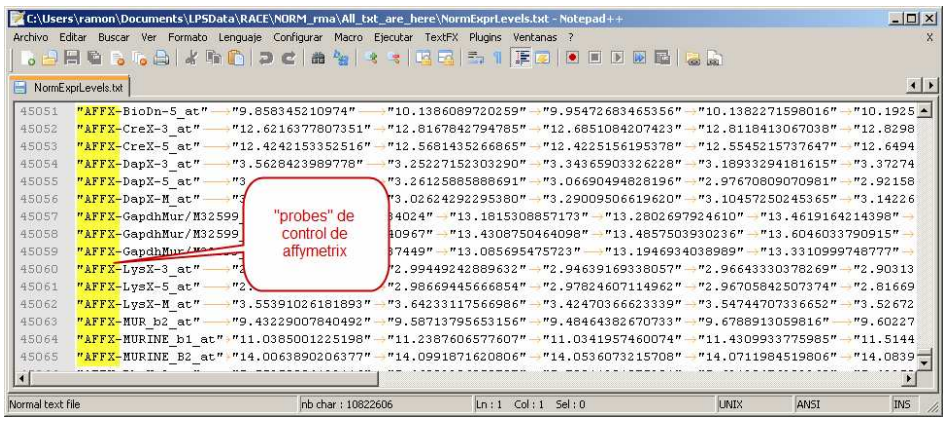


Figura 2. Visualización de los "probes" de control de Affymetrix.

5) Análisis estadístico. En todos los casos se ha utilizado la librería limma para el análisis estadístico del fichero de expresión. Independientemente del número de factores analizados los pasos a realizar son:

   a. Cargar los datos como un objeto "expreSet". Si partimos de un analisis en R, el objeto se obtiene directamente como resultado de la normalización.

   b. Definir la matriz de diseño del modelo: La matriz contiene tantas filas como experimentos y tantas columnas como combinaciones de factores x niveles, (a partir de ahora tratamientos)

   c. Construir una matriz con los nombres que asignaremos a los tratamientos y asignar cada nombre a su correspondiente columna de la matriz de diseño

   d. Ajustar los datos al modelo, obteniendo un objeto de la clase MArrayLM. Los coeficientes ajustados (fit$coef) del modelo lineal son la media logarítmica del valor de la expresión para cada conjunto de probes.

   e. Definir la matriz de contrastes. Es decir, decidir que comparaciones nos interesan de todas las posibles. Los contrastes son combinaciones lineales de los parámetros del modelo de ajuste.

   f. Obtener el ajuste al modelo lineal para los contrastes. Aquí ya obtenemos un conjunto de p-valor para cada comparación. Para incrementar la potencia y reducir el riesgo de los falsos positivos se ajustan también mediante el estadístico eBayes().

   g. Corregir los p-valor para tener en cuenta el problema del "multiple testing". A partir de este punto tenemos ya para cada contraste, con los valores que nos interesan para analizar los datos. Ver figura 3.

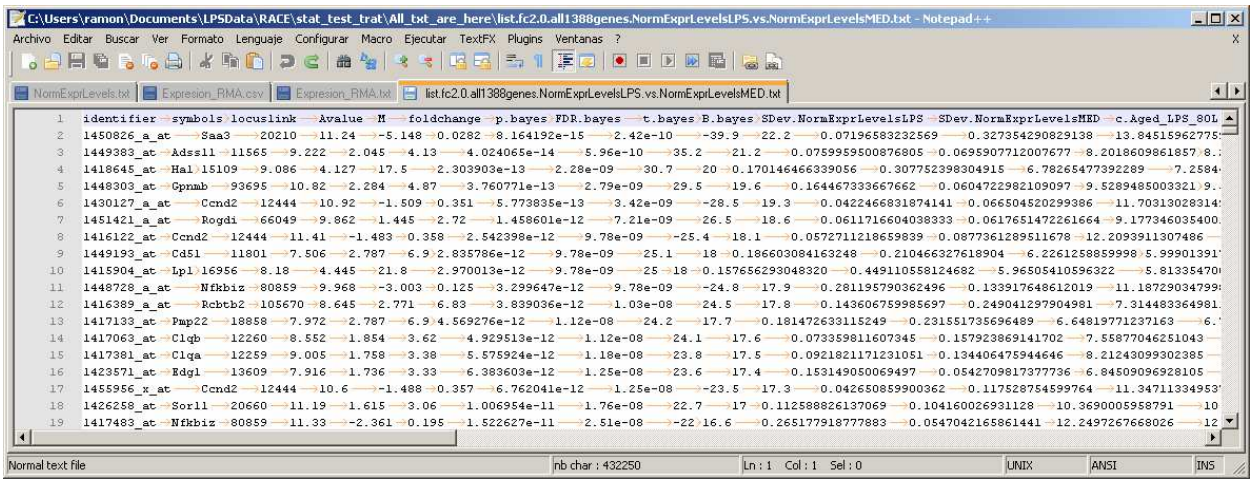


Figura 3. Ejemplo de salida del programa RACE, la tabla incluye todos los valores susceptibles de análisis para cada una de las probes.

h. Analizar los datos y determinar los puntos de corte, para seleccionar los genes diferenciados. Comparar con los datos sin ajustar para comprobar que hemos elegido un punto de corte adecuado.

Control de calidad de los datos con la interface RACE

Entrada de los datos

La interface RACE, permite en unos pocos pasos obtener un conjunto completo de datos utiles para el control de calidad de los chips. Previamente hemos subido al servidor los ficheros .CEL con los datos. Ver figura 4.

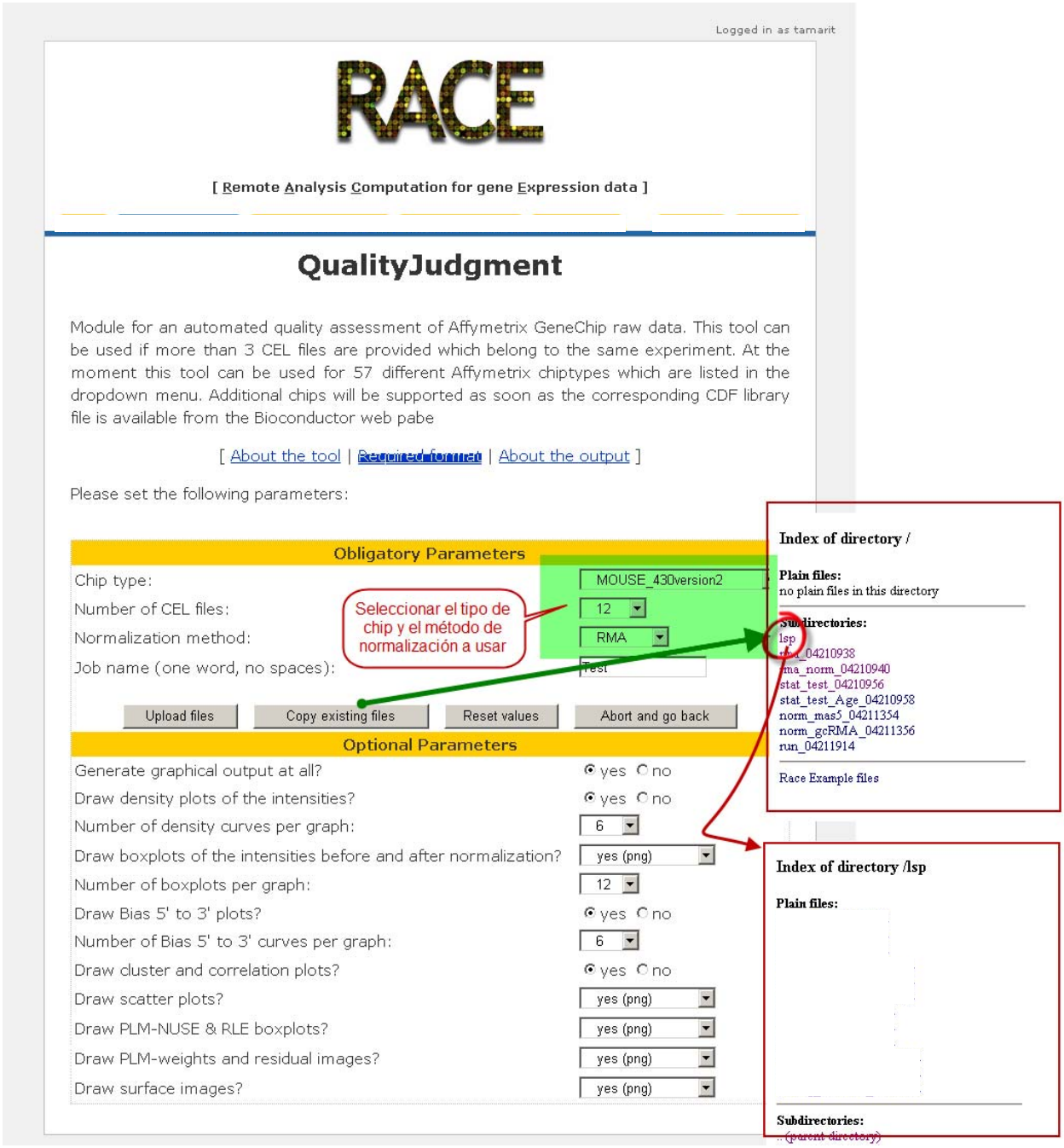


Figura 4. Pagina de entrada de datos para normalización y control de calidad de RACE.

Output de control de calidad y normalización

El Output de control de calidad y normalización devuelve distintos tipos de ficheros (figura 5):

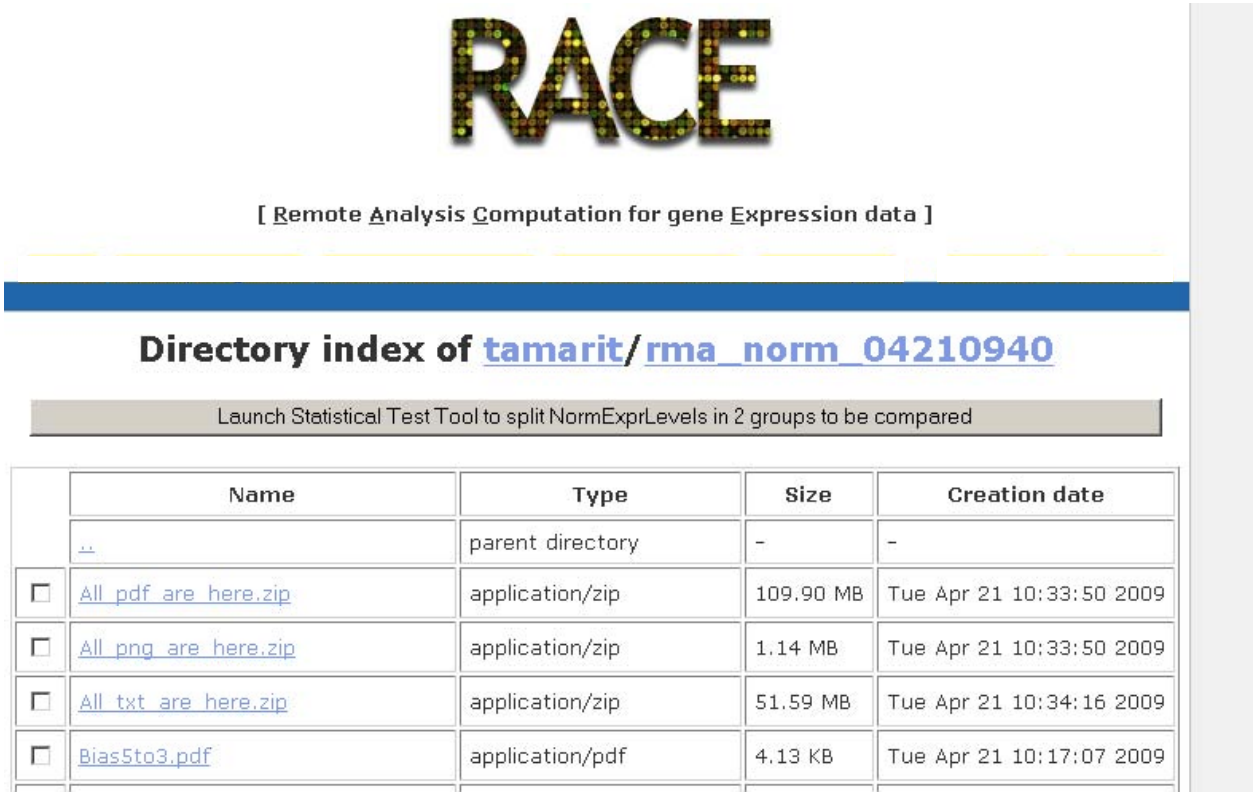


Figura 5. Pagina en donde se almacenan los outputs del proceso de normalización y control de calidad de RACE

En concreto nos interesan:

- Las imágenes png o pdf de los gráficos (individualmente o empaquetadas en zip en todos los casos)
- Los ficheros de expresión normalizados.
- El código en R para reproducir el control de calidad.

En concreto vamos a analizar los siguientes outputs.

Evaluación de las imagenes con color simulado y PLM

Las imágenes en escala de grises (simulación de la densidad) y PLM de los chips no muestran existencia de arañados, polvo o defectos. En la figura 6 adjunto algunas imágenes de los chips.

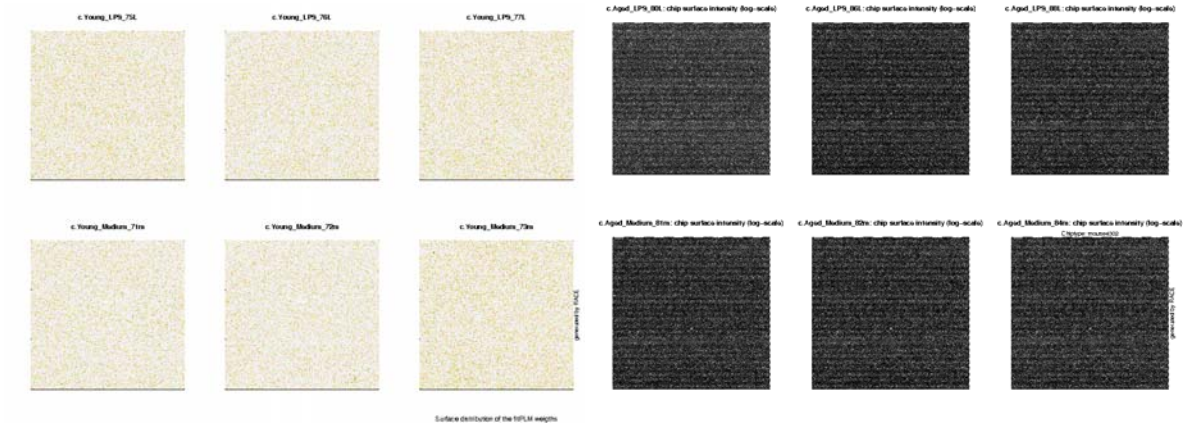


Figura 6.- Imágenes de muestra PLM y en falso color de los chips, se genera un conjunto para cada chip

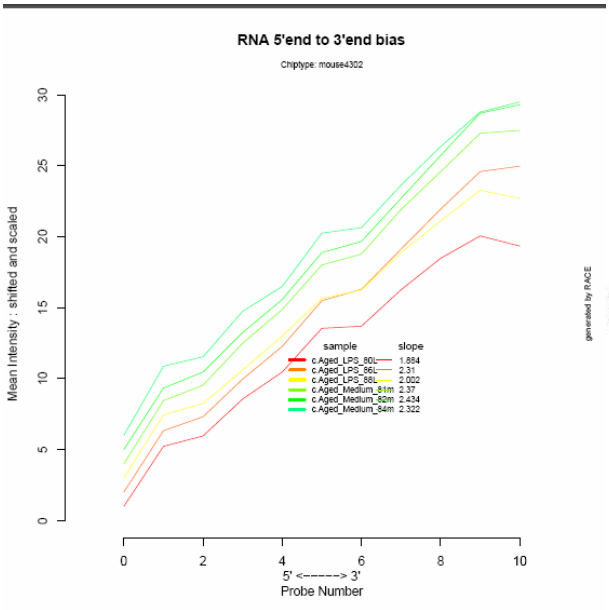


Figura 7. Gráfico de degradación de RNA

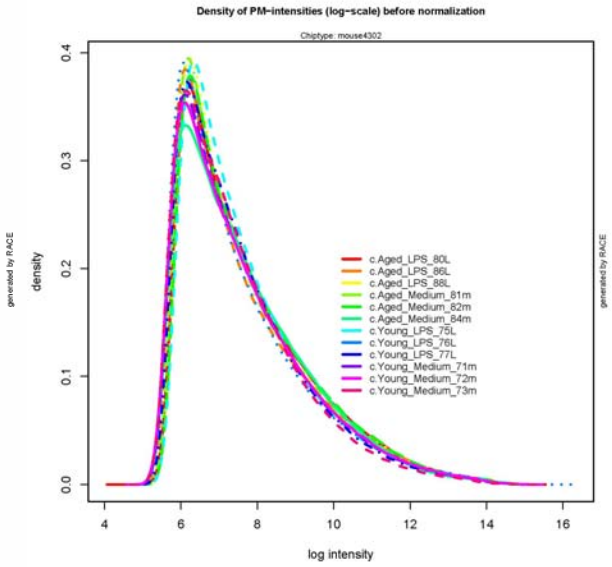


Figura 8. Gráfico de densidades para cada chip

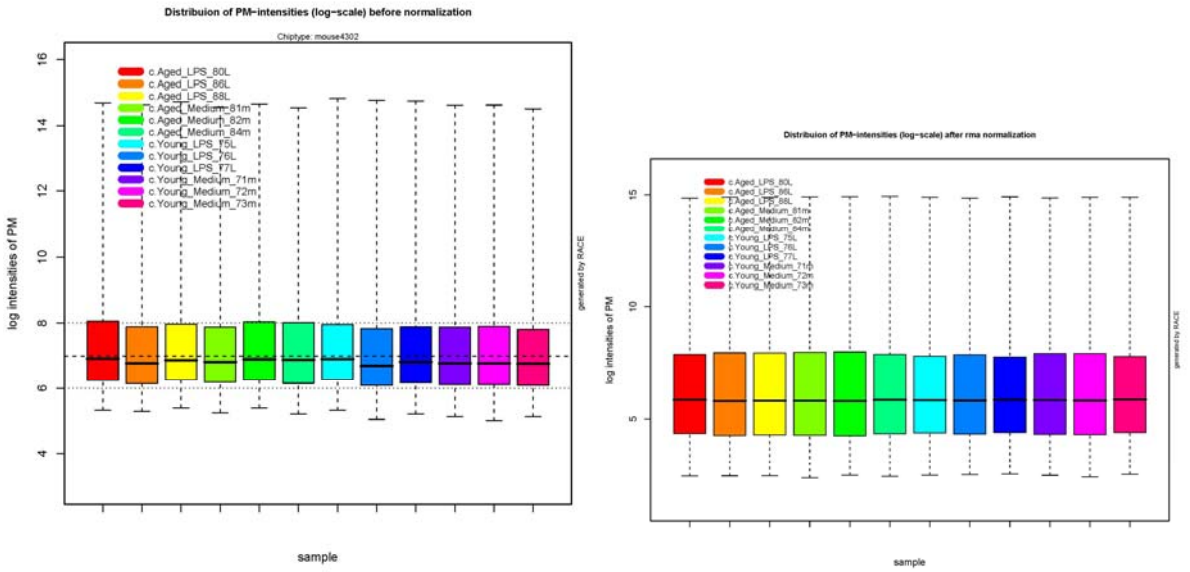


Figura 9. Boxplots de las intensidades antes y después de la normalización .

Evaluación del gráfico de degradación de RNA y densidades

El grafico de degradación de RNA (figura 7) muestra que todos los chips presentan una pendiente similar, por lo que son normales. El grafico de densidades muestra las distribuciones muy centradas y con densidades similares, por lo que el paso de normalización no tendrá efectos muy dramáticos sobre los datos.

Evaluación de los gráficos de control de calidad post - normalización

Dado que es interesante comprobar el efecto de la normalización sobre los datos, se generan distintas graficas que permiten la comparación de los datos antes y después del paso de normalización.

En la figura 9, comprobamos mediante los BoxPlot como las distribuciones de densidad no han cambiado en exceso por el proceso, ya pudimos ver que la calidad de los datos de partida era muy alta.

Mediante la comparación de los niveles de expresión chip a chip, figura 10, comprobamos igualmente que las distribuciones están sobre la diagonal en todos los casos y con dispersiones muy pequeñas. Igualmente, los graficos NUSE y RLE no muestran anomalías en los datos.

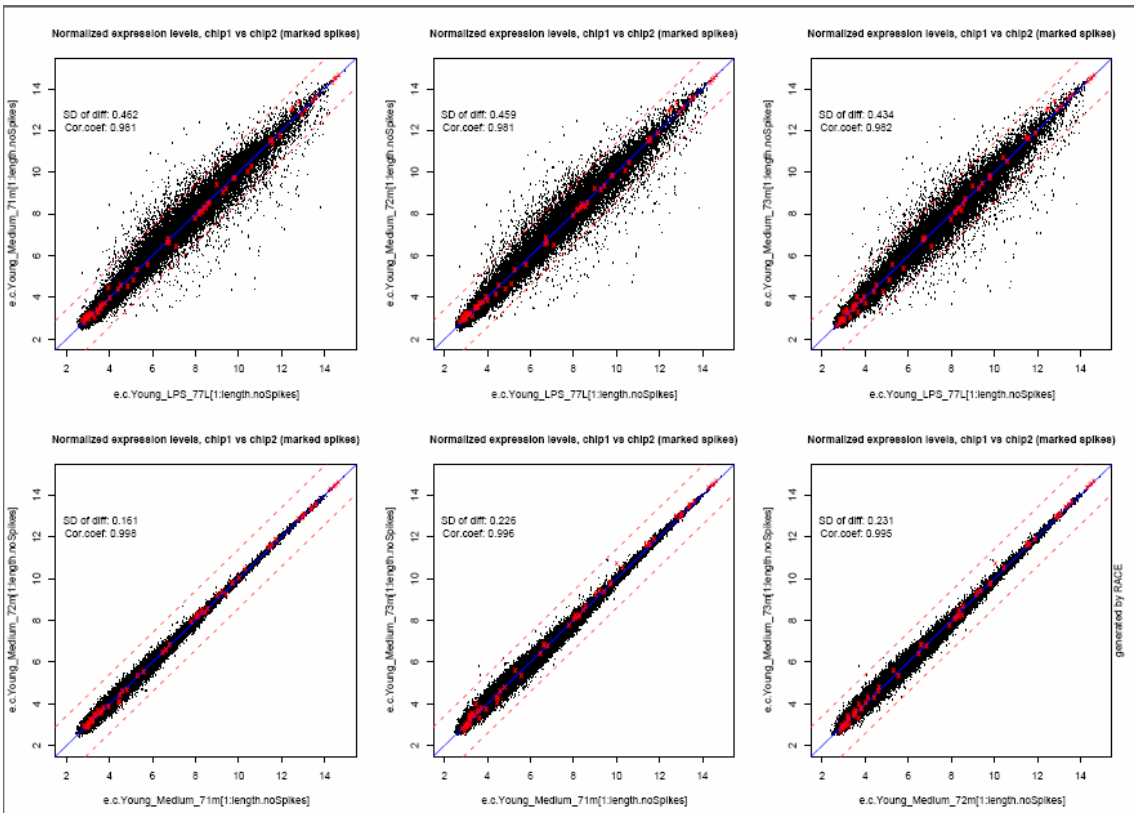


Figura 10. Gráficos de comparación de niveles de expresión normalizados chip a chip. Se generan 12x12 gráficos, aquí se presentan únicamente 6 de ellos.

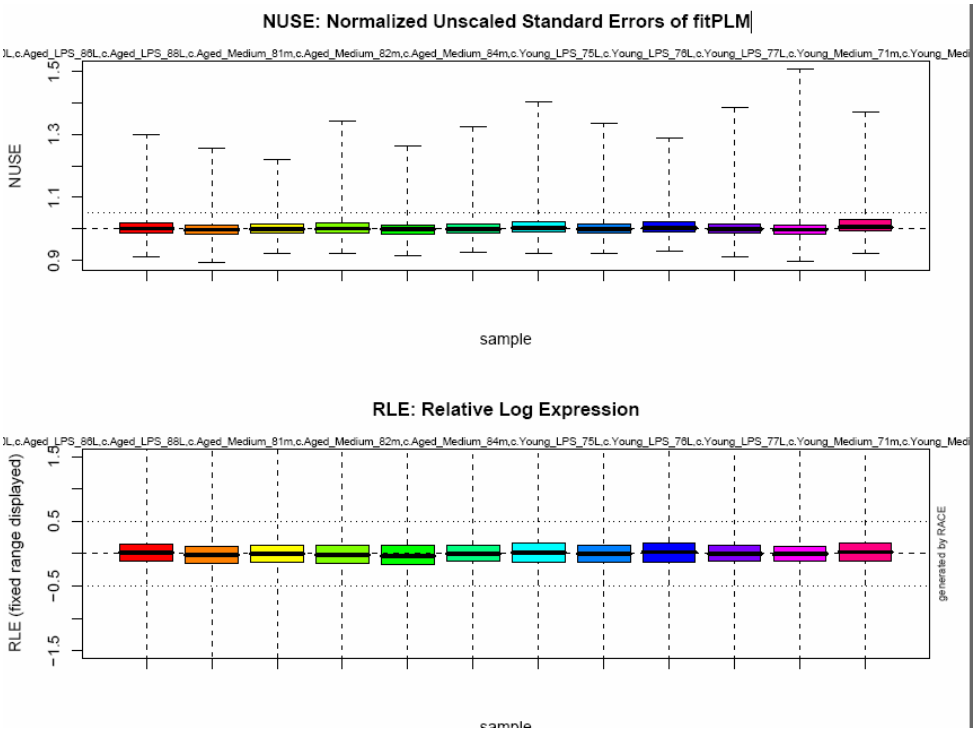


Figura 11. Boxplots de NUSE y RLE.

Código R para reproducir la normalización

Adjunto parte del codigo en R que genera RACE. Omito por simplicidad las partes que generan los gráficos.

```
#
  # This R-script was generated by RACE - a Remote Analysis Computation
  # for gene Expression data (http://race.unil.ch).
  # tamarit submitted this analysis request on Tue Apr 21 09:40:58 2009 to the DAFL-RACE
  # The corresponding data are stored for 7 days in /services/dafl-
files/tamarit/rma_norm_04210940

  # Attention: This script is adapted for R 2.01 running under linux

  # Rename the cel-files to the internal naming system name1a, name1b...
  # @*move_command

  # read in R-library files
  # following R-library files must be installed on the computer:

  library(affy)
  library(mouse4302cdf)
  library(Biobase, warn.conflicts=FALSE)
  library(affyPLM)
  library(fields)


  #if( "rma" == "gcrma")
  #  library(rma)


  setwd("/services/dafl-files/tamarit/rma_norm_04210940")   # sets work directory (linux)
  # getwd()          # gives current work directory

  #-------------------------------------------#
  #--- read in the raw data:


  sn <-
c("c.Aged LPS 80L","c.Aged LPS 86L","c.Aged LPS 88L","c.Aged Medium 81m","c.Aged Medium 82m","c.A
ged Medium 84m","c.Young LPS 75L","c.Young LPS 76L","c.Young LPS 77L","c.Young Medium 71m","c.You
ng_Medium_72m","c.Young_Medium_73m")
  raw <- ReadAffy(sampleNames=sn)
  raw
  pData(raw)
……..
#====== Do now the normalization ============#

norm <- rma(raw)
pData(norm)

expr <- exprs(norm)

colnames(expr)=sn
expr2 <- expr
rowNames <- row.names(expr2)
identifier <- as.vector(rowNames)
expr2<- cbind(identifier,expr2)

write.table(expr2, file = "NormExprLevels.txt", sep="\t", row.names=FALSE)
```

```
maxi <-max(exprs(norm))

spikes <- grep("AFFX.+", featureNames(norm))
spikes.names <- featureNames(norm[spikes])
length(spikes.names)
length.noSpikes <- dim(expr)[1]-length(spikes.names)

exprNoSpikes=expr[1:length.noSpikes,];
exprSpikes=expr[(length.noSpikes+1):dim(expr2)[1],];

variable=apply(exprNoSpikes,1,sd)
ordered.genes=order(variable,decreasing=TRUE);

pearson <- as.data.frame(signif(cor(exprNoSpikes), digits = 3),row.names=sn)
write.table(pearson, file = "Pearson.correlation.txt", sep="\t", col.names = NA)
pearson <- cbind(sn,pearson)

#-----------------------------------------#
#--- calculate and save the expression levels:

e.c.Aged_LPS_80L <- exprs(norm)[,1]
e.c.Aged_LPS_86L <- exprs(norm)[,2]
e.c.Aged_LPS_88L <- exprs(norm)[,3]
e.c.Aged_Medium_81m <- exprs(norm)[,4]
e.c.Aged_Medium_82m <- exprs(norm)[,5]
e.c.Aged_Medium_84m <- exprs(norm)[,6]
e.c.Young_LPS_75L <- exprs(norm)[,7]
e.c.Young_LPS_76L <- exprs(norm)[,8]
e.c.Young_LPS_77L <- exprs(norm)[,9]
e.c.Young_Medium_71m <- exprs(norm)[,10]
e.c.Young_Medium_72m <- exprs(norm)[,11]
e.c.Young_Medium_73m <- exprs(norm)[,12]
#-----------------------------------------#

pairs.results <- as.data.frame(identifier)
```

Selección de genes diferencialmente expresados mediante RACE. Experimento con un factor

Entrada de los datos

La interface RACE, permite en unos pocos pasos realizar un análisis sencillo de los datos con un factor. En este caso vamos a analizar los datos con un diseño factorial como el de la tabla 1, para comprobar el efecto del LPS con independencia de la edad.

Lo primero es seleccionar el tipo de chip y en fichero de input (figura 12) , que será el fichero de expresión obtenido en el paso de normalización.

- Se selecciona la opción “split existing file”,
- Seleccionamos el fichero “NormExpeLevels.txt”
- Seleccionamos las marcas de verificación en los chips con tratamiento con LPS para la compación A y los chips con medio para el tratamiento B.

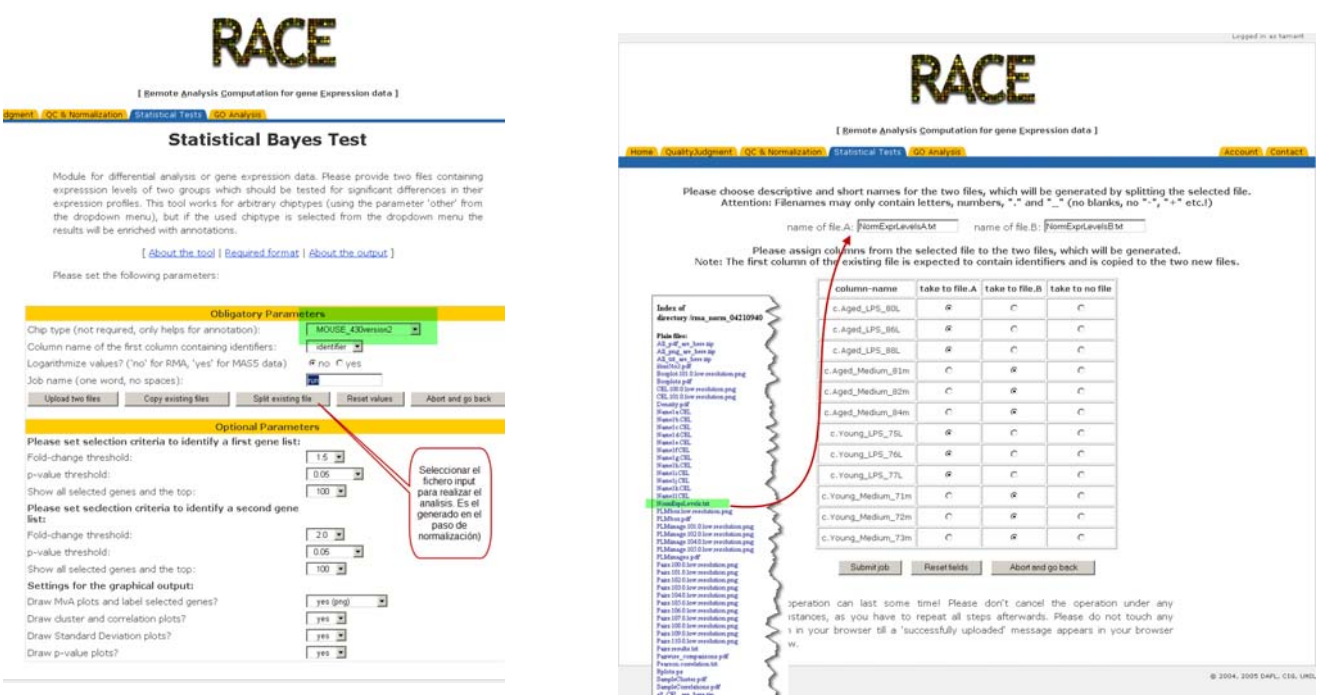


Figura 12. Paginas para realizar el test estadístico mediante RACE

Output del análisis de un factor

En pocos minutos se obtiene un análisis completo en la carpeta del proyecto (Figura 13).

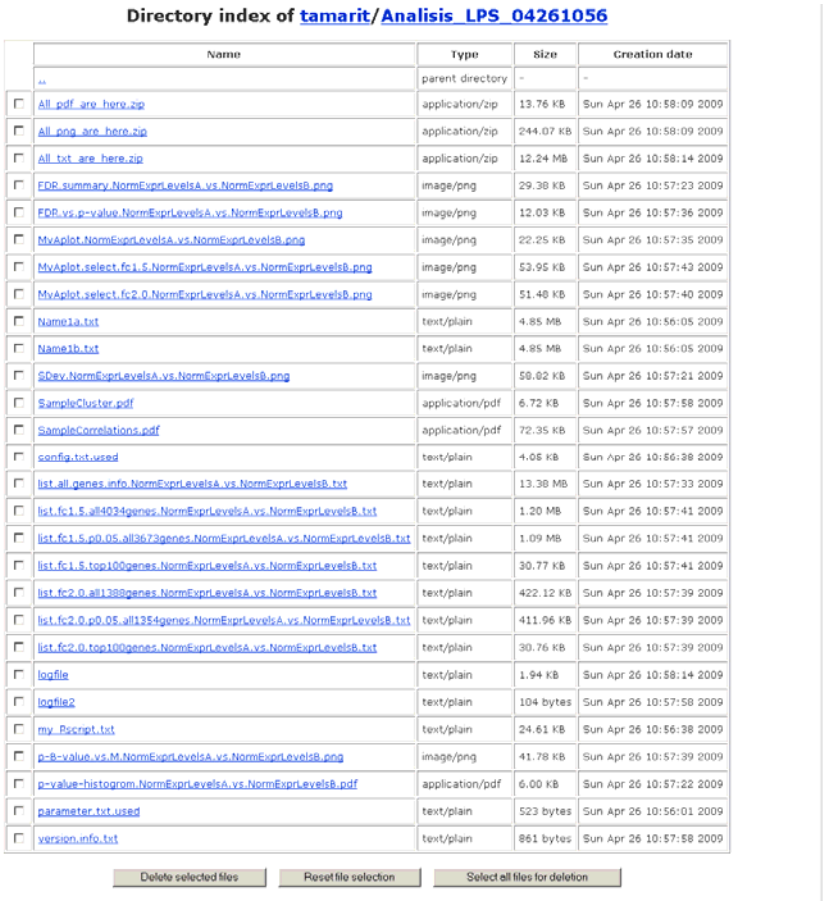


Figura 13. output del proceso de analisis de un factor.

De la amplia información que proporciona el output vamos analizar la que creo que es más interesante.

Análisis de cluster.

El análisis de cluster nos permite observar como están interrelacionados los chips. Aunque las relaciones entre ellos es algo que a priori ya sabemos, por ejemplo supongamos que no conocíamos la posibilidad de que existiera una dependencia entre la edad. El siguiente grafico nos muestra este hecho con claridad.

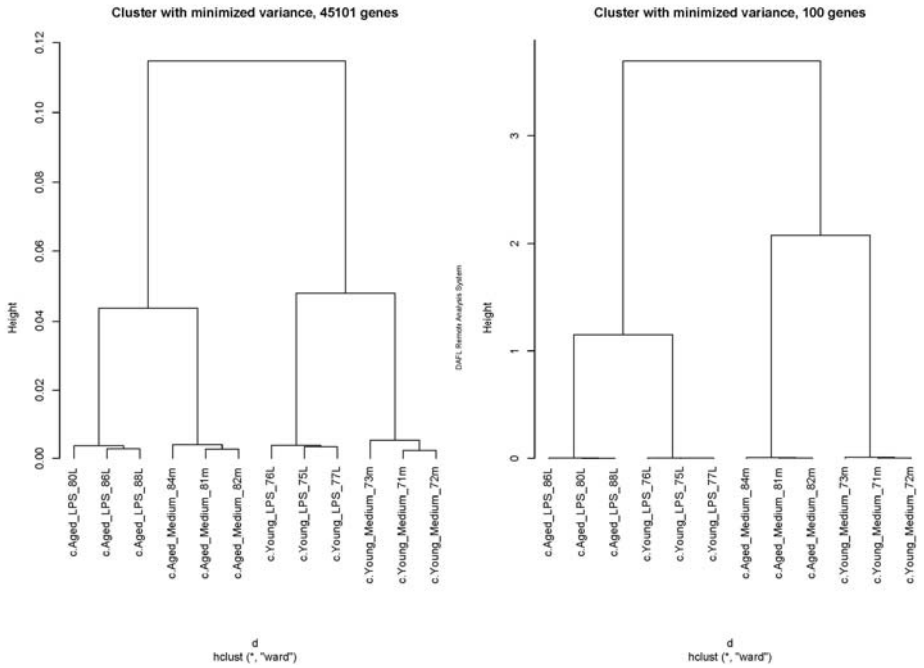


Figura 14. Análisis de cluster de los chips.

Análisis de la distribución de los p-values corregidos por ebayes..

La distribución de los p-valor (Figura 15 ya nos informa que:

a. Tenemos genes con expresión diferencial,
b. El criterio de corte 0.05 es acertado,
c. No parece que existan artificios en los resultados, la trasformación logarítmica de los p-valor nos daría una gausiana perfecta.

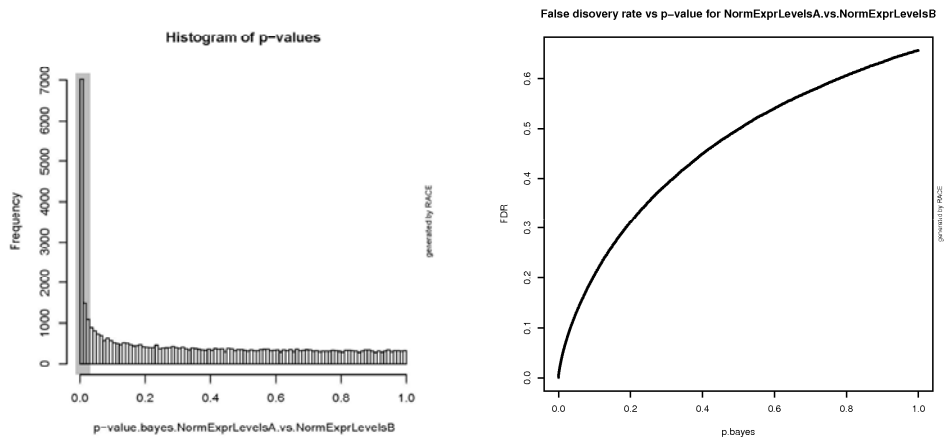


Figura 15. Distribución de los p-valor y tasa de FDRvsP-palor

Análisis del FDR.

El grafico es un resumen de la estimación del FDR. En un primer paso se calcula la fracción de genes que no muestran cambio de expresión, que ser en este caso es aproximadamente 67 %. El siguiente grafico es zoom del p-valor vs FDR, vemos que usando un p-valor de 0.03 podemos esperar un 10% de falsos positivos, dado que hemos usado un 0.05, vamos a tener gran cantidad de falsos positivos. Los siguientes graficos nos muestran el numero de falsos positivos en función de los puntos de corte (genes seleccionados como diferencialmente expresados).

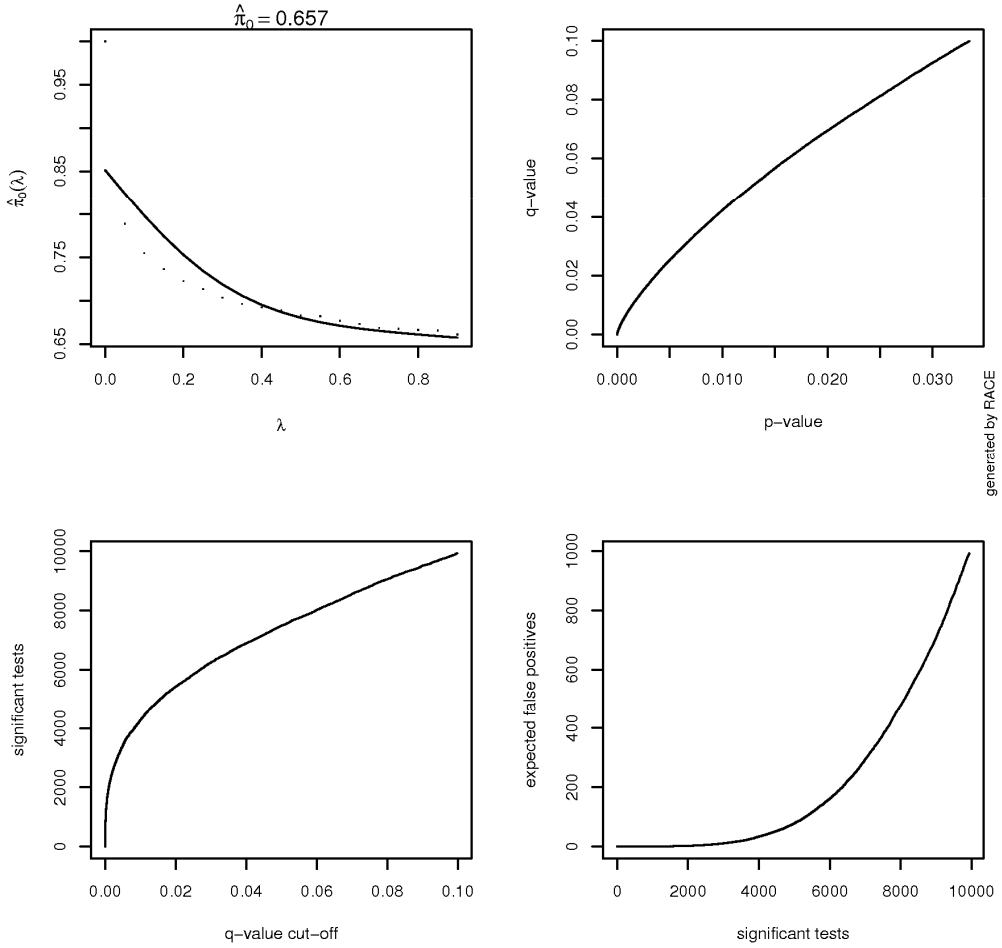


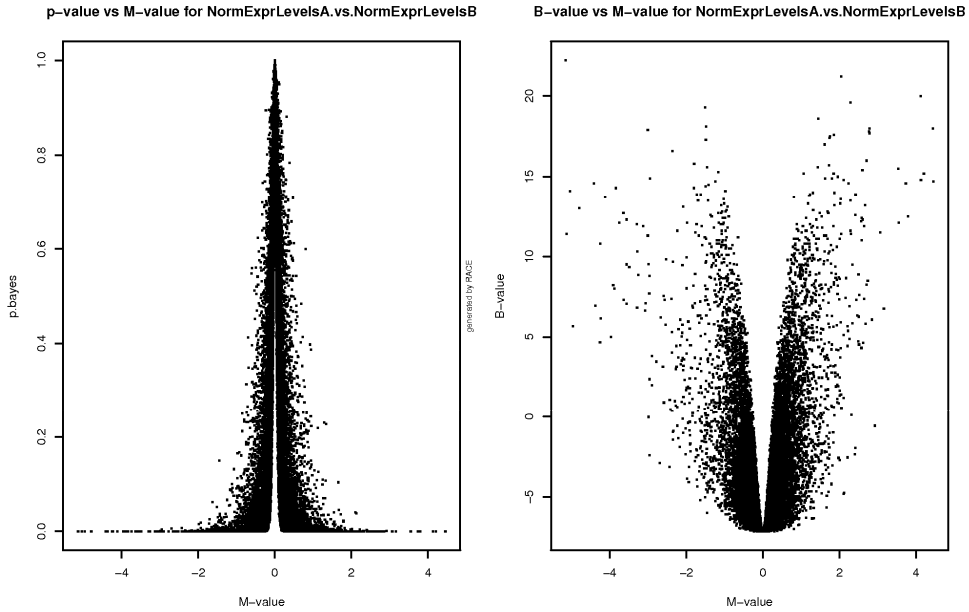
Figura 16. Resumen del análisis FDR

Graficos volcano, MAplot y la lista top 100

Los graficos volcano (figura 17) nos permiten en un primer momento fijar los puntos de corte adecuados, y ver donde se sitúan nuestros genes diferencialmente expresados. El grafico MAplot (uno de ellos y a que se obtienen dos juegos con los distintos parámetros introducidos en el input)) que se obtiene con RACE es particularmente informatico.

El grafico MAplot (figura 17) observamos con mucha claridad cuales son los genes diferencialmente expresados. Los genes de la lista (top-100) en base a fc (fold. Change) y p.bayes están marcado con un circulo rojo, los que están en la parte superior son sobre-expresados y en la inferior los sub-expresados.

Entre los listados que se obtienen destaca el que nos facilita la lista top-100, incluyendo todos los valores interesantes(figura 18).



(¿?¿?¿Es normal que en el volcano los genes perdidos a la izquierda esten maás bajos que los perdidos a la derecha¿?¿?)

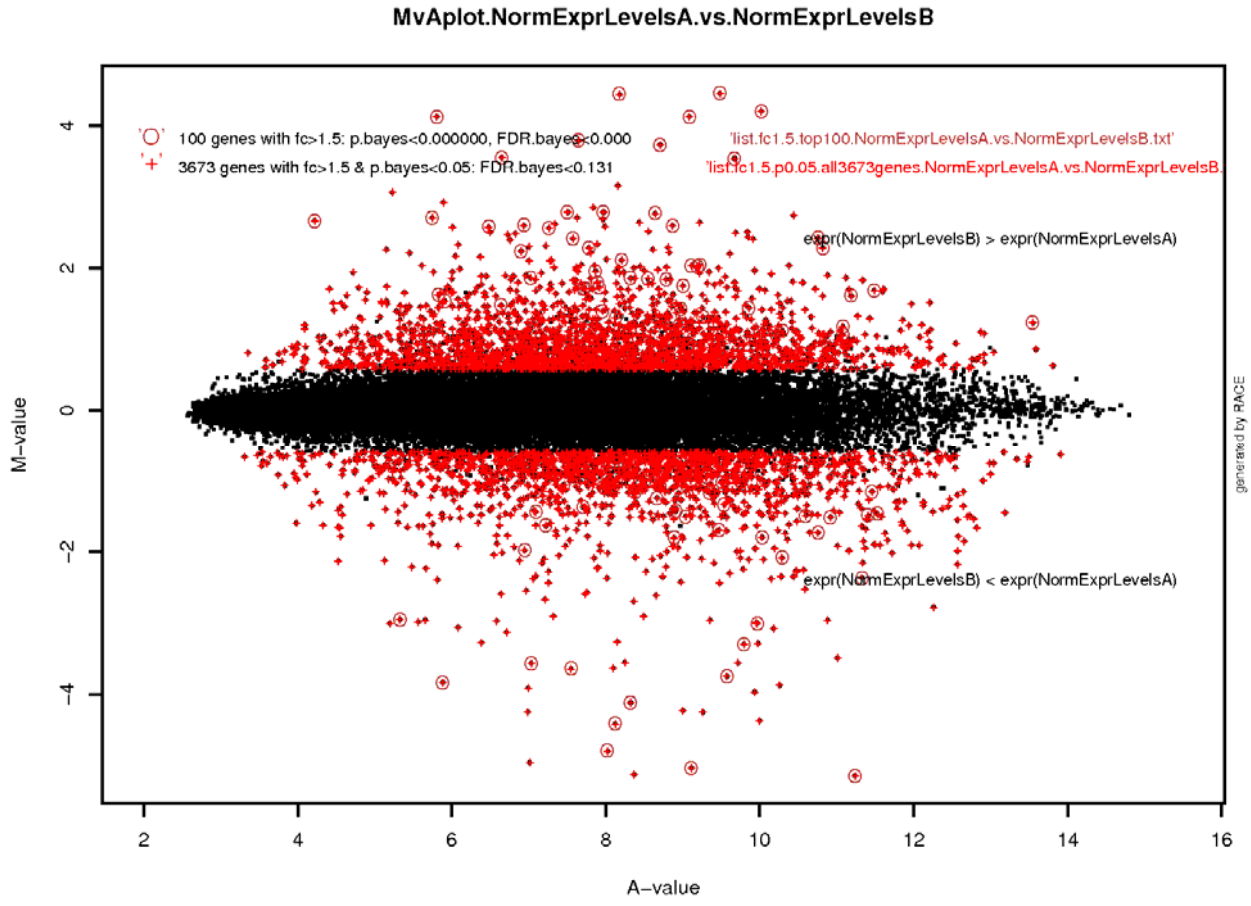


Figura 17. Graficos Volcano y MAplot (100 genes – fc>1.5 p.bayes<0.05)

(ESTA ES LA FIGURA CON “MES TRELLAT” DE TODA LA PEC)

identifier	symbols	locuslink	Avalue	M	foldchange	p.bayes	FDR.bayes	t.bayes	B.bayes	SDev.NormExprLevelsA	SDev.NormExprLevelsB
1450826_a_at	Saa3	20210	11.24	-5.148	0.0282	8.164192e-15	2.42e-10	-39.9	22.2	0.071965833232569	0.327354290829138 13.8451596277599
1449383_at	Adss1l	11565	9.222	2.045	4.13	4.024065e-14	5.96e-10	35.2	21.2	0.0759959500876805	0.0695907712007677 8.2018609861857
1418645_at	Hal	15109	9.086	4.127	17.5	2.303903e-13	2.28e-09	30.7	20	0.170146466339056	0.307752398304915 6.78265477392289
1448303_at	Gpnmb	93695	10.82	2.284	4.87	3.760771e-13	2.79e-09	29.5	19.6	0.164467333667662	0.0604722982109097 9.5289485003321
1430127_a_at	Ccnd2	12444	10.92	-1.509	0.351	5.773835e-13	3.42e-09	-28.5	19.3	0.0422466831874141	0.066504520299386 11.7031302831496
1451421_a_at	Rogdi	66049	9.862	1.445	2.72	1.458601e-12	7.21e-09	26.5	18.6	0.0611716604038333	0.0617651472261664 9.17734603540016
1416122_at	Ccnd2	12444	11.41	-1.483	0.358	2.542398e-12	9.78e-09	-25.4	18.1	0.0572711218659839	0.0877361289511678 12.2093911307486
1449193_at	Cd51	11801	7.506	2.787	6.9	2.835786e-12	9.78e-09	25.1	18	0.186603084163248	0.210466327618904 6.2261258859998
1415904_at	Lpl	16956	8.18	4.445	21.8	2.970013e-12	9.78e-09	25	18	0.157656293048320	0.449110558124682 5.96505410596322
1448728_a_at	Nfkbiz	80859	9.968	-3.003	0.125	3.299647e-12	9.78e-09	-24.8	17.9	0.281195790362496	0.133917648612019 11.1872903479985
1416389_a_at	Rcbtb2	105670	8.645	2.771	6.83	3.839036e-12	1.03e-08	24.5	17.8	0.143606759985697	0.249041297904981 7.31448336498114
1417133_at	Pmp22	18858	7.972	2.787	6.9	4.569276e-12	1.12e-08	24.2	17.7	0.181472633115249	0.231551735696489 6.64819771237163
1417063_at	C1qb	12260	8.552	1.854	3.62	4.929513e-12	1.12e-08	24.1	17.6	0.073359811607345	0.157923869141702 7.55877046251043
1417381_at	C1qa	12259	9.005	1.758	3.38	5.575924e-12	1.18e-08	23.8	17.5	0.0921821171231051	0.134406475944646 8.21243099302385
1423571_at	Edg1	13609	7.916	1.736	3.33	6.383603e-12	1.25e-08	23.6	17.4	0.153149050069497	0.0542709817377736 6.84509096928105
1455956_x_at	Ccnd2	12444	10.6	-1.488	0.357	6.762041e-12	1.25e-08	-23.5	17.3	0.042650859900362	0.117528754599764 11.3471133495379
1426258_at	Sorl1	20660	11.19	1.615	3.06	1.006954e-11	1.76e-08	22.7	17	0.112588826137069	0.104160026931128 10.3690005958791
1417483_at	Nfkbiz	80859	11.33	-2.361	0.195	1.522627e-11	2.51e-08	-22	16.6	0.265177918777883	0.0547042165861441 12.2497267668026
1416390_at	Rcbtb2	105670	5.746	2.711	6.55	3.077084e-11	4.8e-08	20.8	16	0.22256182749768	0.256855794811924 4.22544011236279
1449453_at	Bst1	12182	10.03	-1.794	0.288	4.097767e-11	6.07e-08	-20.3	15.8	0.0879393300188604	0.192267323952611 10.8320690657742
1434366_x_at	C1qb	12260	7.73	1.724	3.3	4.336173e-11	6.12e-08	20.2	15.7	0.146106743602117	0.138850181824982 6.67932610992812
1435745_at	5031439G07Rik	223739	8.978	1.431	2.7	4.826954e-11	6.37e-08	20	15.6	0.113847136692432	0.104477204184411 8.13352
1434745_at	Ccnd2	12444	11.53	-1.46	0.363	4.943761e-11	6.37e-08	-20	15.6	0.128391482658526	0.0956001588278722 12.0987128351206
1456046_at	Cd93	17064	9.675	3.539	11.6	5.655495e-11	6.98e-08	19.8	15.5	0.342068598628042	0.336788855961615 8.19259256557734
1417378_at	Cadm1	54725	8.867	2.6	6.06	6.273842e-11	7.44e-08	19.6	15.4	0.222888957446012	0.264326871114024 7.55034639212715
1428389_s_at	Wdr43	72515	9.353	-1.166	0.446	7.199463e-11	7.87e-08	-19.4	15.3	0.0557939315073291	0.096659319713972 9.9918227821698
1419821_s_at	Idh1	15926	8.785	1.848	3.6	7.318017e-11	7.87e-08	19.4	15.2	0.172191015890433	0.157897871687780 7.90723289958009
1450883_a_at	Cd36	12491	10.02	4.205	18.4	7.434436e-11	7.87e-08	19.4	15.2	0.450801749264420	0.378476759289835 7.33575857205601

Figura 18. Imagen del fichero" list.fc1.5.top100genes.NormExprLevelsA.vs.NormExprLevelsB.txt"

Código R para reproducir el análisis.

Adjunto parte el codigo R para reproducir el análisis. En amarillo he marcado las partes interesantes del análisis, y he borrado las intrascendentes (las que generan los gráficos e informas) (líneas en rojo).

```
#
  # This R-script was generated by RACE - a Remote Analysis Computation
  # for gene Expression data (http://race.unil.ch).
  # tamarit submitted this analysis request on Sun Apr 26 10:56:01 2009 to the DAFL-RACE
  # The corresponding data are stored for 7 days in /services/dafl-
files/tamarit/Analisis_LPS_04261056

  # Attention: This script is adapted for R 2.01 running under linux
  # To run this script on your local computer all paths mus be changed from
  # /services/dafl-files/tamarit/Analisis_LPS_04261056 to your working directory
  # Rename the input files to the internal naming system name1a, name1b...
  # @*move_command

  # read in R-library files
  # following R-library files must be installed on the computer:

  library(affy)
  library(Biobase, warn.conflicts=FALSE)
  library(fields)
  library(limma)
  library(annotate)
  library(qvalue)

  if( !("mouse4302" == "notKnown"))
    library(mouse4302cdf)

  setwd("/services/dafl-files/tamarit/Analisis_LPS_04261056")   # sets work directory (linux)
  # getwd()          # gives current work directory

  vec <- 1:2       # shall contain #(group1), #(group2)
  vec <- as.vector(vec)
```

```
  groups <- 1:2       # shall contain name(group1), name(group2)
  groups <- as.vector(groups)


  #-------------------------------------------#
  #--- read in the raw data:
  in.files <- list.files(pattern="Name")
  z=1


  in.file <- in.files[z]
  c.NormExprLevelsA <- read.table(in.file, header=TRUE, na.strings = c("NA","
","na","nan","NAN"),blank.lines.skip=TRUE,fill=TRUE)
  if( names(c.NormExprLevelsA[1])!="key" & names(c.NormExprLevelsA[1])!="AffyID" &
names(c.NormExprLevelsA[1])!="CloneID" & names(c.NormExprLevelsA[1])!="identifier")
   {
       sink("/services/dafl-files/tamarit/Analisis_LPS_04261056/logfile2")
       print("The first comlumn of c.NormExprLevelsA has non of the names
,'key','AffyID','CloneID', 'identifier' therfore R was stopped")
       print("Please make also sure that c.NormExprLevelsA is saved as tab-delimited text file
(excel-formats are not recognized). ")
        sink()
       q(save = "yes")
   }


    id.vector <- c.NormExprLevelsA$identifier
    id.unique <- unique(id.vector)
.............


  factor <- c(rep(0,vec[1]),rep(1,vec[2]))
  design=model.matrix(~1+factor)


  fit <- lmFit(data,design)
  ebfit <- eBayes(fit)


  sort.table <-
topTable(ebfit,coef=2,genelist=rownames(data),adjust="none",number=dim(data)[1])


  if (ncol(sort.table) == 6) { sort.table=sort.table[,c(1:4,6)]}


  colnames(sort.table) <- c("identifier","M","t.bayes","p.bayes","B")


  Typ <-"mouse4302"



  if(Typ == "ath1121501")
  {
       annot <- read.csv("/var/www/templates/AnnotationFiles/ATH1-RACE annot.csv", header=TRUE,
na.strings = c("NA","---"," "),fill=TRUE, sep=",")
       annot.text<-print("The annotation for ath1121501 comes from the Affymetrix annotation
file last update Sept 2005")
            data.out <- merge(annot, data.add,  by.x="AffyID", by.y="identifier")
  }


  #-------------------------------------------#
  #------visualiz correlation between samples :
  if("yes" == "yes")
  {
  variable <- apply(data,1,sd)
  ordered.genes <- order(variable,decreasing=TRUE)
  z=100
```

Selección de genes diferencialmente expresados con BioConductor. Experimento con dos factores

Consideraciones previas.

Los pasos relativos a carga, normalización y filtrado son identicos al caso anterior, lo único que cambiamos es el diseño factorial del experimento que será el de la tabla 1.

Dado que en este ejemplo vamos a construir nosotros mismos el código en R, la parte relacionada con la carga, control de calidad y normalización no los comentaré.

Carga, normalización filtrado..

El codigo R que nos permite llevar a cabo estos pasos es el siguiente:

```
#####################################
### PARAMETROS DEL ANALISIS
#####################################

### 1- DIRECTORIO DE DATOS Y RESULTADOS
workingDir <- "C:\\Users\\ramon\\Documents\\LPSData"
resultsDir <- "C:\\Users\\ramon\\Documents\\LPSData\\results_two"
setwd(workingDir)


### FICHERO DE "TARGETS"
targetsFile <- "LSPtargets.txt"

### FICHERO DE ANOTACIONES
anotPackage <- "mouse4302.db"

### METODO DE NORMALIZACION
normMethod <- "RMA"
## "MAS5", "VSN", "RMA", "GCRMA"

### FILTRADO POR VARIABILIDAD
filterByVar <- TRUE
variabilityThreshold <- 66

#####################################
### CARGA DE LOS DATOS
#####################################

###setwd(dataDir)
setwd(dataDir)
require(affy)
require("mouse4302.db")
rawData <-ReadAffy()
sampleInfo <-read.AnnotatedDataFrame ("LSPtargets.txt")
phenoData(rawData) <- sampleInfo
experimentInfo <-new("MIAME", name="",
  title="EXPERIMENTO LPS",
experimentData (rawData) <-experimentInfo
annotation(rawData) <- "mouse4302.db"

#########################
### NORMALIZACIÓN
#########################
stopifnot(require(affy))
```

```
normalitza <-function (dades, metode){
  switch(metode,
    RMA =  rma(rawData),    # Creates expression values using RMA method.
    GCRMA = gcrma(rawData),
    MAS5 = mas5 (rawData) )
}
eset_norm <- normalitza (rawData, normMethod)

#########################
### FILTRADO
#########################
 eset_norm_filtered <-eset_norm

 if (filterByVar){
   sds<- apply(exprs(eset_norm), 1, sd)
   variability_threshold <- variabilityThreshold/100
   f2 <-function (x)
    if (sd(x)< variability_threshold) return(FALSE) else return(TRUE)
   ffun2<-filterfun(f2)
   which2 <- genefilter(exprs(eset_norm), ffun2)
   sum (which2)
   eset_norm_filtered <-eset_norm[which2,]
   }
```

Como resultado del script obtenemos un objeto “eset_norm_filtered” de la clase exprSet , que puede ser utilizado en las posteriores etapas del análisis.

Los parámetros del análisis son:

```
### AJUSTE DE "MULTIPLE TESTING"
adjustmethod <- "fdr"
# "none", "Bonferroni", "FDR"

### AJUSTE DE "MULTIPLE COMPARISONS"
multCompMethod <- "separate"
# "separate", "Global", "Nested"

### P-VALOR PARA DISCRIMINAR GENES

pvalueType <- "unadjusted"
# ("unadjusted", "adjusted")

### LIMITE p-VALOR SIN AJUSTAR
unadjusted.pValue.cutoff <- 0.05

### LIMITE p-VALOR AJUSTADO
adjusted.pValue.cutoff <- 0.25

p.value.cutoff <- ifelse (pvalueType=="unadjusted",unadjusted.pValue.cutoff,
adjusted.pValue.cutoff)
```

Definir la matriz de diseño del modelo

Construir una matriz con los nombres que asignaremos a los tratamientos y asignar cada nombre a su correspondiente columna de la matriz de diseño. Es decir

Para los dos factores y teniendo en cuenta el orden de nuestros ficheros (ojo¡¡) el código R queda:

```
###MATRIZ DE DISEÑO
design <-as.matrix (data.frame(Aged.LPS =  c(1,1,1,0,0,0,0,0,0,0,0,0),
                               Aged.MED =  c(0,0,0,1,1,1,0,0,0,0,0,0),
                               Young.LPS = c(0,0,0,0,0,0,1,1,1,0,0,0),
                               Young.MED = c(0,0,0,0,0,0,0,0,0,1,1,1)))
```

igualmente podemos hacer:

```
TS <- factor(TS, levels=c("Aged.LPS" ,"Aged.MED","Young.LPS","Young.MED"))
design <- model.matrix(~0+TS)
colnames(design) <- levels(TS)
```

y R nos construye la matriz en base a los factores. (más sencillo¡¡¡)

```
> TS
 [1] Aged.LPS  Aged.LPS  Aged.LPS  Aged.MED  Aged.MED  Aged.MED  Young.LPS
 [8] Young.LPS Young.LPS Young.MED Young.MED Young.MED
Levels: Aged.LPS Aged.MED Young.LPS Young.MED

> design
   Aged.LPS Aged.MED Young.LPS Young.MED
1         1        0         0         0
2         1        0         0         0
3         1        0         0         0
4         0        1         0         0
5         0        1         0         0
6         0        1         0         0
7         0        0         1         0
8         0        0         1         0
9         0        0         1         0
10        0        0         0         1
11        0        0         0         1
12        0        0         0         1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$TS
[1] "contr.treatment"
```

Ajustar los datos al modelo

Mediante el siguiente código obtenemos un objeto de la clase MArrayLM. Los coeficientes ajustados (fit$coef) del modelo lineal son la media logarítmica del valor de la expresión para cada conjunto de probes.

```
my.eset <- eset_norm_filtered
require(limma)

####################################################
### 5.1: linearmodelfit
####################################################
fit<-lmFit(my.eset, design)

> fit
An object of class "MArrayLM"
$coefficients
           Aged.LPS Aged.MED Young.LPS Young.MED
1415694_at 8.690726 7.967451  9.785454  8.545976
```

```
1415800_at 7.346198 3.740525  8.257183  4.192073
1415801_at 6.328098 4.730419  6.558419  5.204197
1415802_at 9.933655 8.367766 10.183866  8.762050
1415810_at 9.415777 9.063652  7.463254  7.219140
2544 more rows ...

$rank
[1] 4

$assign
[1] 1 1 1 1

$qr
$qr
    Aged.LPS   Aged.MED Young.LPS Young.MED
1 -1.7320508  0.0000000  0.000000  0.000000
2  0.5773503 -1.7320508  0.000000  0.000000
3  0.5773503  0.0000000 -1.732051  0.000000
4  0.0000000  0.5773503  0.000000 -1.732051
5  0.0000000  0.5773503  0.000000  0.000000
7 more rows ...

$qraux
[1] 1.577350 1.000000 1.000000 1.000000

$pivot
[1] 1 2 3 4

$tol
[1] 1e-07

$rank
[1] 4


$df.residual
[1] 8 8 8 8 8
2544 more elements ...

$sigma
1415694_at 1415800_at 1415801_at 1415802_at 1415810_at
0.08240141 0.13161273 0.18872598 0.15760300 0.10321168
2544 more elements ...

$cov.coefficients
           Aged.LPS  Aged.MED Young.LPS Young.MED
Aged.LPS  0.3333333 0.0000000 0.0000000 0.0000000
Aged.MED  0.0000000 0.3333333 0.0000000 0.0000000
Young.LPS 0.0000000 0.0000000 0.3333333 0.0000000
Young.MED 0.0000000 0.0000000 0.0000000 0.3333333

$stdev.unscaled
            Aged.LPS  Aged.MED Young.LPS Young.MED
1415694_at 0.5773503 0.5773503 0.5773503 0.5773503
1415800_at 0.5773503 0.5773503 0.5773503 0.5773503
1415801_at 0.5773503 0.5773503 0.5773503 0.5773503
1415802_at 0.5773503 0.5773503 0.5773503 0.5773503
1415810_at 0.5773503 0.5773503 0.5773503 0.5773503
2544 more rows ...
```

```
$pivot
[1] 1 2 3 4

$genes
[1] "1415694_at" "1415800_at" "1415801_at" "1415802_at" "1415810_at"
2544 more rows ...

$Amean
1415694_at 1415800_at 1415801_at 1415802_at 1415810_at
  8.747402   5.883995   5.705283   9.311834   8.290456
2544 more elements ...

$method
[1] "ls"

$design
  Aged.LPS Aged.MED Young.LPS Young.MED
1        1        0         0         0
2        1        0         0         0
3        1        0         0         0
4        0        1         0         0
5        0        1         0         0
7 more rows ...
```

Definir la matriz de contrastes.

Es decir, decidir que comparaciones nos interesan de todas las posibles. Los contrastes son combinaciones lineales de los parámetros del modelo de ajuste. En este caso nos interesa comparar

- Efecto del LPS en adultos
- Efecto del LPS en jóvenes
- Efecto combinado de la edad y el tratamiento

```
### MATRIZ DE CONTRASTES
require(limma)
cont.matrix <- makeContrasts ( AgedLPSvsAgedMED = Aged.MED-AgedLPS,
                               YoungLPSvsYoungMED = Young.MED-Young.LPS,
                               Inter = (Young.MED-Young.LPS) - (Aged.MED-AgedLPS),
                               levels = design)
detach(package:limma)

> cont.matrix
           Contrasts
Levels      Aged.LPSvsAgedMED YoungLPSvsYoungMED Inter
  Aged.LPS                 -1                  0     1
  Aged.MED                  1                  0    -1
  Young.LPS                 0                 -1    -1
  Young.MED                 0                  1     1
>
```

Obtener el ajuste al modelo lineal para los contrastes.

Aquí ya obtenemos un conjunto de p-valor para cada comparación. Para incrementar la potencia y reducir el riesgo de los falsos positivos se ajustan también mediante el estadístico eBayes(). Los comandos son los siguientes:

```
fit2 <- contrasts.fit(fit, cont.matrix)
fit3 <- eBayes(fit2)
> fit3
```

```
An object of class "MArrayLM"
$coefficients
            Contrasts
             Aged.LPSvsAgedMED YoungLPSvsYoungMED      Inter
  1415694_at        -0.7232754         -1.2394775 -0.5162020
  1415800_at        -3.6056730         -4.0651096 -0.4594366
  1415801_at        -1.5976790         -1.3542224  0.2434566
  1415802_at        -1.5658886         -1.4218154  0.1440732
  1415810_at        -0.3521243         -0.2441138  0.1080105
2544 more rows ...

$t
            Contrasts
             Aged.LPSvsAgedMED YoungLPSvsYoungMED      Inter
  1415694_at         -8.860381         -15.184039 -4.4715016
  1415800_at        -34.177359         -38.532255 -3.0793763
  1415801_at        -11.624776          -9.853376  1.2525690
  1415802_at        -13.086461         -11.882411  0.8513919
  1415810_at         -3.864687          -2.679234  0.8382419
2544 more rows ...

$p.value
            Contrasts
             Aged.LPSvsAgedMED YoungLPSvsYoungMED        Inter
  1415694_at      9.208293e-07       1.841977e-09 0.0006816589
  1415800_at      8.750618e-14       1.962447e-14 0.0090930783
  1415801_at      4.293639e-08       2.832820e-07 0.2331548433
  1415802_at      1.076642e-08       3.329783e-08 0.4104711219
  1415810_at      2.069701e-03       1.938584e-02 0.4175426641
2544 more rows ...

$lods
            Contrasts
             Aged.LPSvsAgedMED YoungLPSvsYoungMED      Inter
  1415694_at          5.259401          11.830088 -0.9854241
  1415800_at         22.184783          23.673574 -3.6102592
  1415801_at          8.516334           6.480691 -6.6110323
  1415802_at          9.985350           8.756891 -7.0270288
  1415810_at         -2.838599          -5.116993 -7.0383341
2544 more rows ...
….
..
```

Corrección del p-valor para tener en cuenta el problema del "multiple testing".

Para corregir los p-valor para tener en cuenta el problema del "multiple testing", usamos decideTest() con el método FDR y un cut-off de 0.01

```
results<-decideTests(fit3 , method="global", adjust.method="fdr", p.value=0.01)
number_genes <- dim(exprs(eset_norm_filtered))[1]

> number_genes
[1] 2549
> results
TestResults matrix
            Contrasts
             Aged.LPSvsAgedMED YoungLPSvsYoungMED Inter
  1415694_at                -1                 -1    -1
  1415800_at                -1                 -1     0
  1415801_at                -1                 -1     0
```

```
  1415802_at                -1                -1      0
  1415810_at                -1                 0      0
2544 more rows ...
```

Análisis de los resultados y selección del punto de corte

Para analizar los datos y determinar si los puntos de corte son adecuados podemos graficar los valores corregidos con un histograma para seleccionar los genes diferenciados y comparar con los datos sin ajustar para comprobar que hemos elegido un punto de corte adecuado. Otra opción (no la he puesto en practica aquí, es buscer los probes de control y aplicar el punto de corte un poco más bajo que su p.valor)

```
FDR_cutoff <- 0.01
histogram_breaks <- 20
uniform_line <- 1/histogram_breaks
p_values <- fit3$p.value
adjusted_p_values <- test_results$adj.P.Val
hist(p_values, freq = F  , col="lightblue", breaks=histogram_breaks,
      main="Histogram of raw and adjusted p-values")
hist(adjusted_p_values, density=6, add=TRUE, freq = F, col="magenta", breaks=histogram_breaks)
abline(h=1, col="green", lwd=2)
abline(v=FDR_cutoff, col="red", lwd=2)
legend(x=0.5, y=12, legend=c("raw p-values","adjusted p-values"),
fill=c("lightblue","magenta"))
```

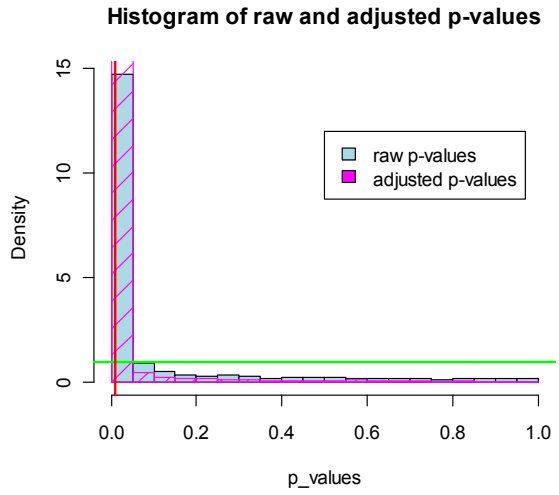


Figura 19. Histograma de p-valor corregido

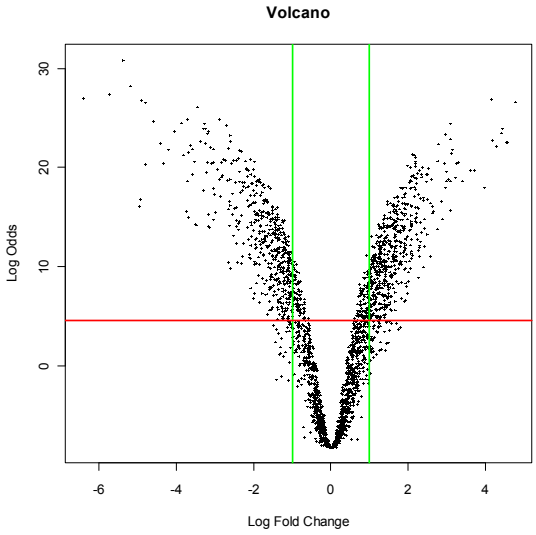


Figura 20. Volcano de los valores corregidos (vaya los de la izquierda están altos¡¡)

Desde R podemos visualizar el Grafico de Vulcano y las distribuciones.

```
volcanoplot(fit3,coef=2,highlight=number_genes,names="" ,main="Volcano fit3")
abline(v=c(-1,1), col="green", lwd=2)
abline(h=-log(FDR_cutoff), col="red", lwd=2)
```

Como vimos en el ejemplo simple de un nivel, parece que la elección de 0.01 como p-valor de corte es acertado.

Extracción de los genes diferencialmente expresados.

Como ultimo paso sacaremos la lista de genes diferencialmente expresados por cada factor

Primero los visualizamos, y podemos obtener una lista completa:

```
> significant_genes[1:10,]
               ID Aged.LPSvsAgedMED YoungLPSvsYoungMED      Inter   AveExpr        F      P.Value    adj.P.Val
2077 1450826_a_at         -4.911288          -5.384695 -0.4734070 11.235243 4752.207 7.893708e-19 2.012106e-15
2456 1457644_s_at         -4.285913          -3.459489  0.8264246 10.257469 2832.172 2.033343e-17 2.591495e-14
1984   1449450_at         -5.163635          -4.909680  0.2539550  9.113705 2652.598 3.066771e-17 2.605733e-14
2038   1450330_at         -4.518963          -5.736705 -1.2177420  8.368986 2245.901 8.709827e-17 5.550337e-14
22     1415904_at          4.132099           4.757366  0.6252671  8.180173 2137.762 1.186813e-16 6.050371e-14
2543   1460667_at         -4.019868          -4.803398 -0.7835303  8.123409 2035.343 1.614539e-16 6.100794e-14
849  1425569_a_at         -2.626651          -5.189053 -2.5624019  6.991260 2023.367 1.675385e-16 6.100794e-14
2394   1456046_at          2.934372           4.143450  1.2090779  9.675397 1912.017 2.388897e-16 6.911325e-14
430    1419607_at         -3.027880          -2.882472  0.1454085 10.883871 1905.540 2.440248e-16 6.911325e-14
166    1417262_at         -4.061533          -3.864056  0.1974764  9.933736 1798.585 3.504653e-16 8.933360e-14
>
```

Para obtener la lista por cada factor usamos la función toptable, previamente recuperamos las anotaciones de los genes:

```
anotPackage <- "mouse4302.db"
require(annotate)
require(anotPackage, character.only=T)
require(annaffy)
geneIDs <- ls(mouse4302cdf)
geneSymbols <- unlist(as.list(mouse4302SYMBOL))
geneNames <- unlist(as.list(mouse4302GENENAME))
geneNames <- substring(geneNames,1,40)
genelist <- data.frame(GeneID=geneIDs,GeneSymbol=geneSymbols,GeneName=geneNames)

topTab.1<-topTable(fit3, number=100, coef=1, adjust=adjustmethod, genelist=genelist)
topTab.2<-topTable(fit3, number=100, coef=2, adjust=adjustmethod, genelist=genelist)
topTab.3<-topTable(fit3, number=100, coef=3, adjust=adjustmethod, genelist=genelist)

if (pvalueType=="unadjusted"){
  topTabSelected.1 <-topTab.1[topTab.1$P.Value > p.value.cutoff,]
  topTabSelected.2 <-topTab.2[topTab.2$P.Value > p.value.cutoff,]
  topTabSelected.3 <-topTab.3[topTab.3$P.Value > p.value.cutoff,]
}else{
  topTabSelected.1 <-topTab.1[topTab.1$adj.P.Value > p.value.cutoff,]
  topTabSelected.2 <-topTab.2[topTab.2$adj.P.Value > p.value.cutoff,]
  topTabSelected.3 <-topTab.3[topTab.3$adj.P.Value > p.value.cutoff,]
}
```

Y los ficheros con las anotaciones nos quedan (por simplicidad únicamente muestro un trozo de uno de ellos):

```
> topTab.1[1:10,]
          GeneID GeneSymbol                                GeneName     logFC   AveExpr         t      P.Value    adj.P.Val        B
2077   1417746_at      Cplx1                              complexin 1 -4.911288 11.235243 -65.69717 2.470264e-17 6.296702e-14 29.81360
2456   1418125_at      Ino80           INO80 homolog (S. cerevisiae) -4.285913 10.257469 -58.56404 1.043677e-16 1.330167e-13 28.57285
1984   1417653_at      Pvalb                              parvalbumin -5.163635  9.113705 -52.78504 3.835717e-16 3.259081e-13 27.40595
439  1416108_a_at      Tmed3 transmembrane emp24 domain containing 3 -3.709388  7.730291 -44.78751 2.997378e-15 1.317205e-12 25.48735
430    1416099_at      Rpl27                   ribosomal protein L27 -3.027880 10.883871 -44.71291 3.060511e-15 1.317205e-12 25.46748
381  1416050_a_at     Scarb1     scavenger receptor class B, member 1 -5.037337  9.997523 -44.21269 3.522717e-15 1.317205e-12 25.33323
272  1415941_s_at    Zfand2a          zinc finger, AN1-type domain 2A -3.107721 12.571783 -43.74392 4.024797e-15 1.317205e-12 25.20573
166    1415835_at     Prl3b1 prolactin family 3, subfamily b, member  -4.061533  9.933736 -43.45294 4.374947e-15 1.317205e-12 25.12575
2382   1418051_at      Ephb6                          Eph receptor B6 -3.648427 11.007410 -43.16802 4.749847e-15 1.317205e-12 25.04682
22     1415691_at       Dlg1      discs, large homolog 1 (Drosophila)  4.132099  8.180173  42.87785 5.167535e-15 1.317205e-12 24.96578
>
```

Igualmente, el diagrama de venn mostrando los genes comunes de los tres contrastes (que no se parece en nada a lo que pone en el articulo original):

```
require(annotate)
require(anotPackage)
require("gplots")

fit.Symbols <- getSYMBOL (rownames(fit3), anotPackage)
result<-decideTests(fit3, method=multCompMethod, adjust.method=adjustmethod,
p.value=p.value.cutoff)
probeNames<-rownames(result)
rownames(result)<-fit.Symbols
sum.res.rows<-apply(abs(result),1,sum)
res.selected<-res[sum.res.rows!=0,]
probeNames.selected<-probeNames[sum.res.rows!=0]
vennDiagram (res.selected[,1:3], main="Genes in common ", cex=0.9)

exprs2cluster <-exprs(eset_norm_filtered)[probeNames.selected,]
colnames(exprs2cluster)<-rownames(pData(eset_norm_filtered))
heatmap.2(exprs2cluster,
          col=redgreen(75), scale="row",
          key=TRUE, symkey=FALSE,
          density.info="none", trace="none", cexCol=0.75)
```

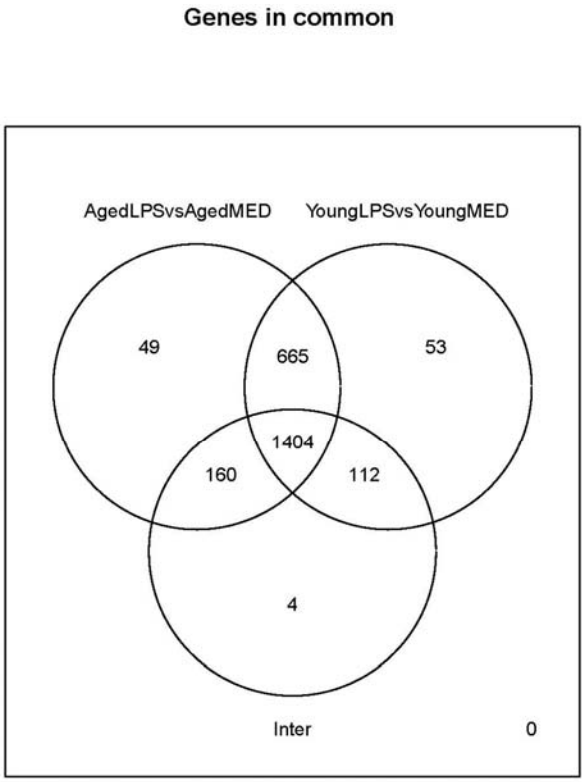
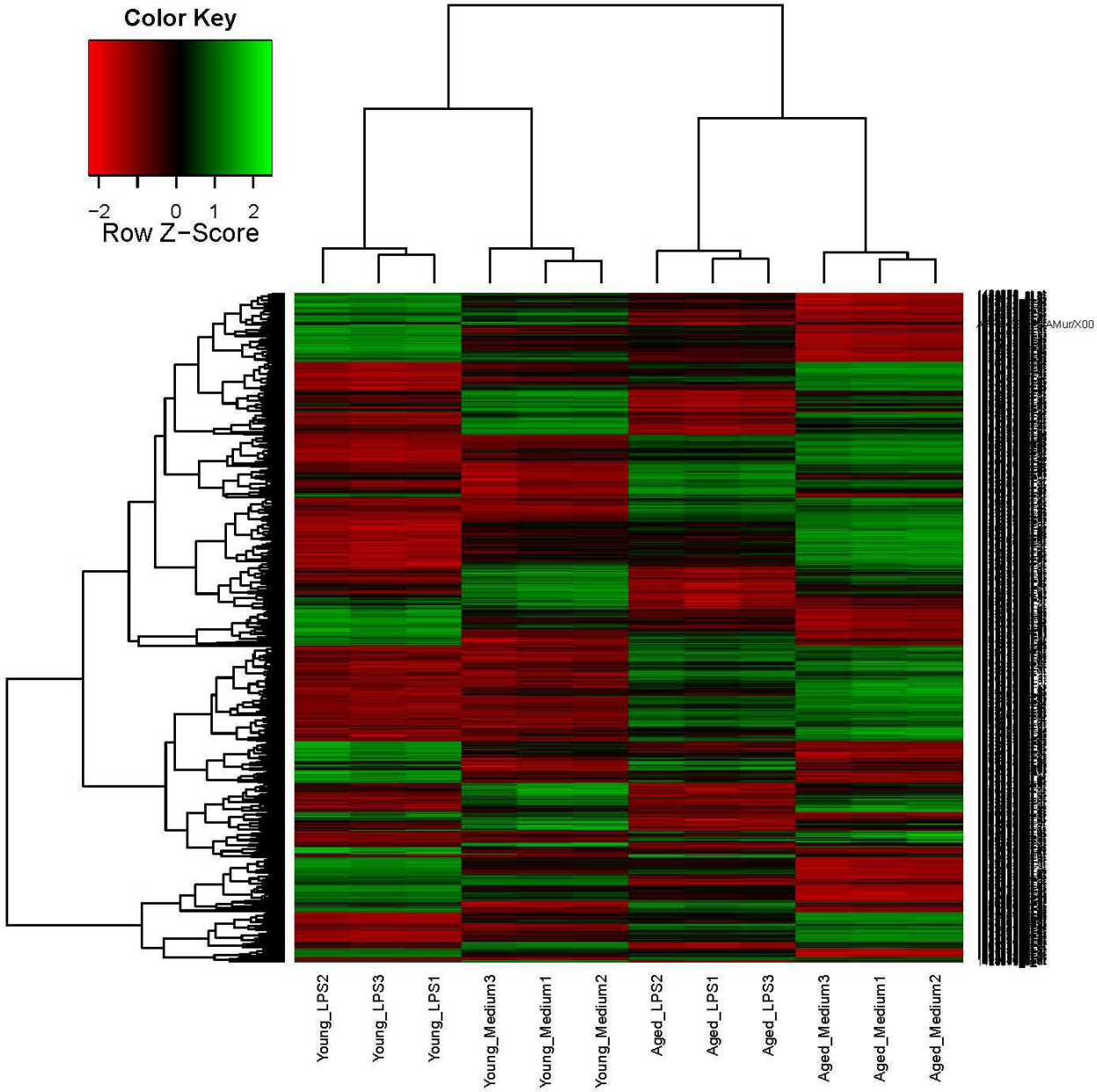
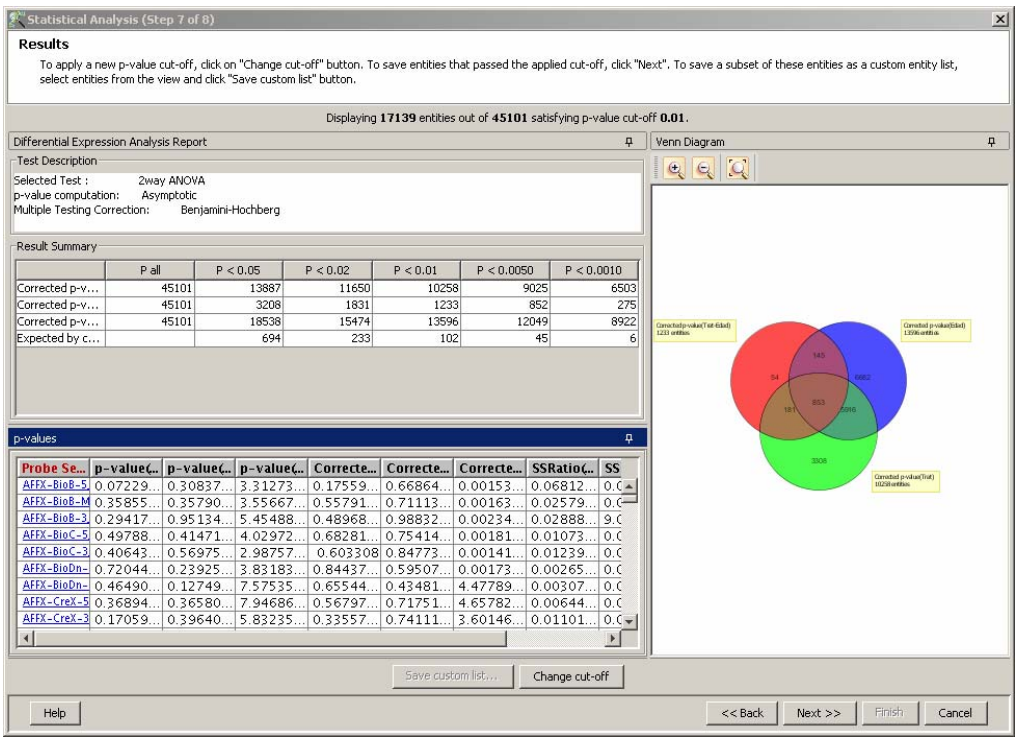


Figura 21. Diagrama de venn de los genes en común.

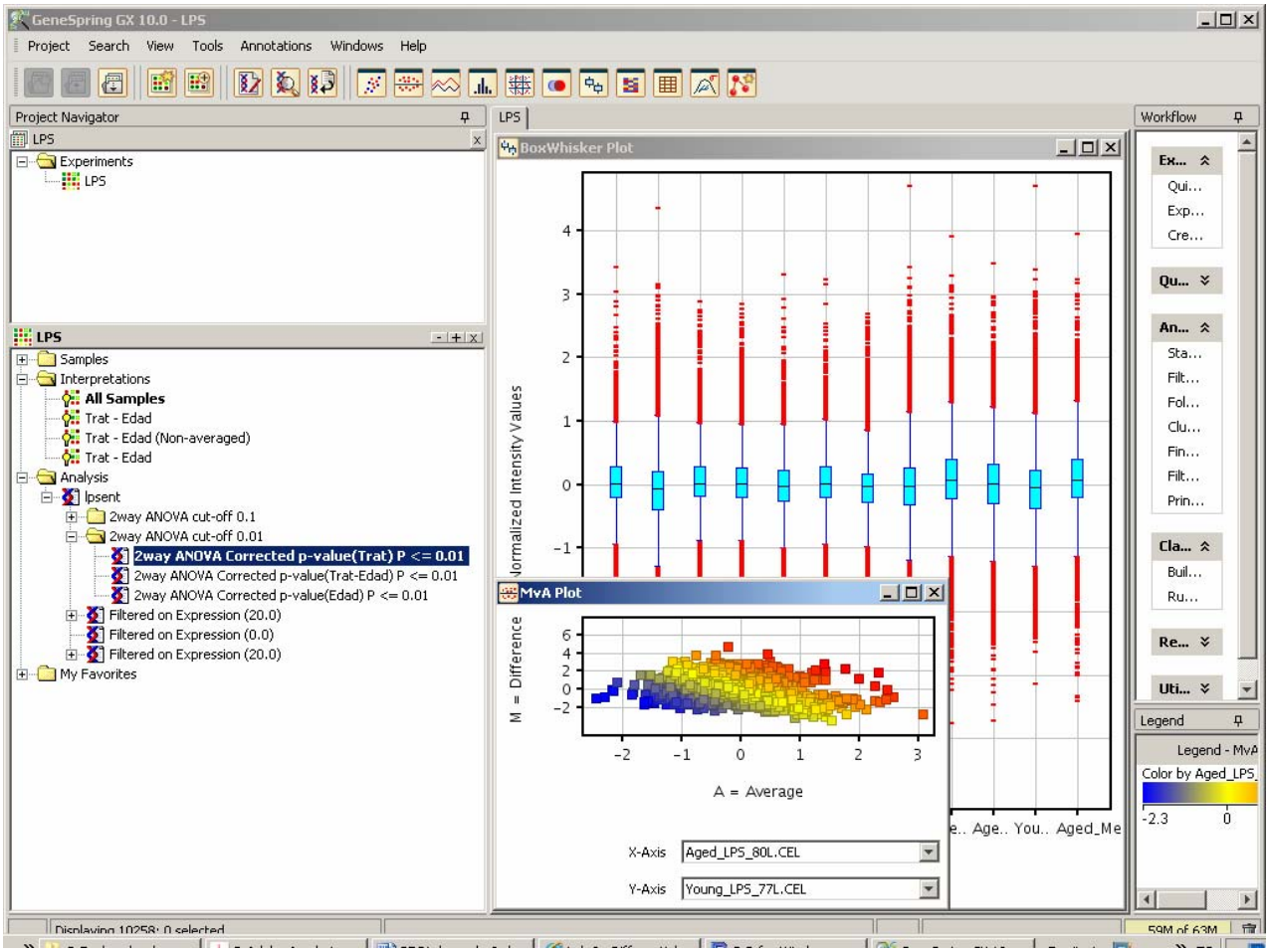
Y el diagrama de perfiles de expresión



Nota Final ….. Con GenesPring se hace en 10 minutos (incluyendo instalación) y sale mas o menos como en el artículo. Y puedes ver fácilmente como se cuelan los “probes de control” y más cosas. La curva de aprendizaje seria mucho más rapida¡¡¡¡ , eso si, lo que hemos aprendido seguro que no se nos olvida.



El venn es casi como el del articulo



Referencias

1) http://www.affymetrix.com/products_services/arrays/specific/mouse430_2.affx#1_1
2)
3) http://race.unil.ch/
4) http://gepas.bioinfo.cipf.es/
5) https://carmaweb.genome.tugraz.at/carma/