Caso resuelto 2: Detección de efecto *batch* en datos de alto rendimiento

Alex Sánchez y Francesc Carmona

Índice

1. Introducción

Muchos datos de alto rendimiento como los microarrays o los de proteómica presentan un tipo particular de complicación técnica que se conoce como efecto *batch*. Dicho efecto consiste en que muestras producidas en el mismo "lote" (mismo día, misma tanda de hibridación, mismo operario, ...) se parecen más entre ellas que muestras producidas en lotes distintos pudiendo llegar a causar confusión en estudios comparativos en los que el efecto batch enmascare el efecto de los tratamientos en estudio.

Si la presencia del efecto batch se conoce o espera *a priori* puede, en principio, ser controlado o cancelado mediante un adecuado diseño experimental, en el que el lote se tratará usualmente como un bloque. En este caso un adecuado balanceo entre bloques y tratamientos puede cancelar el efecto batch. Alternativamente, dicho efecto puede incluirse como un factor en el modelo de análisis de la varianza, lo que eliminara del análisis la variación atribuible al lote.

En muchas (demasiadas) ocasiones el efecto batch no ha sido considerado de antemano, o incluso, cuando sí lo ha sido, puede que no se hayan tenido en cuenta todos los posible efectos (a lo mejor se ha tenido en cuenta el día pero no que los reactivos utilizados provenían de dos lotes o que las piezas se han procesado en grupos, o ...).

En prevención de los problemas que esto puede generar es conveniente realizar algún tipo de análisis exploratorio que permita detectar la posible presen-

cia de estos efectos. En caso de detectarse efectos indeseados puede plantearse su eliminación mediante alguna de las técnicas disponibles para ello.

2. Los datos para el análisis

Los datos para este ejemplo consisten en datos de microarrays de expresión génica, tipo hgu95-a utilizados para analizar un estudio de cancer de mama en el que se analiza el efecto de distintos tratamientos y distintos tiempos de exposición a éstos, sobre la expresión de los genes.

El estudio se halla disponible en la base de datos Gene Expression Omnibus (GEO, `http://www.ncbi.nlm.nih.gov/geo` con número de acceso "GSE848". Para este ejemplo hemos utilizado un subconjunto de estos datos basado en 18 muestras, localizado en el archivo `Breast-Cancer.txt`.

Este archivo, a diferencia de lo habitual en datos de alto rendimiento, contiene las variables (genes) en columnas y las muestras en las filas. Lo primero es leer los datos y separar la información sobre grupos y lotes de las observaciones.

```
> data <-read.table(file.path("data", "Breast_Cancer.txt"), head=T, sep="\t", as.is=TRUE)
> # Fora els controls
> data <- data[-(1:2),]
> targets <- data[,1:5]
> #treure el simbol "+"
> targets[targets=="E2+ICI"]<- "E2.ICI"
> targets[targets=="E2+Ral"]<- "E2.Ral"
> targets[targets=="E2+TOT"]<- "E2.TOT"
> sampleNames <- targets$ShortName <- paste(targets$Treatment,targets$Time,targets$Batch, se
> targets <-targets[,-5]
> x <- t(data[,-(1:5)])
> colnames(x) <-sampleNames
> head(x)

          E2.8.A E2.8.B E2.48.A E2.48.B E2.ICI.8.A E2.ICI.8.B E2.ICI.48.A
X100_g_at  7.934  8.098   8.256   7.619      7.800      7.887       8.016
X101_at    6.878  6.811   6.803   6.990      6.816      7.043       6.717
X102_at    4.492  3.121   4.655   3.878      3.536      2.766       4.343
X103_at    5.244  3.350   3.018   4.278      3.420      2.536       3.916
X104_at    5.174  4.843   5.350   4.597      4.209      4.991       5.186
X105_at    2.104  1.807   2.263   2.536      3.485      1.585       4.036
          E2.ICI.48.B E2.Ral.8.A E2.Ral.8.B E2.Ral.48.A E2.Ral.48.B E2.TOT.8.A
X100_g_at       7.879      7.660      7.613       7.799       7.676      8.126
X101_at         6.772      6.516      6.775       6.206       6.842      6.624
X102_at         3.018      3.350      2.963       4.154       3.170      4.322
X103_at         3.954      3.202      2.459       4.567       2.766      2.722
X104_at         4.781      5.315      4.921       5.229       4.807      4.672
X105_at         4.322      2.807      3.722       4.154       3.459      4.233
          E2.TOT.8.B E2.TOT.48.A E2.TOT.48.B
```

```
X100_g_at       8.038       8.249       7.720
X101_at         6.889       6.675       7.130
X102_at         1.536       4.036       3.858
X103_at         2.511       3.104       3.104
X104_at         5.256       5.057       5.186
X105_at         2.322       3.766       2.202
```

```
> save(x, targets, file=file.path("data","dades.Rda"))
```

La tabla 1 describe los grupos y covariables de que consta el subconjunto de datos analizado:

	Filename	Treatment	Time	Batch	ShortName
3	GSM13099.txt	E2	8	A	E2.8.A
4	GSM13138.txt	E2	8	B	E2.8.B
5	GSM13139.txt	E2	48	A	E2.48.A
6	GSM13140.txt	E2	48	B	E2.48.B
7	GSM15900.txt	E2.ICI	8	A	E2.ICI.8.A
8	GSM15901.txt	E2.ICI	8	B	E2.ICI.8.B
9	GSM15902.txt	E2.ICI	48	A	E2.ICI.48.A
10	GSM15903.txt	E2.ICI	48	B	E2.ICI.48.B
11	GSM15904.txt	E2.Ral	8	A	E2.Ral.8.A
12	GSM15905.txt	E2.Ral	8	B	E2.Ral.8.B
13	GSM15906.txt	E2.Ral	48	A	E2.Ral.48.A
14	GSM15907.txt	E2.Ral	48	B	E2.Ral.48.B
15	GSM15908.txt	E2.TOT	8	A	E2.TOT.8.A
16	GSM15909.txt	E2.TOT	8	B	E2.TOT.8.B
17	GSM15910.txt	E2.TOT	48	A	E2.TOT.48.A
18	GSM15911.txt	E2.TOT	48	B	E2.TOT.48.B

Cuadro 1: Tabla de grupos y covariables presentes en el estudio analizado

3. Análisis del efecto batch

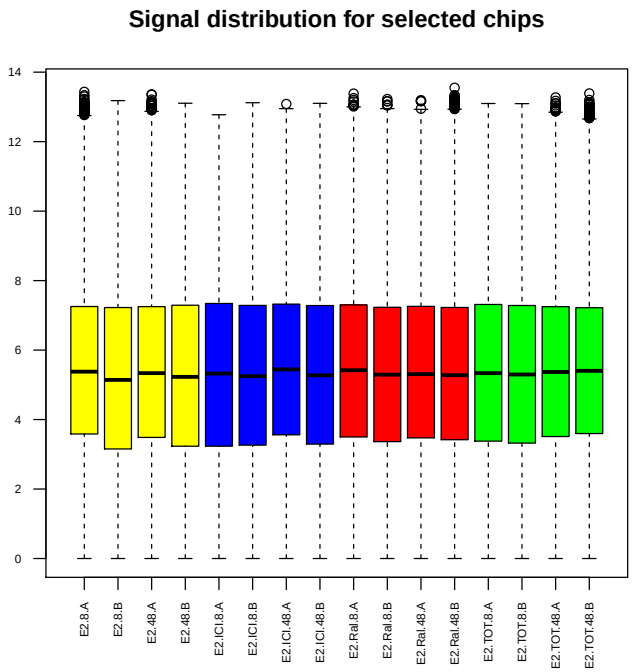
La detección de efectos batch puede hacerse de diversas formas pero esencialmente, de lo que se trata es de ver si las muestras se agrupan por causas distintas a las que se podría esperar, por ejemplo si en vez de parecerse más entre si muestras que han recibido un mismo tratamient, lo hacen las que han recibido algun tratamiento el mismo día.

La detección de estos efectos puede hacerse mediante distintas técnicas de las que, la más popular es el análisis de componentes principales. Una vez detectado si existe efecto batch es posible mirar de eliminarlo. Si cada tratamiento en estudio está presente en cada lote suele ser posible separar ambos efectos. Si, no es así no suele ser posible evitar cierto grado de *confusión* tratamiento–lote.

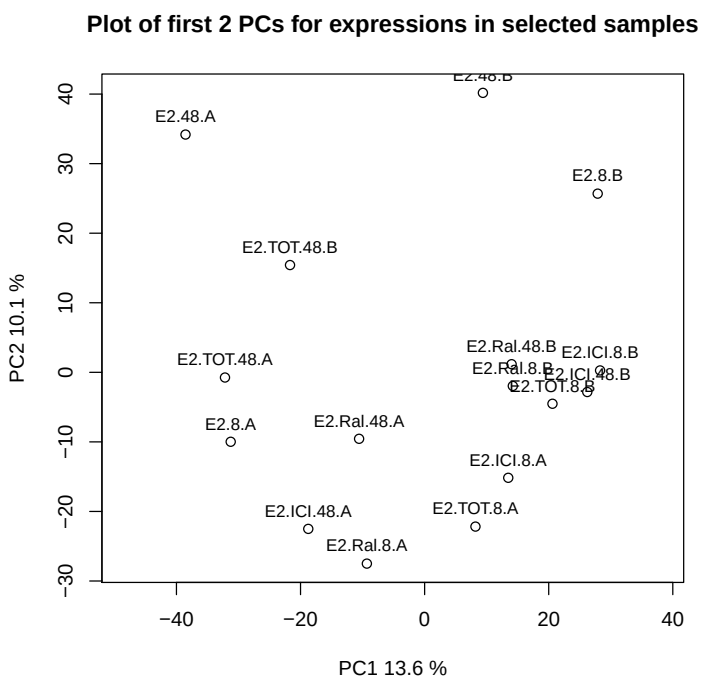
3.1. Detectando el efecto batch

Un diagrama de cajas no muestra ninguna tendencia clara que separe los lotes A y B

```
> boxplot(as.data.frame(x), cex.axis=0.6, col=colores, las=2, names=sampleNames,
+         main="Signal distribution for selected chips")
```



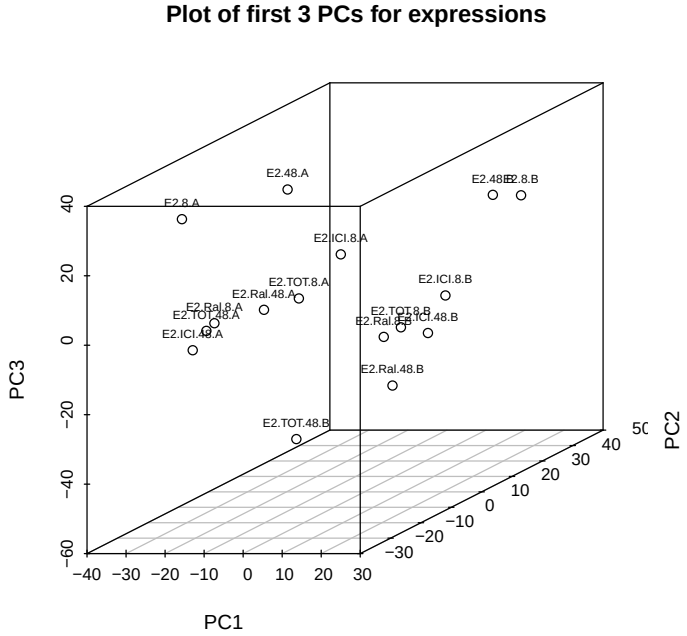

```
> plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE)
+ {
+   pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))
+   loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
+   xlab<-c(paste("PC1",loads[1],"%"))
+   ylab<-c(paste("PC2",loads[2],"%"))
+   if (is.null(colors)) colors=1
+   plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors,
+        xlim=c(min(pcX$x[,1])-10, max(pcX$x[,1])+10))
+   text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=0.8)
+   title(paste("Plot of first 2 PCs for expressions in", dataDesc, sep=" "), cex=0.8)
+ }

> plotPCA(x, labels=sampleNames, dataDesc="selected samples")
```



Es imediato ver que las muestras del grupo A y B se encuentran salvo excepciones agrupadas en cuadrantes distintos.

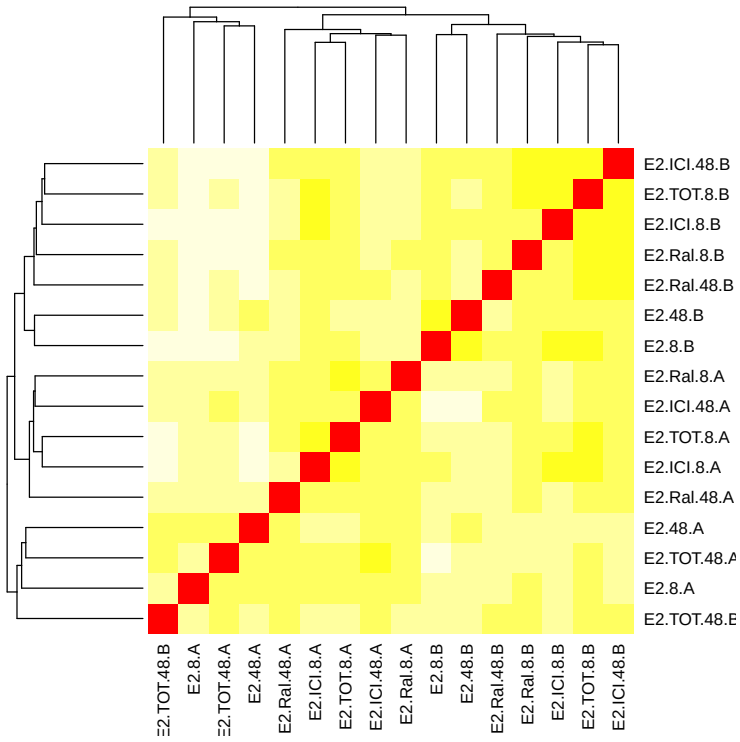
Un gráfico en tres dimensiones permite visualizar mejor esta heterogeneidad

```
> if(!(require(scatterplot3d))) install.packages("scatterplot3d")
> require(scatterplot3d)
> label<- sampleNames
> pcX<-prcomp(t(x), scale=FALSE) # o prcomp(t(X))
> res3d<-scatterplot3d(pcX$x[,1:3], angle=20)
> text(res3d$xyz.convert(pcX$x[,1], pcX$x[,2], pcX$x[,3]),
+      labels=sampleNames, pos=3, cex=0.6)
> title(paste("Plot of first 3 PCs for expressions"), cex=0.8)
```



Un enfoque alternativo aunque relacionado es realizar un análisis basado en distancias. Podemos hacerlo calculando y visualizando la matriz de distancias mediante un mapa de colores o escalamiento multidimensional.

```
> manDist <- dist(t(x))
> heatmap (as.matrix(manDist), col=heat.colors(16))
```




```
> require(MASS)
> sam1<-sammon (manDist, trace=FALSE)
> plot(sam1$points)
> text(sam1$points, targets$Batch, pos=4)
```

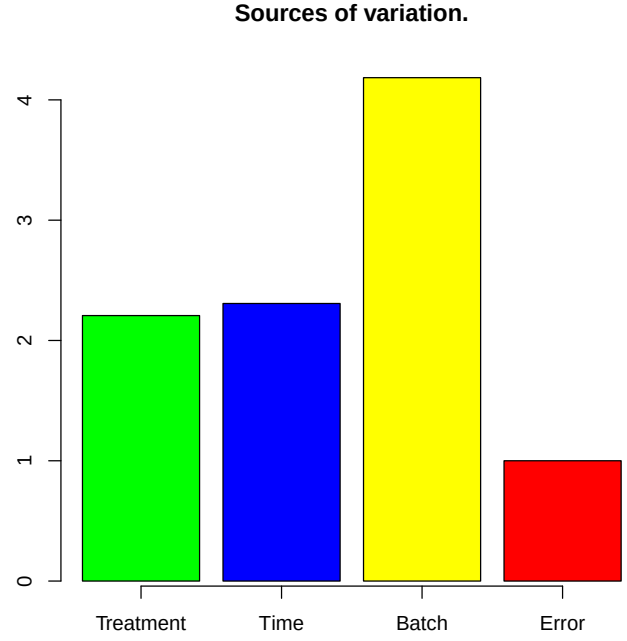
Todas las visualizaciones coinciden en mostrar una separación asociada al factor batch A o B.

3.2. Cuantificación del efecto batch

Si se construye un modelo de análisis de la varianza apropiado es posible cuantificar la importancia media del efecto batch y compararla con el efecto de los tratamientos.

```
> # Prova
> x1<-x[1,]
> treat <- as.factor(targets$Treatment)
> time <-  as.factor(targets$Time)
> batch <-  as.factor(targets$Batch)
> mimod <- x1 ~treat+time+batch
> aov1<-aov(mimod)
> s<-summary(aov1)
```

```
> Fs<- s[[1]][,3]/s[[1]][4,3]
> myLM <- function(x){
+   mimod <- x ~ treat+time+batch
+   s<-summary(aov(mimod))
+   return(s[[1]][,3]/s[[1]][4,3])
+   }
> Fs<- apply(x,1,myLM)
> M <-apply(Fs,1,  mean, na.rm=T)
> names(M) <- rownames(Fs)<- c("Treatment", "Time", "Batch", "Error")

> barplot(M, col=c("green", "blue", "yellow", "red"), axis.lty=1,
+         main="Sources of variation.\n")
```



Como puede verse la principal fuente de variación es el efecto lote.

4. Eliminación del efecto batch

Para cuantificar el efecto batch hemos ajustado un modelo lineal. Dicho modelo permite estimar de los efectos de cada factor. Una forma de eliminar el efecto batch es restar de las observaciones los efectos estimados.

Otra opción más simple es centrar los datos respecto de las medias de cada batch.

Por ejemplo un gen con importante efecto batch es:

```
> highF <- quantile(Fs["Batch",],0.9, na.rm=TRUE)
> lowF <-  quantile(Fs["Batch",],0.1, na.rm=TRUE)
> hFs<- which(Fs[,3] >= highF)[1]
> Fs[,3]

Treatment      Time     Batch     Error
    1.531     6.371    18.566     1.000

> summary(aov(x[3,] ~treat+time+batch))

            Df Sum Sq Mean Sq F value Pr(>F)
treat        3   1.14    0.38    1.53 0.2663
time         1   1.58    1.58    6.37 0.0302 *
batch        1   4.60    4.60   18.57 0.0015 **
Residuals   10   2.48    0.25
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Si a los valores del batch A les restamos su media y a los del B la suya obtenemos

```
>   meanA <- mean(x[3,batch=="A"])
>   meanB <- mean(x[3,batch=="B"])
>   xAdj <-  ifelse(batch=="A", x[3,]-meanA, x[3,]-meanB)
>   mimodAdj <- xAdj ~treat+time+batch
>   aovAdj<-aov(mimodAdj)
>   summary(aovAdj)

            Df Sum Sq Mean Sq F value Pr(>F)
treat        3   1.14   0.379    1.53   0.27
time         1   1.58   1.578    6.37   0.03 *
batch        1   0.00   0.000    0.00   1.00
Residuals   10   2.48   0.248
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

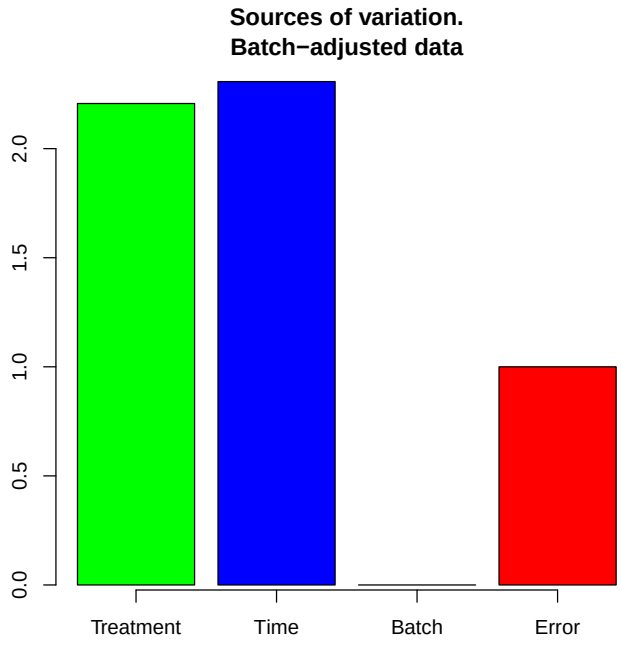
Esto puede hacerse para todos los genes a la vez:

```
> bcX <- cbind(t(scale(t(x[,batch=="A"]), center=TRUE, scale=FALSE)),
+              t(scale(t(x[,batch=="B"]), center=TRUE, scale=FALSE)))
> bcX <-bcX[, colnames(x)]
> # colnames(x)== colnames(bcX)
```

Como puede verse esta operación ha eliminado el efecto Batch de los datos

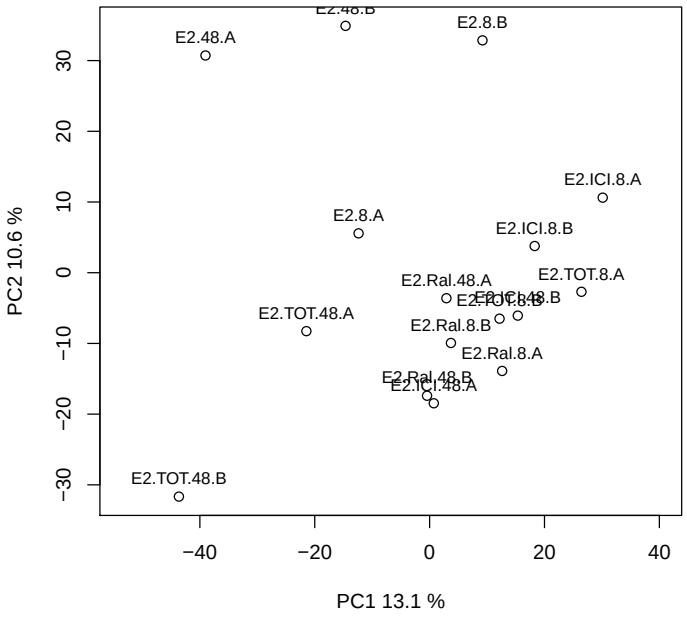
```
> adjFs<- apply(bcX,1,myLM)
> adjM <-apply(adjFs,1,  mean, na.rm=T)
> names(adjM) <- rownames(adjFs)<- c("Treatment", "Time", "Batch", "Error")
```

```
> barplot(adjM, col=c("green", "blue", "yellow", "red"), axis.lty=1,
+         main="Sources of variation.\nBatch-adjusted data")
```



El análisis de componentes principales también muestra la desaparición del efecto batch:

```
> label<- sampleNames
> plotPCA(bcX, labels=label, dataDesc="Batch-adjusted data")
```


Plot of first 2 PCs for expressions in Batch−adjusted data
E2.48.B
E2.8.B
E2.48.A
E2.ICI.8.A
E2.8.A
E2.ICI.8.B
E2.TOT.8.A
E2.Ral.48.A
E2.TOT.8.B
E2.TOT.48.A
E2.Ral.8.B
E2.Ral.8.A
E2.Ral.48.B
E2.ICI.48.A
E2.TOT.48.B
PC2 10.6 %
PC1 13.1 %
30
20
10
0
−10
−20
−30
−40
−20
0
20
40