Limma

Guillermo Ayala Gallego

2024-04-09

Table of contents

El método limma

Referencia

```
Smyth, G. K. Linear Models and Empirical Bayes Methods for
Assessing Differential Expression in Microarray Experiments
Statistical Applications in Genetics and Molecular Biology,
2004, 1, 3
```

Lo previo

- Sirve cuando la respuesta es de caracter continuo (DNA microarrays, Methylation array, ...)
- Respuesta aleatoria:
$$\mathbf{Y}_i = (Y_{i1}, ..., Y_{in})^T.$$
- Valores observados:
$$\mathbf{y}_i = (y_{i1}, ..., y_{in})^T.$$

- Para cada muestra nuestros predictores son las variables fenotípicas observadas en dicha muestra: $(\mathbf{x}_1, ..., \mathbf{x}_n)$, con $\mathbf{x}_j \in \mathbb{R}^p$.
- Denotamos: $E\mathbf{Y}_i = (EY_{i1}, ..., EY_{in})^T$.
  - Asumimos:
$$EY_{ij} = \beta_{i1}x_{j1} + ... + \beta_{ip}x_{jp} = \mathbf{x}_j^T \ _i,$$
    siendo:
  - $\mathbf{x}_j = (x_{j1}, ..., x_{jp})^T$ es el vector de covariables para la j-ésima muestra,
  - $_i = (\beta_{i1}, ..., \beta_{ip})^T$ el vector de coeficientes.

- Sobre la matriz de varianzas de $\mathbf{Y}_i$
$$var(\mathbf{Y}_i) = \sigma_i^2 \mathbf{V}_i,$$
donde $\mathbf{V}_i$ es una matriz definida positiva conocida.

Contrastes

- Estamos interesados en ciertos contrastes sobre el vector de coeficientes $_i$ que corresponden al diseño utilizado en la experiencia.
- Un contraste tiene la forma general

$$\mathbf{a}_i = \mathbf{C}^T \ _i.$$

- Se quiere contrastar

$$H_0 : a_{ij} = 0,$$
$$H_1 : a_{ij} \neq 0.$$

- **Ajustamos un modelo lineal para cada gen** y obtenemos:
- Los estimadores $\hat{}_i$,
- el estimador s_i^2 de σ_i^2 y
- la matriz de covarianzas estimada

$$\widehat{var(\hat{}_i)} = s_i^2 \mathbf{V}_i$$

siendo $\mathbf{V}_i$ una matriz definida positiva que no depende de s_i^2.

- Los estimadores de los contrastes son

$$\hat{\mathbf{a}}_i = \mathbf{C}^T \hat{}_i,$$

y de su matriz de covarianzas es

$$var(\hat{\mathbf{a}}_i) = \sigma_i^2 \mathbf{C}' \mathbf{V}_i \mathbf{C}.$$

La correspondiente matriz de covarianzas estimada es la obtenemos sustituyendo σ_i^2 por s_i^2.

Hipótesis sobre los estimadores

- No se asume necesariamente normalidad para $\mathbf{Y}_i$.
- Tampoco asumimos que el ajuste de los modelos lineales se hace mediante el procedimiento de los mínimos cuadrados.
- **Sí** se asume que:
  - $\hat{\mathbf{a}}_i$ tiene una distribución aproximadamente normal con vector de medias $\mathbf{a}_i$ y matriz de covarianzas $\sigma_i^2 \mathbf{C}^T \mathbf{V}_i \mathbf{C}$.
- También asumimos que s_i^2 siguen aproximadamente una distribución ji-cuadrado escalada.

Consecuencias

- Si denotamos por v_{ij} el j-ésimo elemento de la diagonal de $\mathbf{C}'\mathbf{V}_i\mathbf{C}$ entonces todas las hipótesis indicadas previamente se traducen en los siguientes resultados.
- $\hat{a}_{ij}|a_{ij}, \sigma_i^2 \sim N(a_{ij}, v_{ij}\sigma_i^2)$.
- $s_i^2|\sigma_i^2 \sim \frac{\sigma_i^2}{d_i}\chi^2_{d_i}$ siendo d_i los grados de libertad residuales del modelo ajustado para el i-ésimo gen.
- Asumiendo estas hipótesis se tiene que el siguiente estadístico

$$t_{ij} = \frac{\hat{\beta}_{ij}}{s_i\sqrt{v_{ij}}}$$

  aproximadamente tiene una distribución t de Student con d_i grados de libertad.
- En lo que sigue se asumirá (y no tienen porqué ser cierto) que $\hat{\beta}_i$ y s_i^2 son independientes para los distintos genes.

Modelo jerárquico

- Tenemos muchos test simultáneos.
- Se trata de modelizar el comportamiento entre genes.
- Se opta por modelizar mediante distribuciones de probabilidad sobre los parámetros.
- Es un modelo bayesiano.

Distribuciones a priori sobre los parámetros

- Sobre σ_i^2 la siguiente distribución,

$$\frac{1}{\sigma_i^2} \sim \frac{1}{d_0 s_0^2}\chi^2_{d_0}.$$

- Para un j dado, suponemos que β_{ij} es no nula con una probabilidad conocida

$$P(\beta_{ij} \neq 0) = p_j.$$

- Notemos que p_j tiene el sentido de la proporción esperada de genes que se expresan diferencialmente.
- Para los coeficientes no nulos se asume

$$\beta_{ij}\Big|\sigma_i^2, \beta_{ij} \neq 0 \sim N(0, v_{0j}\sigma_i^2).$$

Distribuciones a posteriori

Asumiendo el modelo jerárquico que acabamos de especificar se tiene que la media de la distribución a posteriori de $1/\sigma_i^2$ condicionada a s_i^2 viene dada por

$$E\left[\frac{1}{\sigma_i^2}\middle| s_i^2\right] = \frac{1}{\tilde{s}_i^2},$$

con

$$\tilde{s}_i^2 = \frac{d_0 s_0^2 + d_i s_i^2}{d_0 + d_i}.$$

- Podemos definir el estadístico t moderado como

$$\tilde{t}_{ij} = \frac{\hat{\beta}_{ij}}{\tilde{s}_i^2 \sqrt{v_{ij}}}.$$

- Se demuestra que los t-estadísticos moderados $\tilde{t}_{ij}$ y las varianzas muestrales residuales s_i^2 se distribuyen independientemente.
- Bajo la hipótesis nula $H_0 : \beta_{ij} = 0$, el t-estadístico moderado $\tilde{t}_{ij}$ sigue una distribución t de Student con $d_i + d_0$ grados de libertad.
- Los grados de libertad que estamos añadiendo d_0 expresan la ganancia de información que obtenemos de utilizar todos los genes siempre **asumiendo el modelo jerárquico**.
- Los valores d_0 y s_0 que se suponen conocidos en lo previo serán estimados a partir de los datos.
- Tenemos un método empírico bayesiano.

Odds a posteriori

- Se consideran los **odds a posteriori** dados por

$$O_{ij} = \frac{P(\beta_{ij} \neq 0 | \tilde{t}_{ij,s_i^2})}{P(\beta_{ij} = 0 | \tilde{t}_{ij,s_i^2})}.$$

- Estos odds prueban que tienen la siguiente expresión.

$$O_{ij} = \frac{p_j}{1-p_j}\left(\frac{v_{ij}}{v_{ij}+v_{0j}}\right)^{1/2}\left(\frac{\tilde{t}_{ij}^2 + d_0 + d_i}{\tilde{t}_{ij}^2 \frac{v_{ij}}{v_{ij}+v_{oj}} + d_o + d_i}\right)^{(1-d_0+d_i)/2}$$

- Se propone trabajar con los **log-odds** dados por

$$B_{ij} = \ln O_{ij}.$$

Relación con método SAM

- Es similar en espíritu pero no es lo mismo.
- Si los grados de libertad verifican $d_o < +\infty$ y $d_i > 0$ entonces los t-estadísticos moderados tienen la siguiente expresión

$$\tilde{t}_{ij} = \left(\frac{d_0 + d_i}{d_i}\right)^{1/2} \frac{\hat{\beta}_{ij}}{\sqrt{s^2_{*,i} v_{ij}}},$$

  siendo

$$s^2_{*,i} = s^2_i + \frac{d_0}{d_i} s^2_0.$$

- Cada varianza muestral es incrementada un valor positivo en principio distinto para cada gen.
- La idea de SAM consiste en considerar

$$t_{*,ij} = \frac{\hat{\beta}_{ij}}{(s_i + a)\sqrt{v_{ij}}}$$

  donde a es un valor positivo determinado por un método ad-hoc. En este método lo que se incrementa es la desviación estándar y no la varianza.
- El t-estadístico moderado incrementa la varianza.
- Además el incremento puede ser distinto para cada gen.

gse25171

Datos

```
pacman::p_load(Biobase)
data(gse25171,package="tamidata2")
pData(gse25171)
```

```
                time  time2        Pi replication
GSM618324.CEL.gz    0  Short Treatment           1
GSM618325.CEL.gz    0  Short   Control           2
GSM618326.CEL.gz    1  Short Treatment           3
GSM618327.CEL.gz    1  Short   Control           1
GSM618328.CEL.gz    6 Medium Treatment           2
GSM618329.CEL.gz    6 Medium   Control           3
GSM618330.CEL.gz   24 Medium Treatment           1
```

```
GSM618331.CEL.gz   24 Medium   Control           2
GSM618332.CEL.gz    0  Short Treatment           3
GSM618333.CEL.gz    0  Short   Control           1
GSM618334.CEL.gz    1  Short Treatment           2
GSM618335.CEL.gz    1  Short   Control           3
GSM618336.CEL.gz    6 Medium Treatment           1
GSM618337.CEL.gz    6 Medium   Control           2
GSM618338.CEL.gz   24 Medium Treatment           3
GSM618339.CEL.gz   24 Medium   Control           1
GSM618340.CEL.gz    0  Short Treatment           2
GSM618341.CEL.gz    0  Short   Control           3
GSM618342.CEL.gz    1  Short Treatment           1
GSM618343.CEL.gz    1  Short   Control           2
GSM618344.CEL.gz    6 Medium Treatment           3
GSM618345.CEL.gz    6 Medium   Control           1
GSM618346.CEL.gz   24 Medium Treatment           2
GSM618347.CEL.gz   24 Medium   Control           3
```

```
dim(gse25171)
```

```
Features  Samples
   22746       24
```

- ¿Cómo pueden influir el tiempo de observación de la muestra `time` y la presencia o no de fósfatos en la expresión del gen `Pi` en la expresión de cada una de las sondas que tenemos en el microarray?

time

```
design = model.matrix(~ pData(gse25171)[,"time"])
head(design)
```

```
  (Intercept) pData(gse25171)[, "time"]
1           1                         0
2           1                         0
3           1                         1
4           1                         1
5           1                         6
6           1                         6
```

- Cambiamos los nombres de la columnas por razones estéticas.

```
colnames(design) = c("intercept","time")
```

- Ajustamos todos los modelos lineales: mismas predictoras y como respuesta cada sonda.

```
fit = limma::lmFit(gse25171,design)
```

```
fit$coefficients
coef(fit)
coefficients(fit)
```

- Vemos los primeros coeficientes y errores estándar estimados.

```
head(coef(fit),n=2)
```

```
          intercept         time
244901_at  5.086076  0.003844458
244902_at  4.843428 -0.005471131
```

```
head(fit$sigma,n=2)
```

```
244901_at 244902_at
0.2840462 0.2291651
```

- Aplicamos el método limma.

```
fit1 = limma::eBayes(fit)
```

- Tenemos los t-estadísticos moderados con

```
head(fit1$t)
```

```
          intercept       time
244901_at  70.20698  0.6569507
244902_at  81.97642 -1.1463402
244903_at  61.65966  1.0441666
244904_at  63.89556 -0.2663003
244905_at 112.93409  0.5087080
244906_at  82.57484  0.2435055
```

- Los valores B o logaritmo natural del cociente de odds a posteriori.

```
head(fit1$lods)
```

```
          intercept      time
244901_at  57.59049 -7.604303
244902_at  61.33907 -7.161476
244903_at  54.42776 -7.272309
244904_at  55.29720 -7.789878
244905_at  68.93538 -7.692872
244906_at  61.51406 -7.795904
```

- El valor estimado de s_0^2 y de d_0.

```
fit1$s2.prior
```

```
[1] 0.03494128
```

```
fit1$df.prior
```

```
[1] 2.161468
```

- Tenemos los estimadores de los errores estándar a posteriori.

```
head(fit$sigma)
```

```
244901_at 244902_at 244903_at 244904_at 244905_at 244906_at
0.2840462 0.2291651 0.4448048 0.3357318 0.1306061 0.2972859
```

```
head(fit1$s2.post)
```

```
 244901_at  244902_at  244903_at  244904_at  244905_at  244906_at
0.07659027 0.05094436 0.18327749 0.10575819 0.01865779 0.08359841
```

- La función `limma::topTable()` nos hace un resumen de los ajustes.
- Ordena (`sort.by`) las sondas por su p-valor original.
- Tenemos el coeficiente estimado en el ajuste `logFC` cuyo nombre induce a confusión.

```
limma::topTable(fit1,coef=2,adjust ="BH")
```

```
            PROBEID ENTREZID         GO EVIDENCE ONTOLOGY      TAIR       logFC
261892_at 261892_at   844423 GO:0000976      IDA       MF AT1G80840 -0.13307030
246253_at 246253_at   829880 GO:0000976      IPI       MF AT4G37260 -0.04069718
262590_at 262590_at   838073 GO:0004842      IDA       MF AT1G15100  0.04440937
250781_at 250781_at   830424 GO:0000976      IBA       MF AT5G05410 -0.07274667
247467_at 247467_at   836333 GO:0005783      IDA       CC AT5G62130  0.04163427
261143_at 261143_at   838565 GO:0005345      ISS       MF AT1G19770 -0.05257990
252958_at 252958_at   830018 GO:0000976      IPI       MF AT4G38620  0.04205502
266316_at 266316_at   817250 GO:0003674       ND       MF AT2G27080 -0.07138841
266658_at 266658_at   817115 GO:0002237      IEA       BP AT2G25735 -0.08795516
248870_at 248870_at   834714 GO:0003674       ND       MF AT5G46710 -0.04162049
            AveExpr         t      P.Value   adj.P.Val        B
261892_at  6.937141 -7.447518 1.048025e-07 0.002383838 7.132046
246253_at  8.829745 -7.002851 2.947146e-07 0.003351789 6.078721
262590_at 10.156525  6.418349 1.191073e-06 0.007960638 4.658987
250781_at  6.178312 -6.247306 1.805989e-06 0.007960638 4.236660
247467_at  6.528512  6.245277 1.814965e-06 0.007960638 4.231632
261143_at  9.696203 -6.185676 2.099878e-06 0.007960638 4.083795
252958_at  7.738408  6.120231 2.465583e-06 0.008011734 3.921082
266316_at  8.375071 -5.866638 4.612016e-06 0.013113114 3.287079
266658_at  6.948016 -5.707526 6.853282e-06 0.016154197 2.886710
248870_at  7.186558 -5.674216 7.447863e-06 0.016154197 2.802669
```

- ¿Cuántas sondas tienen un coeficiente significativamente no nulo?

```
padj = limma::topTable(fit1,coef=2,adjust ="BH",
                       number=nrow(gse25171))[,"adj.P.Val"]
table(padj < 0.05)
```

```
FALSE  TRUE
22707    39
```

time + Pi

- Construimos la matriz de modelo.

```
design = model.matrix(~  pData(gse25171)[,"time"] +
                         pData(gse25171)[,"Pi"] )
```

```
colnames(design) = c("constante","time","Pi")
head(design)
```

```
  constante time Pi
1         1    0  1
2         1    0  0
3         1    1  1
4         1    1  0
5         1    6  1
6         1    6  0
```

- Ajustamos los modelos.

```
fit = limma::lmFit(gse25171,design)
fit1 = limma::eBayes(fit)
```

- Evaluamos si los coeficientes correspondientes a la columna 2 `coef=2` son nulos.
- La columna 2 corresponde con la variable `time`.

```
tt2 = limma::topTable(fit1,coef=2,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
head(tt2)
```

```
            PROBEID ENTREZID         GO EVIDENCE ONTOLOGY      TAIR
244901_at 244901_at     <NA> GO:0000276      IBA       CC ATMG00640
244902_at 244902_at     <NA> GO:0003954      TAS       MF ATMG00650
244903_at 244903_at     <NA> GO:0003674       ND       MF ATMG00660
244904_at 244904_at     <NA> GO:0003674       ND       MF ATMG00670
244905_at 244905_at     <NA> GO:0003674       ND       MF ATMG00680
244906_at 244906_at     <NA> GO:0003674       ND       MF ATMG00690
                 logFC  AveExpr          t   P.Value adj.P.Val         B
244901_at  0.003844458 5.115871  0.6437893 0.5260296 0.9615362 -7.623012
244902_at -0.005471131 4.801026 -1.1232817 0.2728145 0.8610623 -7.197140
244903_at  0.009452352 6.983152  1.0371660 0.3103548 0.8833746 -7.289372
244904_at -0.001831236 5.425114 -0.2725916 0.7875801 0.9907220 -7.798308
244905_at  0.001469313 4.049431  0.4987818 0.6226323 0.9720791 -7.708075
244906_at  0.001488755 6.261286  0.2387066 0.8134303 0.9933244 -7.807305
```

- Evaluamos si los coeficientes correspondientes a la columna 3 `coef=3` son nulos.
- La columna 3 corresponde con la variable `Pi`.

```
tt3 = limma::topTable(fit1,coef=3,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
```

- ¿Cualés y cuántas sondas tienen el coeficiente correspondiente a `time` significativamente no nulo ajustando por el método de Benjamini-Hochberg?

```
tt2.row = which(tt2[,"adj.P.Val"] < .05)
length(tt2.row)
```

```
[1] 146
```

- ¿Cualés y cuántas sondas tienen el coeficiente correspondiente a `Pi` significativamente no nulo ajustando por el método de Benjamini-Hochberg?

```
tt3.row = which(tt3[,"adj.P.Val"] < .05)
length(tt3.row)
```

```
[1] 34
```

- ¿Y cuántas sondas tienen coeficiente significativamente no nulo tanto para una como para la otra variable?

```
intersect(tt2.row,tt3.row)
```

```
 [1]  1118  2308  3025  3894  5776  6740  8384  9258 12851 13892 14979 16243
[13] 16293 16992 17038 20376 21552
```

- ¿Qué sondas son?

```
tt2[intersect(tt2.row,tt3.row),"PROBEID"]
```

```
 [1] "246018_at" "247208_at" "247925_at" "248794_at" "250676_at" "251640_at"
 [7] "253284_at" "254158_at" "257751_at" "258792_at" "259879_at" "261143_at"
[13] "261193_at" "261892_at" "261938_at" "265276_at" "266452_at"
```

¿A qué genes corresponden utilizando su código `ENTREZID`?

```
tt2[intersect(tt2.row,tt3.row),"ENTREZID"]
```

```
 [1] "830934"   "836610"   "835860"   "834768"   "830520"   "28719408"
 [7] "829563"   "828540"   "821400"   "819622"   "843998"   "838565"
[13] "28717292" "844423"   "838857"   "817388"   "818933"
```

time * Pi

- ¿Hay interacción entre las variables predictoras `time` y `Pi`?
- Observemos que sustituimos `+` por `*`.
- La nueva columna de la matriz de modelo se define como el producto de las columnas 2 y 3 correspondientes a las variables originales `time` y `Pi`.
- El modelo es distinto, los coeficientes estimados para la nueva variable y para las ya existentes son distintos y no necesariamente tendremos las mismas sondas.

```
design = model.matrix(~ pData(gse25171)[,"time"] *
                          pData(gse25171)[,"Pi"])
colnames(design) = c("constante","time","Pi","time:Pi")
head(design)
```

```
  constante time Pi time:Pi
1         1    0  1       0
2         1    0  0       0
3         1    1  1       1
4         1    1  0       0
5         1    6  1       6
6         1    6  0       0
```

- ¿Qué sondas tienen interacciones significativas?

```
fit = limma::lmFit(gse25171,design)
fit1 = limma::eBayes(fit)
tt2 = limma::topTable(fit1,coef=2,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt3 = limma::topTable(fit1,coef=3,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt4 = limma::topTable(fit1,coef=4,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
```

- Ahora podemos evaluar los que tienen interacciones significativas.
- Cuando una sonda tiene una interacción positiva se suele asumir que los efectos principales no se deben eliminar del modelo.
- Si hay interacción quiere decir que influyen significativamente y además de un modo distinto.

```
tt4.row = which(tt4[,"adj.P.Val"] < .05)
tt4[tt4.row,c("PROBEID","ENTREZID")]
```

```
            PROBEID ENTREZID
246071_at 246071_at   832137
246576_at 246576_at   840052
248545_at 248545_at   835091
253386_at 253386_at   829440
263851_at 263851_at     <NA>
266132_at 266132_at   819120
267611_at 267611_at   817207
```

time2 * Pi

- Hemos utilizado la variable `time` como numérica (y lo es).
- Podemos ver que en las variables fenotípicas hay una variable `time2` categórica con dos niveles.

```
pData(gse25171)[,"time2"]
```

```
 [1] Short  Short  Short  Short  Medium Medium Medium Medium Short  Short
[11] Short  Short  Medium Medium Medium Medium Short  Short  Short  Short
[21] Medium Medium Medium Medium
Levels: Short Medium
```

- Ajustamos modelo con `time2` y `Pi`.

```
design = model.matrix(~ pData(gse25171)[,"time2"] *
                        pData(gse25171)[,"Pi"])
colnames(design) = c("constante","time2","Pi","time2:Pi")
head(design)
```

```
  constante time2 Pi time2:Pi
1         1     0  1        0
2         1     0  0        0
3         1     0  1        0
4         1     0  0        0
5         1     1  1        1
6         1     1  0        0
```

```
fit = limma::lmFit(gse25171,design)
fit1 = limma::eBayes(fit)
tt2 = limma::topTable(fit1,coef=2,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt3 = limma::topTable(fit1,coef=3,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt4 = limma::topTable(fit1,coef=4,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt4.row = which(tt4[,"adj.P.Val"] < .05)
tt4[tt4.row,c("PROBEID","ENTREZID")]
```

```
            PROBEID ENTREZID
245571_at 245571_at   827120
245579_at 245579_at   827137
247477_at 247477_at   836355
247965_at 247965_at   835755
248618_at 248618_at   835024
248733_at 248733_at     <NA>
249847_at 249847_at   832385
250172_at 250172_at   831283
251232_at 251232_at   825453
251770_at 251770_at   824763
251914_at 251914_at   824560
252312_at 252312_at   824100
254294_at 254294_at   828406
254313_at 254313_at   828341
255782_at 255782_at   838573
256319_at 256319_at   840493
256783_at 256783_at   820572
257007_at 257007_at   820638
257686_at 257686_at   820462
257697_at 257697_at   820452
258054_at 258054_at   820870
258807_at 258807_at   819558
259106_at 259106_at 28718845
259173_at 259173_at   821201
259351_at 259351_at   819677
259828_at 259828_at   843554
260203_at 260203_at   841722
260386_at 260386_at   843739
260784_at 260784_at   837127
261466_at 261466_at   837282
```

```
261745_at 261745_at   837371
261772_at 261772_at   843957
263680_at 263680_at   839583
263876_at 263876_at   816724
264200_at 264200_at   838871
264672_at 264672_at   837504
264752_at 264752_at   838909
265049_at 265049_at   841635
265707_at 265707_at   814868
266743_at 266743_at   814828
267457_at 267457_at   817946
```

- Evaluamos si no hay interacción pero si efecto de `time2`.

```
sel = which(tt4[,"adj.P.Val"] >= .05 & tt2[,"adj.P.Val"] < .05)
length(sel)
```

```
[1] 2402
```

```
head(tt4[sel,c("PROBEID","ENTREZID")])
```

```
            PROBEID ENTREZID
244932_at 244932_at     <NA>
244933_at 244933_at     <NA>
244941_at 244941_at     <NA>
244947_at 244947_at     <NA>
244994_at 244994_at     <NA>
245011_at 245011_at     <NA>
```

- Podemos evaluar si no hay interacción pero sí efecto de `Pi`.

```
sel = which(tt4[,"adj.P.Val"] >= .05 & tt3[,"adj.P.Val"] < .05)
length(sel)
```

```
[1] 0
```

- Esta forma de plantear el análisis supone cada uno de los coeficientes se ha de interpretar como la modificación respecto de un grupo de referencia.
- Sería cuando las variables categóricas toman el primer nivel.
- El grupo de referencia corresponde `time2 = Short` y `Pi = Control`.

```
  levels(pData(gse25171)[,"time2"])
```

```
[1] "Short"  "Medium"
```

```
  levels(pData(gse25171)[,"Pi"])
```

```
[1] "Control"   "Treatment"
```

- Vamos a trabajar con contrastes.
- Construimos variable que combina ambos factores.

```
  time2Pi = vector("list",ncol(gse25171))
  for(i in seq_along(time2Pi))
    time2Pi[[i]] = paste0(pData(gse25171)[,"time2"][i],
                          pData(gse25171)[,"Pi"][i])
  (time2Pi = factor(unlist(time2Pi)))
```

```
 [1] ShortTreatment  ShortControl    ShortTreatment  ShortControl
 [5] MediumTreatment MediumControl   MediumTreatment MediumControl
 [9] ShortTreatment  ShortControl    ShortTreatment  ShortControl
[13] MediumTreatment MediumControl   MediumTreatment MediumControl
[17] ShortTreatment  ShortControl    ShortTreatment  ShortControl
[21] MediumTreatment MediumControl   MediumTreatment MediumControl
Levels: MediumControl MediumTreatment ShortControl ShortTreatment
```

- Vamos a ajustar un modelo en donde no vamos a incorporar una constante sino que las variables predictoras nos van a indicar cada una de las categorías que tenemos.

```
  design = model.matrix(~  0 + time2Pi)
  colnames(design) = levels(time2Pi)
  head(design)
```

```
  MediumControl MediumTreatment ShortControl ShortTreatment
1             0               0            0              1
2             0               0            1              0
3             0               0            0              1
4             0               0            1              0
5             0               1            0              0
6             1               0            0              0
```

- Realizamos los ajustes.

```
fit = limma::lmFit(gse25171,design)
```

- Definimos los contrastes.

```
cont.matrix = limma::makeContrasts(
    dif1 =  (MediumControl + MediumTreatment)-
        (ShortControl + ShortTreatment),
    dif2 =  (MediumControl + ShortControl)-
        (MediumTreatment + ShortTreatment),
    dif3 = (MediumControl - ShortControl),
    dif4 = (MediumTreatment - ShortTreatment),
    dif5 = (MediumControl - ShortControl) -
        (MediumTreatment - ShortTreatment),
    levels = design)
```

- Hemos construido la siguiente matriz de contrastes.

```
cont.matrix
```

```
                Contrasts
Levels           dif1 dif2 dif3 dif4 dif5
  MediumControl     1    1    1    0    1
  MediumTreatment   1   -1    0    1   -1
  ShortControl     -1    1   -1    0   -1
  ShortTreatment   -1   -1    0   -1    1
```

- Aplicamos limma.

```
fit2 = limma::contrasts.fit(fit,cont.matrix)
fit2 = limma::eBayes(fit2)
```

- Obtenemos resúmenes.

```
tt1 = limma::topTable(fit2,coef=1,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt2 = limma::topTable(fit2,coef=2,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt3 = limma::topTable(fit2,coef=3,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
tt4 = limma::topTable(fit2,coef=4,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
```

```
tt5 = limma::topTable(fit2,coef=5,adjust ="BH",
                      sort.by="none",number=nrow(gse25171))
```

- Nos fijamos como ilustración en el contraste `dif5` que evalua la interacción.

```
sel = which(tt5[,"adj.P.Val"] < .05)
length(sel)
```

```
[1] 41
```

```
head(tt4[sel,c("PROBEID","ENTREZID")])
```

```
            PROBEID ENTREZID
245571_at 245571_at   827120
245579_at 245579_at   827137
247477_at 247477_at   836355
247965_at 247965_at   835755
248618_at 248618_at   835024
248733_at 248733_at     <NA>
```

gse44456

Datos

- Leemos los datos `tamidata::gse44456`.

```
library(Biobase)
data(gse44456,package="tamidata")
```

- Vamos a considerar la variable que nos da el tiempo desde la muerte de la persona.

```
pData(gse44456)[,"postmortenint"]
```

```
 [1] 16.75 27.00 29.00 24.00 24.00 24.00 17.00 12.00 21.00 36.00 19.00 36.00
[13] 68.00 46.00 23.00 48.00 58.50 62.00 30.00 21.00 19.50 24.00 59.50 50.00
[25] 22.00 37.00 41.00 16.00 18.00 16.00  9.00 11.00 20.00 56.00 15.00 43.00
[37] 11.00 32.00 21.50
```

- Cargamos Limma.

```
pacman::p_load(limma)
```

- Determinamos la matriz de diseño.

```
design = model.matrix(~ pData(gse44456)[,"postmortenint"])
colnames(design) = c("intercept","postmortenint")
```

- Podemos ver que la matriz `design`.

```
head(design)
```

```
  intercept postmortenint
1         1         16.75
2         1         27.00
3         1         29.00
4         1         24.00
5         1         24.00
6         1         24.00
```

- Ajustamos los modelos lineales.

```
fit = lmFit(gse44456,design)
```

Los coeficientes ajustados los tenemos con

```
head(fit$coefficients)
```

```
        intercept postmortenint
7892501  2.955131   0.001088853
7892502  4.283626   0.003912042
7892503  3.452548   0.003984278
7892504 10.254139  -0.001083914
7892505  2.944068   0.005281933
7892506  3.224897   0.001557732
```

Ajustamos el modelo jerárquico.

```
fit1 = eBayes(fit)
```

- Si la constante por la que multiplicamos growthrate es nula indicará que no hay una dependencia del nivel de expresión respecto de la tasa de crecimiento celular.
- Veamos los resultados.

```
topTable(fit1,coef=2,adjust ="BH")
```

```
              logFC   AveExpr         t      P.Value   adj.P.Val        B
7898750 -0.011508519  6.747570 -6.443278 1.016754e-07 0.003385485 6.398649
8170468  0.013115884  6.012464  5.689956 1.196673e-06 0.013378383 3.937214
7969651 -0.010989325  7.051056 -5.687741 1.205368e-06 0.013378383 3.930008
8111415  0.008178560  2.540914  5.065649 9.095965e-06 0.060272779 1.924620
7895171  0.012638970  8.895633  4.980637 1.195878e-05 0.060272779 1.654193
7959012  0.012611550  4.428257  4.935664 1.381685e-05 0.060272779 1.511583
7972269 -0.009755820  4.360153 -4.919346 1.455932e-05 0.060272779 1.459918
8152664 -0.015744665  4.187651 -4.870722 1.701344e-05 0.060272779 1.306232
7896252 -0.008523726 11.280638 -4.834066 1.912948e-05 0.060272779 1.190642
7921533 -0.013989202  8.399406 -4.826789 1.957948e-05 0.060272779 1.167723
```

- Aunque alguno de los p-valores originales pudieran (marginalmente) considerarse como significativos cuando corregimos por comparaciones múltiples por el método de Benjamini-Hochberg no lo son.

Un diseño cruzado con limma

- Seguimos utilizando tamidata::gse44456.

```
pacman::p_load("limma")
data(gse44456,package="tamidata")
```

- Observamos los datos fenotípicos o metadatos.

```
head(pData(gse44456))
```

```
                                case gender age cirrhosis smoker postmortenint
GSM1085665_HE32H001.CEL      control   male  68        No     No         16.75
GSM1085666_HE32H003.CEL    alcoholic   male  51        No    Yes         27.00
GSM1085667_HE32H004.CEL      control   male  50        No     No         29.00
GSM1085668_HE32H005.CEL    alcoholic   male  50       Yes    Yes         24.00
GSM1085669_HE32H006_2_.CEL   control   male  56        No    Yes         24.00
GSM1085670_HE32H007_2_.CEL alcoholic   male  59        No     No         24.00
```

```
                            pH   batch
GSM1085665_HE32H001.CEL    6.59 Batch 1
GSM1085666_HE32H003.CEL    5.58 Batch 1
GSM1085667_HE32H004.CEL    6.68 Batch 2
GSM1085668_HE32H005.CEL    6.59 Batch 2
GSM1085669_HE32H006_2_.CEL 6.53 Batch 1
GSM1085670_HE32H007_2_.CEL 6.57 Batch 1
```

- Analizamos la posible dependencia con el alcoholismo recogida en la covariable `case`

```
alcoholism = pData(gse44456)[,"case"] # Simplificamos el código
design = model.matrix(~ 0 + alcoholism)
colnames(design) = c("control","alcoholic")
```

- Ajustamos el modelo.

```
fit = lmFit(gse44456,design)
```

```
(contrast.matrix = makeContrasts(dif = control - alcoholic,levels = design))
```

```
           Contrasts
Levels      dif
  control     1
  alcoholic  -1
```

- Estimamos.

```
fit2 = contrasts.fit(fit,contrast.matrix)
fit2 = eBayes(fit2)
```

- Veamos cuáles son significativos.

```
topTable(fit2,coef=1,adjust="BH")
```

```
            logFC  AveExpr         t      P.Value  adj.P.Val        B
7927186 -0.5424921 7.826424 -5.914411 5.718213e-07 0.01903993 5.029195
8125919 -1.1358866 8.313619 -5.628792 1.455678e-06 0.02236997 4.307423
8021081 -1.2885800 8.593822 -5.481851 2.351231e-06 0.02236997 3.935029
7961595  0.5322101 4.180459  5.440839 2.687326e-06 0.02236997 3.831026
7995838 -0.9825655 8.540374 -5.096932 8.199033e-06 0.04757580 2.959048
```

```
8130867  0.5858452 7.566787  5.083113 8.572989e-06 0.04757580 2.924055
8114814  0.3685955 7.694306  5.003841 1.106824e-05 0.05264847 2.723446
7922889  0.5172199 8.438433  4.946850 1.329382e-05 0.05533056 2.579390
8021741  0.7651623 9.183204  4.720399 2.741801e-05 0.10143750 2.008880
8099476  0.5231415 5.584413  4.658028 3.342454e-05 0.10201553 1.852423
```

- En el segunda análisis nos fijamos en dos covariables, el alcoholismo y el sexo de la persona.
- Es un diseño factorial 2×2.
- Veamos cuántos datos tenemos en cada celda.

```
table(pData(gse44456)[,"case"],pData(gse44456)[,"gender"])
```

```
          male female
control     13      6
alcoholic   14      6
```

- En un diseño como este las preguntas habituales son las siguientes:

  1. ¿Qué genes muestran un comportamiento diferenciado o relacionado con el alcoholismo?
  2. ¿Qué genes se comportan de un modo distinto para hombres y mujeres?
  3. ¿Para qué genes los cambios en su expresión según el alcoholismo son distintos en cada uno de los sexos?

- En jerga estadística hablaríamos de los efectos principales de los factores y de la posible interacción.
- Una aproximación simple y efectiva es construir un solo factor con todas las combinaciones de los factores.

```
casegender = vector("list",ncol(gse44456))
for(i in seq_along(casegender))
  casegender[[i]] = paste(pData(gse44456)[,"case"][i],
                          pData(gse44456)[,"gender"][i],sep="")
casegender = factor(unlist(casegender))
```

- Consideremos la siguiente matriz de diseño.

```
design = model.matrix(~  0 + casegender)
colnames(design) = levels(casegender)
```

- Podemos ver que cada coeficiente corresponde con la expresión media para la correspondiente combinación de factores.

```
fit = lmFit(gse44456,design)
```

- Construimos los contrastes en que estamos interesados.

```
cont.matrix = makeContrasts(
    dif1 = controlmale - alcoholicmale,
    dif2 = controlfemale - alcoholicfemale,
    dif12 = (controlmale - alcoholicmale)- (controlfemale - alcoholicfemale),
    levels = design)
fit2 = contrasts.fit(fit,cont.matrix)
fit2 = eBayes(fit2)
```

- Podemos ejecutar el siguiente código y podemos comprobar que ningún contraste es significativo.

```
topTable(fit2,coef=1,adjust="BH")
```

```
            logFC  AveExpr         t      P.Value adj.P.Val        B
7995838 -1.1908577 8.540374 -5.191480 6.771389e-06 0.1780083 2.970243
7961595  0.5846084 4.180459  4.904020 1.681620e-05 0.1780083 2.276111
7927186 -0.5500208 7.826424 -4.877869 1.825956e-05 0.1780083 2.213051
8125919 -1.1587440 8.313619 -4.789756 2.408412e-05 0.1780083 2.000783
8021081 -1.3637359 8.593822 -4.756501 2.673038e-05 0.1780083 1.920761
8173955 -0.9787221 6.510491 -4.588138 4.520781e-05 0.2275383 1.516644
7945371 -0.7678018 7.678374 -4.539905 5.251203e-05 0.2275383 1.401244
7920258 -0.3720446 7.330633 -4.433775 7.291269e-05 0.2275383 1.148043
7995806 -0.7041743 7.104240 -4.378750 8.637237e-05 0.2275383 1.017206
8074335 -0.5712542 7.856554 -4.349571 9.446980e-05 0.2275383 0.947956
```

```
topTable(fit2,coef=2,adjust="BH")
```

```
            logFC  AveExpr         t      P.Value adj.P.Val           B
7894184  1.3218613 5.170831  5.008754 1.208328e-05 0.3838704  0.93974694
7954243 -0.4035454 5.319421 -4.566467 4.835681e-05 0.3838704  0.17117825
7892534  1.0171512 4.687012  4.446053 7.020282e-05 0.3838704 -0.03840022
8117458  0.6684533 5.932542  4.441882 7.111213e-05 0.3838704 -0.04565679
7895726 -0.6515816 7.018058 -4.401748 8.047371e-05 0.3838704 -0.11546177
```

```
8099476  0.8486588 5.584413  4.377076 8.681816e-05 0.3838704 -0.15835720
8133728 -0.7453083 4.825997 -4.328946 1.006376e-04 0.3838704 -0.24198878
8028674 -0.3545895 7.832307 -4.295555 1.114686e-04 0.3838704 -0.29996701
8161507 -0.7139913 4.804273 -4.276255 1.182412e-04 0.3838704 -0.33346079
8130436 -0.6573848 3.958951 -4.249971 1.281156e-04 0.3838704 -0.37905015
```

```r
topTable(fit2,coef=3,adjust="BH")
```

```
            logFC  AveExpr         t      P.Value adj.P.Val         B
7895726  0.8494104 7.018058  4.773443 2.534820e-05 0.5735803 -2.519938
8027746 -1.3351585 4.740675 -4.486664 6.192538e-05 0.5735803 -2.701908
7893842 -0.9733824 5.149378 -4.418380 7.645580e-05 0.5735803 -2.745647
7895503  1.1804148 4.922862  4.338423 9.775600e-05 0.5735803 -2.797025
8162729 -0.5078698 6.114405 -4.312096 1.059678e-04 0.5735803 -2.813976
7893210 -1.3570688 6.241701 -4.306972 1.076427e-04 0.5735803 -2.817277
8050352 -1.0632392 7.273278 -4.258376 1.248731e-04 0.5735803 -2.848612
8117458 -0.7605210 5.932542 -4.204022 1.473500e-04 0.5735803 -2.883714
8009685 -0.6736454 8.596666 -4.169532 1.636148e-04 0.5735803 -2.906014
8002403 -0.8195714 9.355693 -4.147799 1.747505e-04 0.5735803 -2.920075
```