Expresión diferencial (marginal)

Guillermo Ayala Gallego

2025-05-21

Table of contents

El problema

¿De qué hablamos?

- ¿Hay diferencias de la expresión de genes entre dos (o más) grupos distintos?
- Dicho de otro modo, la expresión de un gen es distinta bajo distintos tratamientos.
- Se adopta la aproximación gen-a-gen, esto es, de momento solamente buscamos genes que se expresan diferencialmente sin atender al comportamiento conjunto que puedan tener.

Selección no específica o filtrado independiente

¿En qué consiste?

- La mayor parte de los genes no se expresan de un modo diferenciado entre condiciones.
- O ni se expresan en las distintas condiciones.
- Se puede realizar una preselección, un filtrado que no utilice información sobre las condiciones, utilizamos el perfil de expresión del gen pero no información sobre las muestras.
- A este filtrado se le denomina **filtrado independiente** o **selección no específica**.

Una posibilidad

- Consideramos el gen i-ésimo y tomamos una medida de localizacion como la mediana: u_i.
- Sea $q_{p_1}(u)$ sería el correspondiente percentil de orden p_1 de los valores u_i.
- Consideramos una medida de dispersión como el coeficiente intercuartílico: v_i.
- Sea $q_{p_2}(v)$ el correspondiente percentil de orden p_2 de los valores v_i.
- Nos quedamos con los genes tales que

$$\{i : i = 1, \dots, N; u_i \geq q_{p_1}(u); v_i \geq q_{p_2}(v)\},$$

- Lo lógico es usar bien **mediana y rango intercuartílico** o bien **media y desviación estándar**.
- Otra opción en lugar de localización: k-sobre-A: al menos k muestras tiene un nivel de expresión por encima de un valor mínimo A.

Datos ALL

- Los datos ALL son microarrays de 128 individuos distintos con leucemia linfoblástica aguda, ALL
- De estos individuos 95 corresponden a leucemia linfoblástica precursora aguda de células B y y 33 son leucemia linfoblástica precursora aguda de células T.
- Trabajaremos con las muestras de leucemia linfoblástica precursora aguda de células B.
- Los datos han sido preprocesados utilizando el método RMA.

Seleccionando algunas muestras en gse21942

- Es un `ExpressionSet`.

```
library(Biobase)
```

```
Cargando paquete requerido: BiocGenerics

Cargando paquete requerido: generics


Adjuntando el paquete: 'generics'

The following objects are masked from 'package:base':

    as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
    setequal, union


Adjuntando el paquete: 'BiocGenerics'

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, saveRDS, table, tapply, unique,
    unsplit, which.max, which.min
```

```
Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.
```

- Leemos los datos.

```
data(gse21942,package="tamidata")
```

Atendiendo a la variabilidad

- Calculamos rango intercuartílico para cada gen.

```
gse21942.iqr = apply(exprs(gse21942),1,IQR)
```

- Calculamos un percentil (por ejemplo, mediana) de los rangos intercuartílicos.

```
sel.iqr = (gse21942.iqr > quantile(gse21942.iqr,0.5))
```

- Calculamos la mediana para cada gen.

```
gse21942.median = apply(exprs(gse21942),1,median)
```

- Y la mediana de las medianas.

```
sel.median = (gse21942.median > quantile(gse21942.median,0.5))
```

- ¿Y cómo nos quedamos nos estos genes?

```
gse219421 = gse21942[sel.iqr & sel.median,]
```

k sobre A

- Consideremos el siguiente criterio: si el nivel de expresión mínimo es `c` y tenemos n muestras podemos pedir que un gen determinado se considere activo si en al menos k muestras del total de n su nivel de expresión supere este nivel mínimo de actividad.

- ¿Qué c?

```
quantile(exprs(gse21942))
```

```
       0%       25%       50%       75%      100%
 2.042694  3.721179  5.201626  7.128468 14.822599
```

```
c =  5.468801
```

- Determinamos qué niveles de expresión lo superan.

```
overc  = exprs(gse21942) > c
```

- Contamos el número de muestras que supera c.

```
count.c = apply(overc,1,sum)
```

- Y nos quedamos al menos 5 muestras superan c.

```
sel.c = count.c >= 5
```

- ¿Cuántos hay?

```
table(sel.c)
```

```
sel.c
FALSE  TRUE
10781 10577
```

kOverA

```
library(genefilter)
f1 = kOverA(5,5.468801)
ffun = filterfun(f1)
wh1 =  genefilter(exprs(gse21942), ffun)
```

nsFilter

- Supongamos que queremos aplicar la siguiente selección nos vamos a quedar con aquellas sondas tales que el rango intercuartílico (para todas las muestras) es mayor que la mediana de los rangos intercuartílicos.
- La mediana de la expresión es superior a la mediana de todas las medianas.
- Conocemos su anotación.

```
annotation(gse21942)
```

```
[1] "hgu133plus2"
```

- Cargamos la base de datos correspondiente.

```
library(hgu133plus2.db)
```

```
Cargando paquete requerido: AnnotationDbi
```

```
Cargando paquete requerido: stats4
```

```
Cargando paquete requerido: IRanges
```

```
Cargando paquete requerido: S4Vectors
```

```

Adjuntando el paquete: 'S4Vectors'
```

```
The following object is masked from 'package:utils':

    findMatches
```

```
The following objects are masked from 'package:base':

    expand.grid, I, unname
```

```
Cargando paquete requerido: org.Hs.eg.db
```

- Y filtramos.

```
gse21942.filt1 = nsFilter(gse21942,var.func=IQR,var.cutoff=0.5,
                          require.GOBP=TRUE)
gse21942_1 = gse21942.filt1$eset
```

- Tomamos la mediana por gen y nos quedamos con los genes cuya mediana sea mayor que la mediana de las medianas.

```
gse21942.filt2 = nsFilter(gse21942,var.func=median,var.cutoff=0.5,
  require.GOBP=TRUE)
gse21942_2 = gse21942.filt2$eset
```

- Consideramos simultáneamente las dos condiciones.

```
sel = intersect(featureNames(gse21942_1),featureNames(gse21942_2))
gse21942.sel = gse21942[sel,]
```

Log fold-change

El logaritmo del cociente de las medias

- Queremos comparar dos grupos.
- x_{ij} (y_{ij}) es la expresión de la i-ésima característica en la j-ésima muestra del primer grupo (segundo grupo).
- Tomamos los logaritmos (en base 2 o `log2`) de los valores originales:

$$u_{ij} = \log_2(x_{ij}),\ v_{ij} = \log_2(y_{ij}).$$

- Para cada gen, tendremos las expresiones medias para la i-ésima característica

$$\bar{x}_{i\cdot} = \sum_{j=1}^{n_1} \frac{x_{ij}}{n_1}$$

  $\bar{y}_{i\cdot}$, $\bar{u}_{i\cdot}$, $\bar{v}_{i\cdot}$.
- ¿Qué se entiende por **fold-change**?

- Dos son las interpretaciones **distintas** de este valor.

$$FC_i^{(1)} = \frac{\bar{x}_{i\cdot}}{\bar{y}_{i\cdot}}.$$

y

$$FC_i^{(2)} = \bar{u}_{i\cdot} - \bar{v}_{i\cdot},$$

- El log-fold change es el logaritmo en base 2 de los fold-change que acabamos de definir.
- Si el cociente anterior es mayor que `c` entonces diríamos que el gen se expresa de un modo diferencial en ambos grupos.
- Tenemos una expresión diferencial del gen i si

$$\left|\log_2\left(FC_i\right)\right| \geq c$$

Expresión diferencial marginal

Dos grupos

- Utilizamos los datos gse21942.

```
data(gse21942,package="tamidata")
```

- Nos fijamos en un gen.

```
probeRow = 1345
fData(gse21942)[probeRow,]
```

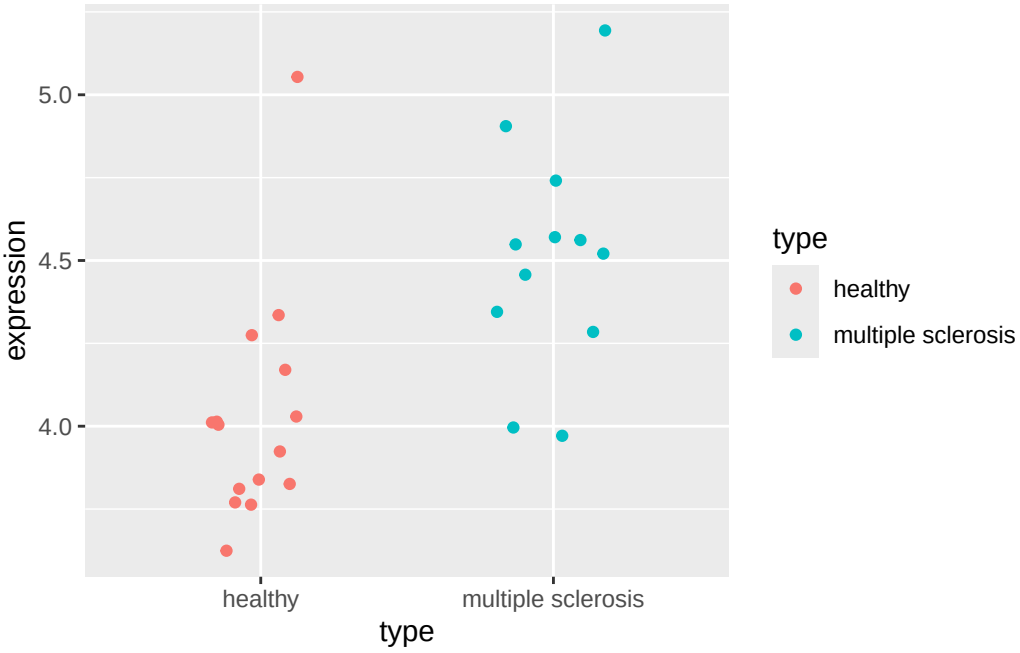
```
          PROBEID ENTREZID         ENSEMBL
1938 1554564_a_at   121665 ENSG00000157837
```

- Guardamos niveles de expresión.

```
x0 = pData(gse21942)[,"FactorValue..DISEASE.STATE."]
y0 = exprs(gse21942)[probeRow,]
```

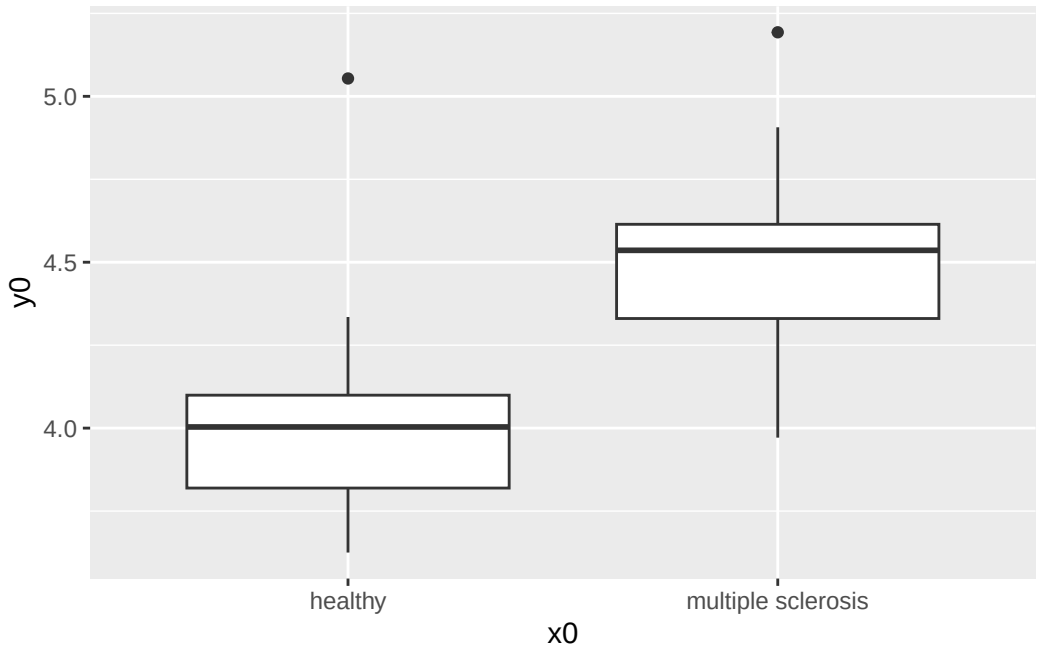
Perfil de expresión

```
pacman::p_load(ggplot2)
df=data.frame(id = 1:ncol(gse21942),expression = y0,type = x0)
ggplot(df,aes(x=type,y=expression,color=type)) +
    geom_jitter(position=position_jitter(0.2))
```



Un diagrama de cajas

```
df = data.frame(id=1:length(x0),x0,y0)
ggplot(df,aes(x=x0,y=y0)) + geom_boxplot()
```



Comparamos dos condiciones con `t.test`

- Tenemos datos relativos a niveles de expresión bajo dos condiciones experimentales.
- Denotamos por Y_1 el nivel de expresión aleatorio que observamos bajo la primera condición y por Y_2 lo mismo pero con la segunda condición.
- $Y_1 \sim N(\mu_1, \sigma^2)$
- $Y_2 \sim N(\mu_2, \sigma^2)$
- Nos planteamos el contraste

$$H_0 : \mu_1 = \mu_2,$$
$$H_1 : \mu_1 \neq \mu_2.$$

- Calculamos

$$T = \frac{\bar{Y}_1 - \bar{Y}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

con

$$S_p = \hat{\sigma}^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

siendo

$$S_1^2 = \sum_{i=1}^{n_1} \frac{(Y_{1i} - \bar{Y}_1)^2}{n_1 - 1}.$$

y S_2^2 lo mismo sustituyendo Y_{1i} por Y_{2i}.

- Rechazamos la hipótesis nula cuando

$$|T| > t_{\nu,1-\alpha/2}$$

- El p-valor viene dado por

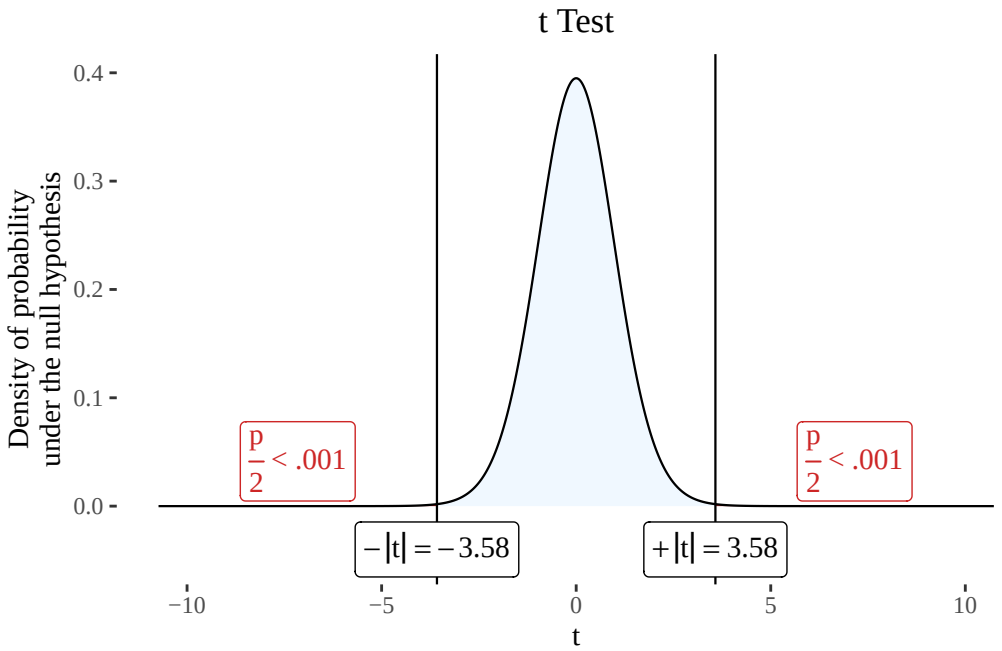
$$p = P(|T| \geq t_0)$$

siendo t_0 el valor observado de T en cada caso.

```
t.test(y0 ~ x0,var.equal=TRUE)
```

```
    Two Sample t-test

data:  y0 by x0
t = -3.5777, df = 25, p-value = 0.001452
alternative hypothesis: true difference in means between group healthy and group multiple scl
95 percent confidence interval:
 -0.7543060 -0.2031418
sample estimates:
           mean in group healthy mean in group multiple sclerosis
                        4.029890                         4.508614
```



- Comparación simultánea de todos los genes

Hacemos un t-test para todos los genes

```
library(genefilter)
tt = genefilter::rowttests(gse21942,
                           pData(gse21942)[,"FactorValue..DISEASE.STATE."])
```

```
head(tt)
```

```
           statistic          dm     p.value
1007_s_at -3.0779491 -0.20712393 0.005003033
1053_at    3.3695549  0.21636794 0.002445187
117_at     0.2727579  0.03740568 0.787279726
121_at    -0.5428162 -0.02430009 0.592063311
1255_g_at -1.1845653 -0.05402389 0.247329067
1294_at   -1.2452406 -0.09874406 0.224589177
```

- ¿Cuándo tests mostrarían una expresión diferencial entre ambos grupos?

```
table(tt$p.value < 0.05)
```

```
FALSE  TRUE
15011  6347
```

- Como vemos tenemos demasiados genes que muestran expresión diferencial.
- Y no parece muy razonable esto.
- El nivel de significación α es una cota superior del error tipo I cuando realizamos **un solo test**.
- Pero no estamos aplicando un solo test.
- Estamos rechazando muchas hipótesis nulas (no diferencia entre grupos para cada gen) cuando no deberíamos de hacerlo.

Comparamos dos condiciones con test de Kolmogorov-Smirnov

- Aplicamos el test.

```
(ks1345 = ks.test(exprs(gse21942)[1345,]~
                    pData(gse21942)[,"FactorValue..DISEASE.STATE."]))
```

```
    Exact two-sample Kolmogorov-Smirnov test

data:  exprs(gse21942)[1345, ] by pData(gse21942)[, "FactorValue..DISEASE.STATE."]
D = 0.7, p-value = 0.001305
alternative hypothesis: two-sided
```

- Para todas las filas de la matriz de expresión.

```
ks.gse21942 = sapply(1:nrow(gse21942),function(i)
    ks.test(exprs(gse21942)[i,]~
                pData(gse21942)[,"FactorValue..DISEASE.STATE."])$p.value)
```

¿Cuántas sondas son significativas?

```
table(ks.gse21942 <= 0.05)
```

```
FALSE  TRUE
15916  5442
```

Comparamos dos condiciones con un test de permutación de Fisher-Pitman

```
pacman::p_load(coin)
(per1345 = coin::oneway_test(exprs(gse21942)[1345,]~
                                pData(gse21942)[,"FactorValue..DISEASE.STATE."]))
```

```
    Asymptotic Two-Sample Fisher-Pitman Permutation Test

data:  exprs(gse21942)[1345, ] by
     pData(gse21942)[, "FactorValue..DISEASE.STATE."] (healthy, multiple sclerosis)
Z = -2.9672, p-value = 0.003005
alternative hypothesis: true mu is not equal to 0
```

Otra vez podemos repetir el análisis para todas las filas.

```
per.gse21942 = sapply(1:nrow(gse21942),function(i)
    pvalue(coin::oneway_test(exprs(gse21942)[i,]~
                pData(gse21942)[,"FactorValue..DISEASE.STATE."])))
```

¿Cuántos son significativos?

```
table(per.gse21942<.05)
```

```
FALSE  TRUE
15118  6240
```