

Expresión diferencial con datos RNA-seq

Guillermo Ayala Gallego

5/30/23

Table of contents

Introducción	2
Paquetes	2
Data	2
Una muestra por condición	2
Un test para comparar proporciones	2
edgeR::binomTest	2
edgeR clásico	4
Introducción	4
Estimación de una dispersión común	4
qCML: Quantile-adjusted conditional maximum likelihood	6
Un test exacto	7
Contraste de hipótesis	7
Dispersiones posiblemente distintas	8
Un análisis con edgeR	9
DGEList	9
Eliminando genes con conteos bajos	11
edgeR utilizando modelo lineal generalizado	13
Modelo	13
TCGA-COAD	14
Bibliografía	18

Introducción

Paquetes

```
pacman::p_load(SummarizedExperiment,edgeR,ggplot2)
```

Data

```
data(PRJNA297664,package="tamidata")
```

```
colData(PRJNA297664)[,"treatment"]
```

```
[1] Wild          Wild          SEC66 deletion Wild          SEC66 deletion  
[6] SEC66 deletion  
Levels: Wild SEC66 deletion
```

Una muestra por condición

Un test para comparar proporciones

- Queremos comparar dos condiciones y disponemos de una muestra en cada una de las condiciones.
- En este caso para cada gen tendríamos una tabla 2×2 como

Muestra	Gen	Resto	Total
1	y_1	$m_1 - y_1$	m_1
2	y_2	$m_2 - y_2$	m_2
Total	$y_1 + y_2$	$m_1 + m_2 - y_1 - y_2$	$m_1 + m_2$

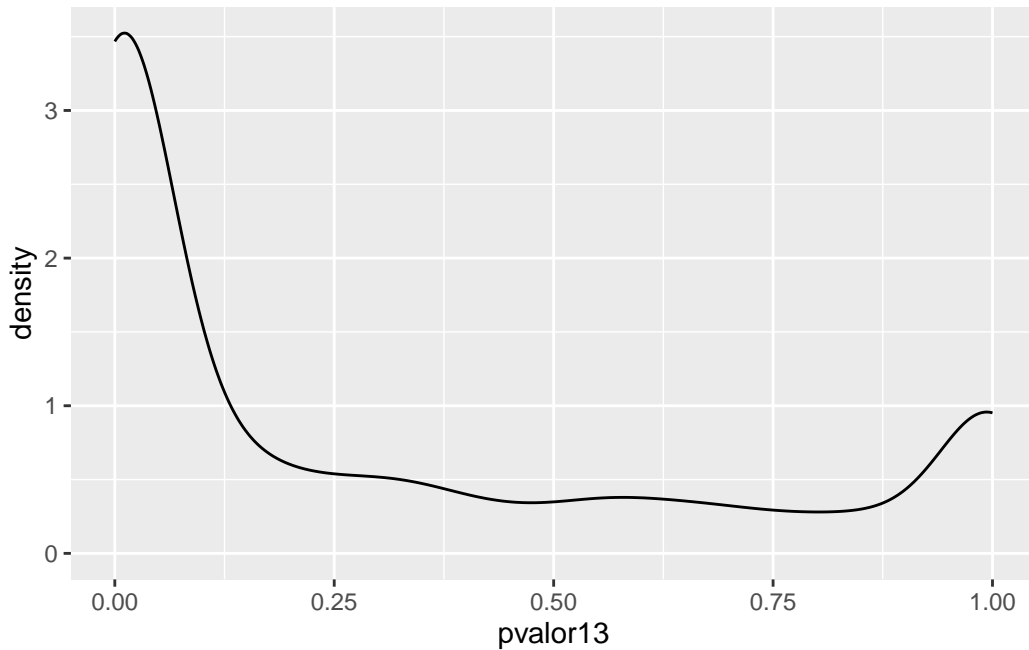
edgeR::binomTest

- Para el gen i -ésimo tendríamos los conteos x_{i1} e x_{i2} en las dos muestras siendo m_1 y m_2 los tamaños de las librerías.
- Suponemos fijo el número total de muestras para el gen i ($n_i = x_{i1} + x_{i2}$) y tamaño total de las dos librerías m_1 y m_2

- Bajo estas dos hipótesis previas vamos a contrastar que $H_i : p_1 = m_1/(m_1 + m_2)$ frente a $H_i : p_1 \neq m_1/(m_1 + m_2)$
- Como ilustración nos fijamos en las muestras 1 y 3.

```
pvalor13 = edgeR::binomTest(assay(PRJNA297664)[,1], assay(PRJNA297664)[,3])
```

```
pacman::p_load(ggplot2)
df = data.frame(pvalor13)
ggplot2::ggplot(df, aes(x=pvalor13)) + geom_density()
```

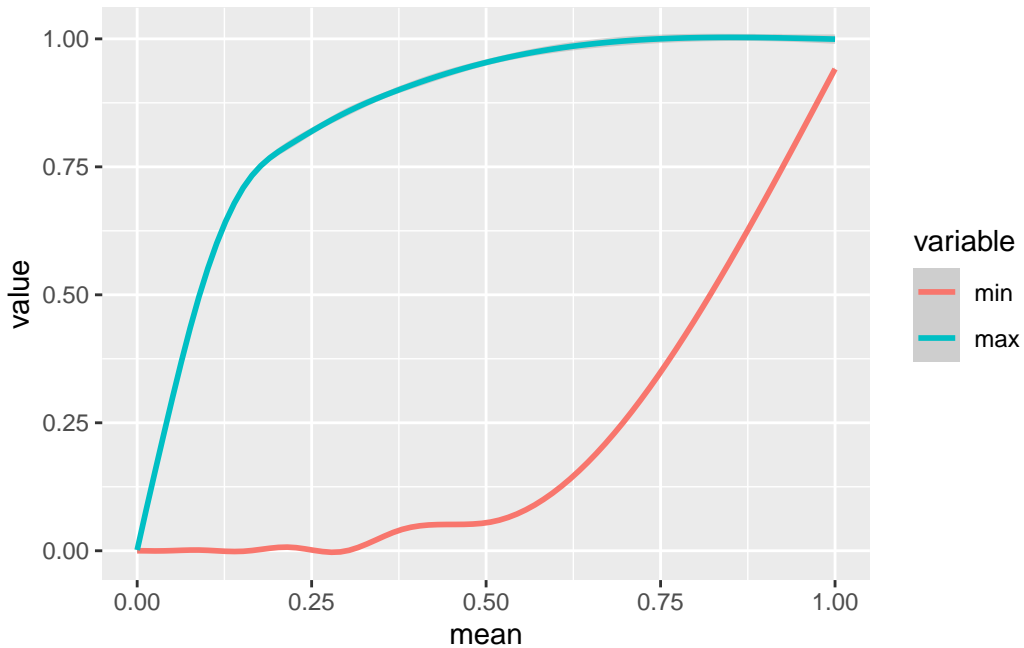


```
pares = rbind(c(1,3),c(1,5),c(1,6),c(2,3),c(2,5),c(2,6),c(4,3),
              c(4,5),c(4,6))
pvalores = apply(pares,1,
                 function(x) binomTest(assay(PRJNA297664)[,x[1]],
                                       assay(PRJNA297664)[,x[2]]))
```

```
y = apply(pvalores,1,function(x) c(mean(x),min(x),max(x)))
df = data.frame(t(y))
names(df) = c("mean","min","max")
df=df[sort(df[,"mean"],index.return=TRUE)$ix,]
```

```
df1 = reshape2::melt(df, id="mean")
pp = ggplot(df1, aes(x=mean, y=value, color=variable))
pp + geom_smooth()
```

``geom_smooth()`` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'



edgeR clásico

Introducción

- Para dos grupos.
- Es un test exacto.
- Propuesto en Robinson and Smyth (2008) y Robinson and Smyth (2007) (en este orden).

Estimación de una dispersión común

- Consideramos una característica en n muestras (o librerías) de tamaños distintos.
- Sea m_i el tamaño de la i -ésima librería (total de lecturas).

- Sea λ la proporción que hay en una librería cualquiera de la característica en que estamos interesados.
- Si $Y \sim NB(\mu, \phi)$ entonces la función de probabilidad viene dada por

$$f(y|\mu, \phi) = P(Y = y|\mu, \phi) = \frac{\Gamma(y + \phi^{-1})}{\Gamma(\phi^{-1})\Gamma(y + 1)} \left(\frac{1}{1 + \mu\phi}\right)^{\phi^{-1}} \left(\frac{\mu}{\phi^{-1} + \mu}\right)^y.$$

- Se puede comprobar que:
 - su media es $E(Y) = \mu$,
 - su varianza es $var(Y) = \mu + \phi\mu^2$.
- **Asumimos** que Y_i tiene una distribución binomial negativa con media

$$\mu_i = m_i\lambda$$

y con dispersión ϕ .

- Supongamos un caso ideal (improbable): todas las librerías tienen el mismo tamaño: $m_i = m$ para $i = 1, \dots, n$.
- Bajo esta condición se verifica que $Z = \sum_{i=1}^n Y_i \sim NB(nm\lambda, \phi/n)$.
- Además la logverosimilitud condicionada al valor de $Z = z$ no depende de λ y podemos estimar el valor de ϕ .
- Se estimaría ϕ maximizando la logverosimilitud condicionada (a la suma de todos los conteos del gen) dada por

$$l_{\mathbf{y}|z} = \left[\sum_{i=1}^n \log \Gamma(y_i + 1/\phi) \right] + \log \Gamma(n/\phi) - \log \Gamma(z + n/\phi) - n \log \Gamma(1/\phi),$$

donde $\mathbf{y} = (y_1, \dots, y_n)'$ son los conteos observados.

- En el caso en que los tamaños de las librerías son distintos la verosimilitud no tiene una expresión simple.

- La idea consiste en modificar los valores observados y_i y generar unos datos **nuevos** en que los tamaños de las librerías **sí** que coincidan: los **pseudodatos**.
- El tamaño común de las librerías será la media geométrica de los tamaños observados:

$$m^* = \left(\prod_{i=1}^n m_i \right)^{\frac{1}{n}}.$$

- ¿Y cómo transformamos los conteos y_i ?

qCML: Quantile-adjusted conditional maximum likelihood

1. Inicializamos ϕ (por ejemplo, con el estimador máximo verosímil condicionado sin realizar ningún ajuste).
2. Dado el valor estimado de ϕ , estimamos λ maximizando la verosimilitud para el valor dado de la dispersión.
3. Suponemos que cada conteo y_i es un valor observado de una distribución binomial negativa con media $m_i\lambda$ y parámetro de dispersión ϕ . Calculamos los percentiles

$$p_i = P(Y < y_i | m_i\lambda, \phi) + \frac{1}{2}P(Y = y_i | m_i\lambda, \phi),$$

para $i = 1, \dots, n$.

4. Suponemos ahora una distribución binomial negativa con media $m^*\lambda$ y dispersión ϕ .
 1. Determinamos qué valor sería el percentil de orden p_i en la nueva distribución. Ya no tiene porqué ser un dato entero e incluso puede ser negativo.
 2. Los pseudodatos para un mismo gen tienen aproximadamente la misma distribución.
5. Estimamos la dispersión ϕ con los pseudodatos utilizando la verosimilitud condicionada a la (pseudo) suma total de un gen.
6. Repetimos desde 2 hasta 5 hasta que converja ϕ .

Un test exacto

- Tenemos dos condiciones a comparar.
- Consideramos el gen i y denotamos por
 - y_{ijk} el conteo para la muestra k en la condición j del gen i donde $j = 1, 2$ y $k = 1, \dots, n_j$.
- Los tamaños de la librería k de la condición j será m_{jk} .
- Los tamaños de las librerías **no** son iguales dentro de cada clase.
- Aplicamos el procedimiento qCML dentro de cada clase.
- Tendremos un tamaño común m_j en la condición j .
- Estimamos ϕ maximizando

$$l(\phi) = \sum_{i=1}^N l_i(\phi).$$

con

$$l_i(\phi) = \sum_{j=1}^2 \left(\sum_{k=1}^{n_j} \log \Gamma(y_{ijk} + \phi^{-1}) + \log \Gamma(n_j \phi^{-1}) - \log \Gamma(z_{ij} + n_j \phi^{-1}) - n_j \log \Gamma(\phi^{-1}) \right).$$

siendo $z_{ij} = y_{ij} = \sum_{k=1}^{n_j} y_{ijk}$.

- El estimador de ϕ será $\hat{\phi}_C$.

Contraste de hipótesis

- Estamos asumiendo

$$EY_{ijk} = m_{jk} \lambda_{ij},$$

- La hipótesis nula de que no hay diferencias entre las medias de los conteos en las dos condiciones para el i -ésimo gen, se formularía como

$$H_i : \lambda_{i1} = \lambda_{i2},$$

$$K_i : \lambda_{i1} \neq \lambda_{i2}.$$

- Bajo la hipótesis nula H_i no tendríamos diferencia en el valor de λ entre las condiciones y sería un valor común λ_i .
- Aplicamos método qCML a **todas** las muestras: y_{ijk} son los pseudodatos.
- Utilizando el estimador $\hat{\phi}_C$ y los conteos y_{ijk} podemos estimar λ_i .
- Bajo la hipótesis nula de no diferencia entre grupos tendríamos que $Y_{ij\cdot} = \sum_{k=1}^{n_j} Y_{ijk} \sim NB(n_j m^* \hat{\lambda}_i, \hat{\phi}_C / n_j)$.
- $Y_{i1\cdot}$ e $Y_{i2\cdot}$ son independientes.
- La suma $Y_{i1\cdot} + Y_{i2\cdot}$ también tiene una distribución binomial negativa: $Y_{i1\cdot} + Y_{i2\cdot} = \sum_{i=1}^2 \sum_{k=1}^{n_j} Y_{ijk} \sim NB((n_1 + n_2) m^* \hat{\lambda}_i, \hat{\phi}_C / (n_1 + n_2))$
- Podemos considerar la distribución condicionada del vector aleatorio $(Y_{i1\cdot}, Y_{i2\cdot})$ a la suma $Y_{i1\cdot} + Y_{i2\cdot}$ y considerar las probabilidades de los conteos conjuntos que **son menos probables que el observado**.
- La suma de estas probabilidades nos daría el p-valor del test.

Dispersiones posiblemente distintas

- Una misma dispersión sobre una cantidad grande de genes no es muy razonable.
- Buena desde el punto de vista del modelo y su estimación pero nos aleja de los datos.
- Supongamos dispersiones posiblemente distintas para los distintos genes, ϕ_i .
- Proponen

$$WL(\phi_i) = l_i(\phi_i) + \alpha l(\phi_i)$$

siendo α el peso que se da a la verosimilitud global.

- Es una función que propone un compromiso entre considerar l como función a maximizar considerando la misma contribución a todos los genes y l_i en donde solamente consideramos los conteos del propio gen.
- En Robinson & Smyth (2008) proponen un método de estimación de α .

Un análisis con edgeR

DGEList

- Tenemos un SummarizedExperiment.
- Construimos un DGEList.

```
x = DGEList(counts = assay(PRJNA297664),  
            group = colData(PRJNA297664)[,"treatment"])
```

- La clase del nuevo objeto es

```
class(x)
```

```
[1] "DGEList"  
attr(,"package")  
[1] "edgeR"
```

Y sus atributos son

```
attributes(x)
```

```
$class  
[1] "DGEList"  
attr(,"package")  
[1] "edgeR"
```

```
$names  
[1] "counts" "samples"
```

La matriz de conteos

```
x$counts
```

Tenemos la componente `samples`.

```
class(x$samples)
```

```
[1] "data.frame"
```

Las primeras filas son

```
head(x$samples)
```

```
      group lib.size norm.factors
Sample1   Wild  4788536           1
Sample2   Wild  9387986           1
Sample3 SEC66 deletion 9599910           1
Sample4   Wild  8896028           1
Sample5 SEC66 deletion 9003755           1
Sample6 SEC66 deletion 9002105           1
```

Tenemos un factor.

```
x$samples[,"group"]
```

```
[1] Wild           Wild           SEC66 deletion Wild           SEC66 deletion
[6] SEC66 deletion
Levels: Wild SEC66 deletion
```

Los tamaños de las librerías son

```
x$samples[,"lib.size"]
```

```
[1] 4788536 9387986 9599910 8896028 9003755 9002105
```

Los factores de normalización son

```
x$samples[,"norm.factors"]
```

```
[1] 1 1 1 1 1 1
```

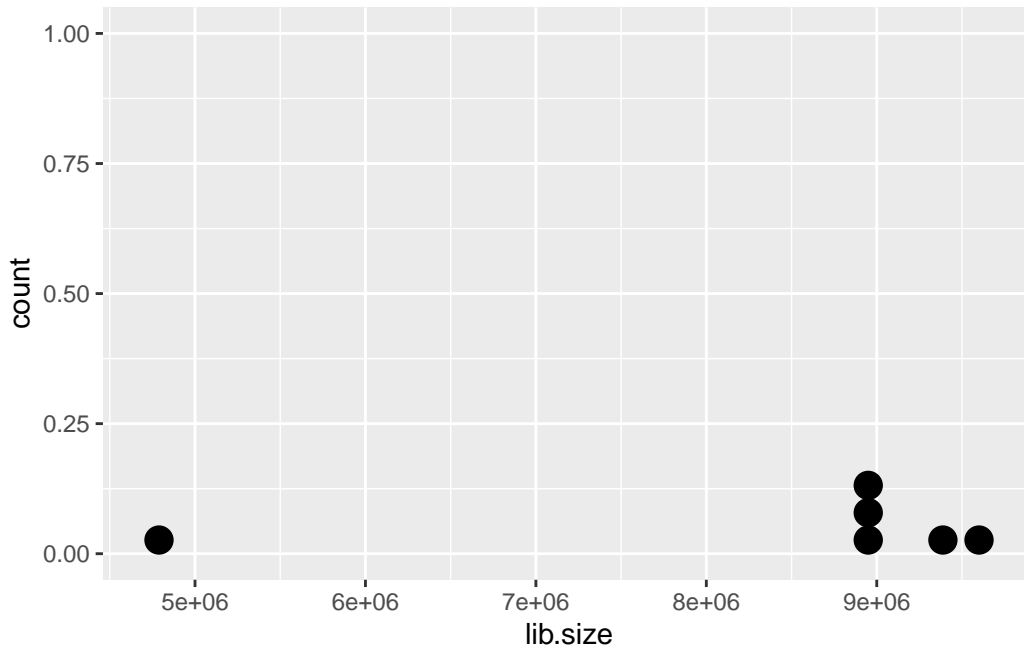
Veamos una descripción de los tamaños de las librerías.

```
summary(x$samples[,"lib.size"])
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
4788536 8922547 9002930 8446387 9291928 9599910
```

```
ggplot(x$samples,aes(x= lib.size)) + geom_dotplot()
```

Bin width defaults to 1/30 of the range of the data. Pick better value with ``binwidth``.



Eliminando genes con conteos bajos

Utilizamos los conteos por millón CPM en lugar de los conteos originales. Obligamos a que se recalculen los tamaños de las librerías.

```
keep = rowSums(cpm(x)>1) >= 2  
x <- x[keep, , keep.lib.sizes=FALSE]
```

```
dge = DGEList(counts=assay(PRJNA297664),  
              group=colData(PRJNA297664)[,"treatment"])  
dge.c = estimateCommonDisp(dge) ##Estimamos dispersión común  
dge.c$common.dispersion
```

```
[1] 0.01170892
```

```
dge.t = estimateTagwiseDisp(dge.c) ##Dispersiones por gen
et.c = exactTest(dge.c)
et.t = exactTest(dge.t)
```

Y vemos los resultados.

```
topTags(et.c)
```

```
Comparison of groups: SEC66 deletion-Wild
      logFC  logCPM      PValue      FDR
YBR171W -10.150220 6.114922 2.128721e-258 1.516926e-254
YCR021C  -1.928779 8.463733 4.873506e-47 1.736430e-43
YBR054W  -1.878579 7.156683 3.518798e-43 8.358319e-40
YGL255W  -1.846348 7.518466 1.930761e-42 3.439650e-39
YNR034W-A -2.176676 4.670571 1.175800e-40 1.675750e-37
YBR093C  -1.694281 8.563323 2.731791e-37 3.244457e-34
YFR053C   1.621598 6.384697 2.625927e-31 2.673194e-28
YER150W  -1.560027 5.479642 2.325258e-26 2.071224e-23
YDR171W  -1.383170 7.792428 2.142035e-25 1.696016e-22
YDR214W   1.362108 7.982939 1.024220e-24 7.298590e-22
```

```
topTags(et.t)
```

```
Comparison of groups: SEC66 deletion-Wild
      logFC  logCPM      PValue      FDR
YBR171W -10.150503 6.114922 5.636512e-296 4.016579e-292
YGL255W  -1.846051 7.518466 5.459134e-30 1.945090e-26
YBR093C  -1.694086 8.563323 5.843701e-27 1.388074e-23
YNR034W-A -2.174452 4.670571 7.411466e-22 1.320353e-18
YDR214W   1.362104 7.982939 2.334672e-20 3.327375e-17
YLR109W   1.240894 9.510958 4.583075e-20 5.443166e-17
YHR215W  -1.208860 9.533270 2.172522e-18 2.211627e-15
YAR071W  -1.211709 9.219219 3.764430e-18 3.353166e-15
YMR186W   1.109256 11.622933 9.483967e-18 7.509195e-15
YKL161C   1.100257 5.723199 1.070358e-16 7.627369e-14
```

edgeR utilizando modelo lineal generalizado

Modelo

- Y_{ij} el conteo aleatorio (número de lecturas alineadas) para el gen i en la muestra j .
- Denotamos por $m_j = \sum_{i=1}^N y_{.j}$ la profundidad de secuenciación o total de lecturas de la muestra j .
- Utilizamos como función de enlace el logaritmo natural.
- Consideramos la profundidad de secuenciación como offset (un modelo de tasas sobre la profundidad de secuenciación).
- El modelo para la media es

$$\ln \mu_{ij} = \mathbf{x}_j^T \boldsymbol{\beta}_i + \ln m_j.$$

- En el modelo las variables predictoras son comunes a todos los genes.
- Asumimos que la componente aleatoria sigue una distribución binomial negativa (con el parámetro de dispersión conocido)
- Entonces

$$\text{var}(Y_{ijk}) = \mu_{ij} + \phi_i \mu_{ij}^2,$$

siendo ϕ_i el parámetro de dispersión que hemos de asumir conocido o, de otro modo, tenemos que estimarlo previamente.

- En McCarthy, Chen, and Smyth (2012) muestran cómo estimar por máxima verosimilitud el vector de coeficientes $\boldsymbol{\beta}_i$.
- Utilizan una modificación de los mínimos cuadrados iterativamente reponderados (IR-WLS).
- El parámetro de dispersión se estima maximizando la logverosimilitud penalizada definida como

$$APL_i(\phi_i) = \ell(\phi_i; \mathbf{y}_i, \hat{\boldsymbol{\beta}}_i) - \frac{1}{2} \ln |\mathbb{I}_i|$$

siendo:

- \mathbf{y}_i los conteos para el gen i ,
- $\hat{\boldsymbol{\beta}}_i$ el vector de coeficientes,
- $\ell()$ es la función de logverosimilitud
- $|\mathbb{I}_i|$ el determinante de la matriz de información de Fisher para el i -ésimo gen.

TCGA-COAD

```
pacman::p_load(edgeR, SummarizedExperiment)
load(paste0(dirTamiData, "tcga_coad.rda"))
```

- Nos centramos en las variables fenotípicas `age_at_diagnosis` y `tissue_or_organ_of_origin`.

```
summary(colData(tcga_coad)[,"age_at_diagnosis"])
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
11391  20604   24896   24146  28204   32872     4
```

```
table(colData(tcga_coad)[,"tissue_or_organ_of_origin"])
```

Ascending colon	Cecum	Colon, NOS
72	70	59
Descending colon	Hepatic flexure of colon	Rectosigmoid junction
15	12	3
Sigmoid colon	Splenic flexure of colon	Transverse colon
76	6	13

- Hemos de eliminar aquellas muestras que tienen las variables predictoras con datos faltantes ya que las funciones que siguen no los admiten.

```
torm1 = which(is.na(colData(tcga_coad)$"age_at_diagnosis"))
torm2 = which(is.na(colData(tcga_coad)$ "tissue_or_organ_of_origin"))
toremove = union(torm1,torm2)
tcga_coad = tcga_coad[,-toremove]
```

- Construimos el objeto `DGEList` sin indicar ninguna variable `group` ni ninguna matriz de modelo y eliminamos genes con conteos bajos.

```
dge = DGEList(counts=assay(tcga_coad))
to_keep = rowSums(cpm(dge) > 0.5) > 20
dge = dge[to_keep,keep.lib.sizes=FALSE]
dim(dge)
```

```
[1] 16158  324
```

- Construimos la matriz de modelo con las dos variables predictoras, una de carácter categórico y la otra numérica.
- Cambiamos los nombres de las columnas de la matriz de modelo.

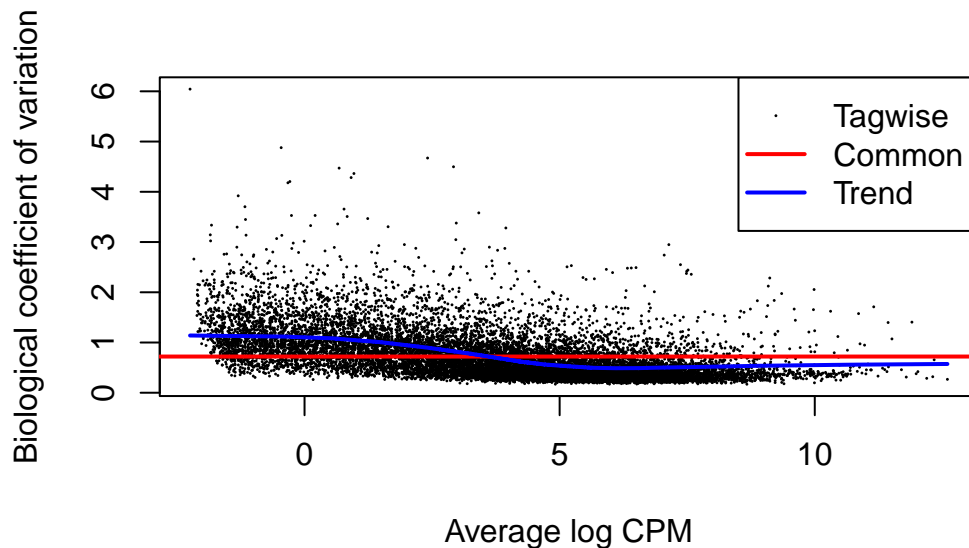
```
design0 = model.matrix(~ 0 +
  colData(tcga_coad)$"tissue_or_organ_of_origin"
  + colData(tcga_coad)$"age_at_diagnosis")
y = levels(colData(tcga_coad)$"tissue_or_organ_of_origin")
y = sapply(y,function(x) gsub(" ","_",x)) ## Eliminamos espacios
y = sapply(y,function(x) gsub(",","_",x)) ## Eliminamos las comas
colnames(design0) = c(y,"age_at_diagnosis")
```

- Estimamos las dispersiones por tres métodos distintos:
 - Asumiendo una dispersión común,
 - una por gen y
 - con una relación media-varianza.

```
dge = estimateDisp(dge,design=design0)
```

Si solo queremos una de las tres opciones podemos usar las funciones `estimateGLMCommonDisp()`, `estimateGLMTagwiseDisp()` y `estimateGLMTrendedDisp()`.

```
plotBCV(dge)
```



```
fit = glmFit(dge, design=design0)
```

- Veamos si influye la variable `age_at_diagnosis`.
- Si observamos la matriz de modelo `design0` corresponde con la columna 10 de la matriz de modelo.
- Se realiza un test del cociente de verosimilitudes.

```
lrt1 = glmLRT(fit,coef="age_at_diagnosis")  
lrt1 = glmLRT(fit,coef=10) ## Equivalente a la línea anterior  
topTags(lrt1)
```

```
Coefficient: age_at_diagnosis  
              logFC      logCPM      LR      PValue      FDR  
UGT2B10 -0.0002335397  0.07018047  68.64932  1.176251e-16  1.753945e-12  
KCNH3   -0.0001438677  -0.73269543  67.44089  2.170993e-16  1.753945e-12  
CPS1    -0.0002306034  4.32352487  60.85641  6.139324e-15  3.306640e-11  
SULT1E1 -0.0002374397   0.83203192  57.37333  3.604616e-14  1.456085e-10  
GPR64   -0.0001654538   0.52261384  55.61329  8.822524e-14  2.851087e-10  
UPK1A   -0.0002044708  -0.94768507  53.99073  2.014376e-13  5.424715e-10  
KRT81   -0.0001417146  -0.31911518  50.55783  1.157049e-12  2.670799e-09  
DLX5    -0.0001507914  -0.47209136  45.87384  1.261190e-11  2.547288e-08  
EPHX3   -0.0001195753  -0.07584401  45.21348  1.766851e-11  2.897626e-08  
CACNA1I -0.0001300370  -0.65221580  45.18437  1.793308e-11  2.897626e-08
```

- Podemos evaluar toda la variable `tissue_or_organ_of_origin`.

```
lrt2 = glmLRT(fit,coef=1:9)  
topTags(lrt2)
```

```
Coefficient: Ascending_colon Cecum Colon_NOS Descending_colon Hepatic_flexure_of_colon Rec  
              logFC.Ascending_colon logFC.Cecum logFC.Colon_NOS  
RBM44          -23.03403    -22.79346        -23.20739  
LPAL2          -22.83425    -22.63042        -22.87998  
C6orf52        -22.69677    -22.74012        -22.53655  
SLC5A10        -22.59678    -22.67309        -22.35748  
APOBEC3H       -22.53489    -22.57474        -22.19753  
LINC00574      -22.53141    -22.30988        -21.93773  
ATOX7          -22.50255    -22.51006        -22.89692  
GRAPL          -22.47544    -21.90377        -21.80912  
C6orf201       -22.45426    -22.61184        -22.56993  
RPL23AP64      -22.43706    -22.48745        -22.71604
```


	logFC.Descending_colon	logFC.Hepatic_flexure_of_colon		
RBM44	-22.81666	-22.98087		
LPAL2	-22.60164	-22.63563		
C6orf52	-23.72080	-22.34341		
SLC5A10	-22.81269	-22.55916		
APOBEC3H	-22.89885	-22.24700		
LINC00574	-22.53327	-21.59098		
ATOH7	-22.36426	-21.88149		
GRAPL	-22.20629	-22.98721		
C6orf201	-22.39796	-22.18401		
RPL23AP64	-22.31727	-22.72084		
	logFC.Rectosigmoid_junction	logFC.Sigmoid_colon		
RBM44	-23.39152	-22.83901		
LPAL2	-23.04283	-22.15825		
C6orf52	-22.72652	-22.77770		
SLC5A10	-22.73793	-22.61990		
APOBEC3H	-23.32153	-22.97932		
LINC00574	-22.95598	-22.59596		
ATOH7	-22.15867	-21.84132		
GRAPL	-23.39572	-22.33813		
C6orf201	-23.12831	-22.58081		
RPL23AP64	-22.02019	-22.25008		
	logFC.Splenic_flexure_of_colon	logFC.Transverse_colon	logCPM	
RBM44	-23.79191	-19.16905	-1.002324	
LPAL2	-21.86348	-22.40498	-1.565987	
C6orf52	-23.11693	-22.67335	-1.343997	
SLC5A10	-22.40977	-22.32619	-1.460341	
APOBEC3H	-23.29649	-22.81347	-1.370931	
LINC00574	-22.88247	-22.21847	-1.774904	
ATOH7	-23.02326	-22.96452	-1.657356	
GRAPL	-22.30881	-21.52397	-1.620106	
C6orf201	-22.25201	-21.85007	-1.707904	
RPL23AP64	-22.98331	-21.64723	-1.707398	
	LR	PValue	FDR	
RBM44	3115.908	0	0	
LPAL2	1931.307	0	0	
C6orf52	1762.968	0	0	
SLC5A10	4317.670	0	0	
APOBEC3H	1790.031	0	0	
LINC00574	1759.550	0	0	
ATOH7	2369.495	0	0	
GRAPL	1708.911	0	0	
C6orf201	2875.556	0	0	

RPL23AP64 2940.747 0 0

- Y elegir los contraste que queramos.
- Mostramos una comparación entre dos grupos.

```
AD = makeContrasts(contrast1 = Ascending_colon - Descending_colon,  
                  levels=design0)  
lrt3 = glmLRT(fit,contrast = AD)  
topTags(lrt3)
```

```
Coefficient: 1*Ascending_colon -1*Descending_colon  
              logFC      logCPM      LR      PValue      FDR  
ACTL8  -3.626585  1.9449815  41.05906  1.476981e-10  1.765755e-06  
DBH    -2.980882  -0.6717054  40.29327  2.185611e-10  1.765755e-06  
IGFN1  -3.772058  0.1735114  38.82410  4.637663e-10  2.497845e-06  
PCCA   -1.854167  6.0403726  37.84606  7.655289e-10  3.092354e-06  
INHA   -3.544573  -1.2757898  34.69282  3.860522e-09  1.247566e-05  
FLT3   -2.515272  -0.6541790  34.27551  4.783649e-09  1.288237e-05  
MUM1L1 -2.947467  -0.5294079  29.86079  4.642074e-08  1.071523e-04  
MYO3B  -2.250829  -1.0459367  28.57745  9.002435e-08  1.818267e-04  
KRT14   8.581216  2.8419848  26.64655  2.442861e-07  4.385750e-04  
PPP4R4 -2.363616  -1.4762586  24.69543  6.714311e-07  1.084898e-03
```

Bibliografía

- McCarthy, Davis J., Yunshun Chen, and Gordon K. Smyth. 2012. "Differential Expression Analysis of Multifactor RNA-Seq Experiments with Respect to Biological Variation." *Nucleic Acids Research* 40 (10): 4288–97. <https://doi.org/10.1093/nar/gks042>.
- Robinson, Mark D., and Gordon K. Smyth. 2007. "Moderated Statistical Tests for Assessing Differences in Tag Abundance." *Bioinformatics* 23 (21): 2881–87. <https://doi.org/10.1093/bioinformatics/btm453>.
- . 2008. "Small-Sample Estimation of Negative Binomial Dispersion, with Applications to SAGE Data." *Biostatistics* 9 (2): 321–32. <https://doi.org/10.1093/biostatistics/kxm030>.