

# Expresión diferencial con datos RNA-seq

Guillermo Ayala Gallego

2025-03-05

## Table of contents

<b>Introducción</b>	<b>1</b>
Paquetes . . . . .	1
<b>edgeR utilizando modelo lineal generalizado</b>	<b>1</b>
Modelo . . . . .	1
TCGA-COAD . . . . .	2
Bibliografía . . . . .	7

## Introducción

### Paquetes

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

## 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  $\phi_i$  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]
```

- 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)$"tissue_or_organ_of_origin"[to_keep]
                      + colData(tcga_coad)$"age_at_diagnosis"[to_keep])
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")
```

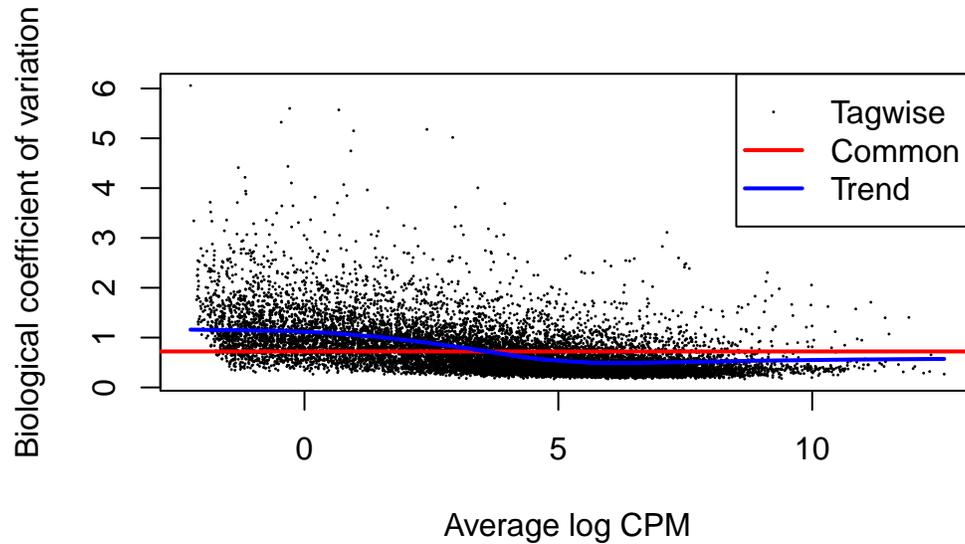
- Estimamos las dispersiones por tres métodos distintos:
  - Asumiendo una dispersión común,
  - una por gen y

– 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	Rectosigmoid_junction	Sigmoid_colon
	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			
ATOH7	-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

## Bibliografia

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>.