T1_Microarray

Francisco Martínez Picó

10/05/2020

Explicación del experimento

El experimento elegido estudia la disfunción de la barrera mucosa intestinal en pacientes de **Enfermedades Inflamatorias Intestintales**, más concretamente, de **enfermedad de Crohn** (*Crohn's Disease*, CD). Las Enfermedades Inflamatorias Intestinales (*Inflamatory Bowel Diseases*, IBD) son un conjunto de patologías de origen autoinmune que causan afectaciones inflamatorias en el sistema digestivo (predominantemente en el intestino) de los pacientes. Agrupa varias enfermedades, pero las más conocidas son la enfermedad de Crohn y la colitis ulcerosa. Curiosamente, estas enfermedades son más frecuentes en países industrializados (los conocidos como países del primer mundo).

Tanto la enfermedad de Crohn como la colitis ulcerosa se reconocieron de manera relativamente reciente, respectivamente en los siglos XIX y primer tercio del XX. Como otra gran cantidad de patologías autoinmunes, son de origen desconocido y no tienen cura. Los estudios apuntan a que un desencadenante desconocido (posiblemente un agente infeccioso o un antígeno de la alimentación) provoca una respuesta inflamatoria desordenada que termina atacando al propio intestino. Cada uno de los tipos de IBD tiene unos síntomas específicos, pero para ambas son comunes la diarrea (incluso apareciendo sangre o moco en heces), pérdida de peso, debilidad y la afectación del estado general.

En estudios previos, tras la identificación de los genes de susceptibilidad se ha demostrado que una disfunción en la barrera mucosa del intestino es clave para desarrollar CD. Sin embargo, el mecanismo de esta disfunción no ha sido todavía dilucidado. Por tanto, el objetivo de este estudio es el de elucidar el mecanismo molecular que subyace a la expresión de la **defensina alfa 6 humana (*human alfa-defensin* 6, HD6)** en pacientes con CD.

Se recogieron muestras de 15 pacientes con enfermedad de Crohn y 9 controles sanos mediante una endoscopia. Estas muestras eran de tejido sano (no inflamado).

El estudio obtuvo varios resultados, pero el más interesante fue que la expresión de HD6 (y no la de HD5) se encontraba reducida en el tejido no inflamado de pacientes con enfermedad de Crohn. Esto podría contribuir a la disfunción de la barrera mucosa intestinal.

Se publicó un artículo que puede encontrarse en el siguiente link: https://www.ncbi.nlm.nih.gov/pubmed/26891258

Cargando paquetes

Cargamos los paquetes necesarios:

- *Biobase*: este paquete para trabajar con Bioconductor contiene estructuras estandarizadas de datos para representar información genómica. Es decir, es necesario para trabajar con ExpressionSet.
- *GEOquery*: este paquete lo utilizaremos para acceder al repositorio del NCBI Gene Expression Omnibus (GEO) y descargar nuestros datos.
- *ArrayExpress*: nos permite acceder al repositorio europeo de ArrayExpress para descargar los ficheros necesarios.
- *affy*: nos permite trabajar y analizar datos obtenidos por Microarrays de Affymetrix.

- *affyPLM*: se trata de una extensión del paquete affy. Nos permitirá dibujar el MA-plot sobre el objeto ExpressionSet obtenido al realizar RMA.
- *geneplotter*: para llevar a cabo el control de calidad sobre los ExpressionSets obtenidos tras aplicar el método RMA sobre los datos crudos. -*AnnotationDbi*: este paquete nos permitirá realizar la anotación.

Dataset

Los datos pueden encontrarse en GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE69762

También pueden encontrarse en ArrayExpress: https://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-69762/

Los descargaremos desde GEO. Como necesitamos descargar los datos "crudos" utilizaremos la función "getGEOsuppFiles" (si utilizaramos "GEOquery" se bajarían los datos procesados). Esta función descarga un ".tar" dentro de un subdirectorio que crea (cuyo nombre es el código del estudio). Podemos descomprimirlo desde la terminal con "tar xvf archivo". Seguidamente, cargamos los datos:

```
getGEOSuppFiles("GSE69762")
system("tar xvf GSE69762/GSE69762_RAW.tar")
gse69762raw  = ReadAffy()
system("rm -fr GSE69762")
system("rm *CEL.gz")
```

Ahora ya tenemos los datos crudos cargados en R como un objeto "AffyBatch" con el nombre de "gse69762".

Primera aproximación a nuestro datos

Para conocer un poco más los datos con los que estamos trabajando. Estos trozos de código están preparados para no ejecutarse. Son simplemente para que el usuario conozca un poco más los datos con los que trabajaremos:

```
class(gse69762raw)
```

Se trata de un objeto de la clase AffyBatch.

```
dim(exprs(gse69762raw))
```

Tenemos 1.102.500 filas y 30 columnas. Es decir, trabajaremos con 1.102.500 sondas (contando PM y MM, además de las replicas presentes para cada sonda) y 30 muestras (explicadas en los detalles del experimento).

```
annotation(gse69762raw)
```

Se utiliza el chip "hugene10stv1".

Control de calidad

Analizaremos nuestros datos con el objetivo de decidir si es necesario descartar alguna de las muestras.

MA-plot

Para representar nuestros datos genómicos, utilizaremos en primer lugar la aplicación en MA-plots de los dibujos Media-Diferencia (Tukey) o dibujos de Bland-Altman.

Se tienen pares de puntos, que serán dos arrays o un array experimental y un array representativo, y se desea ver hasta que punto hay coincidencia. En estos dibujos, cada punto de los que se representa tiene en el eje x el valor "(xi+yi)/2" y en el eje y "xi-yi", donde la 'i' hace referencia a cada sonda y 'x' e 'y' hacen referencia a los arrays.

Se intenta representar la coincidencia, es decir, donde haya mucha hibridación, debería aparecer mucha hibridación en otros arrays. Idealmente, la imagen obtenida no debe ser muy abierta o cónica (donde la expresion fuera muy grande y la diferencia tambien fuera muy grande).

En la imagen, como puntos, aparecen solo los puntos extremos. Los cercanos a la linea horizontal se representan como una nube.

Para hacer nuestra comparación con muchos arrays se construye un array artificial. En este array artificial, para cada posición de cada sonda se toma la mediana (ya que esta, respecto a la muestra, es insensible a observaciones muy extremas) de los valores del resto de arrays. Se construye así esta especie de array artificial representativo donde cada posición tiene la mediana. Este microarray recibe el nombre de **chip de referencia pseudo-mediana**, y se utiliza para comparar con el resto de arrays experimentales.

```
affy::MAplot(gse69762raw,type="pm",plot.method="smoothScatter")
```

Analizando estos MA-plots obtenidos, vemos que los puntos se distribuyen formando una nube más o menos horizontal en torno al 0. Esto quiere decir que hay coincidencia en la hibridación, es decir: las posiciones en las que se obtiene una señal de hibridación mayor son coincidentes entre arrays.

Estimadores de densidad

A continuación, vamos a mostrar el comportamiento de los niveles de expresión para todas las sondas. Consideramos la posibilidad de que alguno de ellos tenga un nivel de ruido excesivo y, por tanto, debamos eliminarlo. Para llevar a cabo esta estimación de la densidad de los datos utilizaremos estimadores Kernel (no paramétricos) de densidad.

La idea básica es tratar de estimar la densidad en un punto x0 que se elige (no es un punto observado de los datos). Se elige un valor de H (ancho de banda) positivo. Este H tiene un valor relevante y arbitrario, pero su efecto sobre la representación es menor que el número de clases elegido tiene sobre un histograma. Seguidamente, se utilizan funciones kernel K(x) (que son funciones continuas, simetricas y positivas que van de 0 a 1). Se cuentan los puntos cercanos al elegido. Estos puntos pesan más si están más cerca del elegido. La función kernel utilizada puede variar.

```
affy::hist(gse69762raw)
```

Entre las distintas muestras, no hay una gran diferencia para los estimadores de la densidad: se puede observar un valor máximo de la densidad en la parte izquierda de la curva, causando una asimetría a la derecha.

Diagrama de cajas

Del mismo modo, podemos utilizar los diagramas de cajas para valorar la variabilidad de las expresiones en cada array y entre arrays.

Entre muestras, globalmente, los diagramas de cajas deberían ser similares. En un diagrama de cajas, la línea central es la mediana de esa muestra, la parte inferior de la caja es el primer cuartil, la parte superior el tercer cuartil (por lo que la mitad de los datos están dentro de la caja y los otros dos cuartos caen dentro de los bigotes).

```
affy::boxplot(gse69762raw)
```

A simple vista, tampoco se observa una gran diferencia entre las diferentes muestras. Por tanto, podemos decir que la dispersión entre muestras es similar. La mayoría de los datos se encuentran concentrados entre los mismos valores, más o menos.

Las distribuciones en todos los casos son simétricas, aunque se observa que los bigotes superiores son mayores. Esto correspondería con la expresión alta de ciertos genes (mayor señal en las sondas).

No parece haber ningún valor atípico o *outlier*.

Conclusión del control de calidad

Como resultado a las pruebas realizadas anteriormente, decidimos no descartar ninguna muestra.

Preprocesado

Para llevar a cabo el preprocesado de las muestras, utilizaremos el método *Robust Multichip Average* (RMA).

Robust Multichip Average (RMA)

Este método coge todo el experimento para corregir ruidos técnicos y eliminarlos. Trabaja con toda la experimentación (multichip). Además, utiliza únicamente el *perfect-matching*, ignorando el *miss-matching*. Esto supone una diferencia con el método original de MAS5 (que trabajaba tanto con PM como con MM). El RMA consta de tres pasos:

1. **Corrección de fondo**: normalmente, algunos genes que no deberían hibridar lo hacen, produciendo la conocida como "hibridación inespecífica". Por tanto, realmente, no existe el 0 o el negro en la imagen digital de la que partíamos. Como consecuencia, surge un problema: ¿qué es el fondo? Para solucionar esto, de manera resumida, se hace un promedio local para suavizar la imagen (denominado corrección de fondo). Realmente, este método no se encuentra explicado detalladamente en la bibliografía. Es decir, en este paso, se decide qué consideraremos 0. La corrección de fondo era mejor en el método de MAS5.
2. **Normalización de cuantiles**: se obtiene una matriz en los que cada columna tiene una muestra. En cada fila habría una sonda (estas sondas se repitirían 11 veces por gen, aunque ocupando filas distintas). Por tanto, tenemos 'N' (numero de muestras) x 'm' (número de sondas). Se cogen las columnas y se ordenan los valores de menor a mayor por filas (de cada columna). En la matriz, localizamos y observamos los valores más peqeños en la primera fila. Seguidamente, se haya la media de esos valores y se pone en otra matriz en toda la primera fila (repitiendolo n veces). Es decir, se obtienen desde la media de los más pequeños hasta la media de los más mayores. Después, se recupera el orden (la posición original) en cada columna. A esto se le llama normalización de cuantiles. Se modifican los valores originales de manera que se reemplazan los valores por las medias con frecuencia 1/N. Conseguimos que los niveles de gris sean "iguales" en todas las imágenes. Esto es similar a ecualizar una imagen, donde se redistribuyen los valores de gris (para ocupar más valores de los disponibles).
3. **Median polish**: este paso es específico de Affymetrix. Se repite la sonda un número determinado de veces a lo largo del array. Previamente, ya se ha quitado el ruido de fondo y se han normalizado todos los arrays. Ahora, tenemos 11 valores distintos para una misma sonda. Se elige una sonda determinada que se repite un número de veces, se monta una matriz. Se ponen sus 'x' valores en cada una de las columnas. Los margenes de la tabla (inferior y derecho) se inicializan a 0. Se hace un proceso iterativo por filas o por columnas (resultado final varía un poco según lo que se eliga).Se fija un criterio de parada: cuando los valores no varíen mucho ya con cada iteración. Se trabaja con mediana (en lugar de media) para evitar que valores muy extremos (debido a ruido) afecten. Además, evita aplicar modelos lineales por lo mismo, porque en estos se trabaja con el cuadrado de los datos y saldrían valores muy grandes.

Esta función implementada en el paquete 'affy' y nos devuelve un objeto de la clase ExpressionSet:

```
gse69762 = affy::rma(gse69762raw)
```

Control de calidad sobre los datos preprocesados

A continuación, aplicaremos el mismo control de calidad que hemos aplicado en el caso anterior, pero sobre los datos (ExpressionSet) obtenidos al aplicar el método RMA. Como ahora trabajaremos con el objeto ExpressionSet (diferente de AffyBatch de los datos "crudos"), utilizaremos otras funciones distintas para llevar a cabo este control de calidad.

MA-plot

```
affy::MAplot(gse69762,plot.method="smoothScatter") # Recordar: necesitamos cargar paquete affyPLM.
```

Al igual que en el caso anterior, podemos observar que la nube de puntos se encuentra centrada alrededor del 0. En este caso, esta agrupación en torno al 0 es todavía más acusada.

Estimadores de densidad

```
geneplotter::multidensity(exprs(gse69762),legend=FALSE) # Omitimos la leyenda para poder observar toda
```

Sigue habiendo una acumulación de valores en la parte izquierda de la gráfica. Sin embargo, esta parece haberse suavizado.

Diagrama de cajas

```
graphics::boxplot(exprs(gse69762))
```

Los datos se concentran en las mismas franjas de valores en las diferentes muestras. Las cajas siguen siendo simétricas, pero la asimetría del bigote superior se ha acentuado.

Anotación

Datos fenotípicos (pData)

Teníamos 30 muestras en el experimento. Para conocer más información de cada una de ellas, podemos volver a acceder a la web de GEO (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE69762) e ir al apartado de *samples*. Ahí podemos encontrar más información biológica sobre cada una de las muestras (aparecen en el mismo orden que las columnas de nuestro ExpressionSet):

- GSM1708491 -> nonIBD_jejunum_upper
- GSM1708492 -> nonIBD_jejunum_lower
- GSM1708493 -> nonIBD_ileum_lower
- GSM1708494 -> nonIBD_ascending colon
- GSM1708495 -> IBD_jejunum_upper
- GSM1708496 -> IBD_jejunum_lower
- GSM1708497 -> IBD_ileum_upper
- GSM1708498 -> IBD_ileum_middle
- GSM1708499 -> IBD_ascending colon
- GSM1708500 -> No.6_nonIBD_Jejunum_Upper
- GSM1708501 -> No.6_nonIBD_Jejunum_Lower
- GSM1708502 -> No7_nonIBD_Jejunum_Upper
- GSM1708503 -> No7_nonIBD_Jejunum_Lower
- GSM1708504 -> No.9_CD_Jejunum_Upper
- GSM1708505 -> No.9_CD_Jejunum_Lower
- GSM1708506 -> No.11_CD_Jejunum_Upper
- GSM1708507 -> No.11_CD_Jejunum_Lower
- GSM1708508 -> No.9_nonIBD_Jejunum_Upper
- GSM1708509 -> No.9_nonIBD_Jejunum_Middle
- GSM1708510 -> No.9_nonIBD_Jejunum_Lower
- GSM1708511 -> No.9_nonIBD_Ileum_Upper
- GSM1708512 -> No.9_nonIBD_Ileum_Middle
- GSM1708513 -> No.9_nonIBD_colon
- GSM1708514 -> No.12_CD_Jejunum_Upper_np

- GSM1708515 -> No.12_CD_Jejunum_Upper_ulcer
- GSM1708516 -> No.12_CD_Jejunum_Middle
- GSM1708517 -> No.12_CD_Jejunum_ Lower
- GSM1708518 -> No.12_CD_Ileum_Upper
- GSM1708519 -> No.12_CD_Ileum_ Lower
- GSM1708520 -> No.12_CD_colon

Como se puede observar, hay mucha información fenotípica de las muestras. Aparece información sobre:

1. **Estado de salud del paciente**: paciente con una Enfermedad Inflamatoria Intestinal (*Inflammatory Bowel Disease*, "IBD"), más concretamente, con enfermedad de Crohn (*Crohn's Disease*, "CD"), o individuo sano ("nonIBD").
2. **Segmento del aparato digestivo de donde se ha obtenido la biopsia**: yeyuno (intestino delgado), íleon (intestino delgado) o colon (intestino grueso).
3. **Sección del segmento de donde se ha obtenido la muestra**: superior o inferior.

Necesitamos crear grupos con estas muestras que, en un principio, parecen ser biológicamente bastante heterogéneas (la histología y afectación varía entre diferentes tractos del sistema digestivo). Por tanto, podemos consultar qué hicieron los autores originales (lo hacemos en el mismo experimento, pero en la web de ArrayExpress https://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-69762/:

Los autores originales consideraron dividir los grupos en **pacientes con CD (CD)** y **controles sanos (CONTROL)**, por lo que dividiremos nuestras muestras en los mismos grupos.

A continuación y echando un vistazo a las columnas de nuestro ExpressionSet, podemos ver que no están ordenadas: no aparecen todos los CD seguidos y, despúes, los controles. Por tanto y en consecuencia, nos montamos un vector adecuado a nuestros datos:

```
# En primer lugar, los factores (0 para los controles, 1 para los enfermos):
f1 = rep.int(0,4)
f2 = rep.int(1,5)
f3 = rep.int(0,4)
f4 = rep.int(1,4)
f5 = rep.int(0,6)
f6 = rep.int(1,7)

factores = c(f1,f2,f3,f4,f5,f6)
```

Y así creamos estos vectores con los factores que utilizaremos para las muestras (columnas) y que se adapta a nuestros datos.

```
type = factor(factores, levels=0:1, labels=c("CONTROL","CD"))

infor_fenotip = data.frame(pData(gse69762),type)
pData(gse69762) = infor_fenotip
```

Consultamos el resultado:

```
pData(gse69762)
```

Datos de las sondas y genes (fData)

Para asignar correctamente el fData a nuestro estudio debemos, en primer lugar, consultar el chip que se ha utilizado y encontrar el fichero de anotación correspondiente. Para consultar el chip utilizado:

```
annotation(gse69762)
```

Buscamos información respecto a este chip en Bioconductor y encontramos dos posibles archivos '.db' (bases de datos) a utilizar. La primera recibe el nombre de "hugene10stprobeset.db" y la segunda

"hugene10sttranscriptcluster.db". Utilizaremos el segundo archivo.

Por lo tanto, en el archivo "hugene10sttranscriptcluster.db" encontraremos la correspondencia entre las sondas utilizadas y la referencia de los genes con los que hibridan de la web ENTREZ (versión americana) y de la web ENSEMBL (versión europea). Es importante recordar que entre estas dos webs no hay bidireccionalidad.

En el archivo "hugene10sttranscriptcluster.db", también encontraremos más información, como el símbolo (nombre) que recibe el gen.

Descargamos este archivo:

```
BiocManager::install("hugene10sttranscriptcluster.db")
```

Y lo cargamos:

```
pacman::p_load(hugene10sttranscriptcluster.db)
```

Queremos asignar a nuestro fData (que actualmente sólo tiene información sobre las sondas) la información que corresponde para el gen con el que hibrida cada sonda. Es decir, para cada fila (sonda) tendremos:

- ID del gen en la web ENTREZ
- ID del gen en la web de ENSEMBL
- Símbolo o nombre que recibe

Podemos ver las claves (*keys*) con las que realizar consultas en "hugene10stprobeset.db":

```
keytypes(hugene10sttranscriptcluster.db)
```

Seguidamente, hacemos la consulta con las keys deseadas y guardamos todo esto en una variable:

```
probeInfo = AnnotationDbi::select(hugene10sttranscriptcluster.db,keys=featureNames(gse69762),columns=c(
```

Observamos que recibimos el siguiente *warning* o aviso: *'select()' returned 1:many mapping between keys and columns*. Es decir, obtenemos una relación 1:muchos. Esto significa que para varias sondas corresponden más de una entrada en alguna de las bases de datos (ENTREZ o ENSEMBL).

Como consecuencia a esto, decidimos quedarnos sólo con la primera coincidencia arbitrariamente. Para ello vemos las posiciones donde aparace la primera correspondencia y nos quedamos sólo con esas del data.frame probeInfo. Lo hacemos para el "PROBEID" y para la referencia de "ENTREZ":

```
posiciones = match(unique(probeInfo[,1]),probeInfo[,1]) # Para PROBEID
probeInfo = probeInfo[posiciones,]

posiciones = match(unique(probeInfo[,2]),probeInfo[,2]) # Para ENTREZ
probeInfo = probeInfo[posiciones,]
```

Pero aún no hemos terminado. Mirando nuestro data.frame probeInfo vemos que hay filas que tiene valores NA. Esto es debido a los controles. Debemos eliminarlos. Eliminamos en primer lugar las sondas que no cuentan con una correspondencia con la web ENTREZ. Seguidamente, eliminaremos el resto de NA:

```
probeInfo = probeInfo[!is.na(probeInfo[,"ENTREZID"]),]
probeInfo = na.omit(probeInfo)
```

Para terminar, asignamos este data.frame que ya tiene la correspondencia sonda - id gen ENTREZ - id gen ENSEMBL - símbolo al fData de nuestro ExpressionSet.

```
fData(gse69762) = probeInfo
```

Consultamos el resultado:

```
fData(gse69762)
```

Llegados a este punto, nuestro ExpressionSet ya tiene información respecto a las muestras (pData): si corresponden a un paciente sano o uno enfermo. También se han asignado los diferentes genes correspondientes a las sondas (fData). Se ha asignado la referencia de las webs de ENTREZ, ENSEMBL, así como el símbolo (nombre) por el que se conocen estos genes.

Para finalizar, incluiremos información sobre el experimento en el objeto de R.

```
infoData = new('MIAME',
        name='Hayashi et al.',
        lab='Not Specified',
        contact ='Kiichiro Tsuchiya <m.kool@dkfz.de>',
        title = 'Gene expression of human small intestine generated by biopsy specimens',
        abstract = 'Summary: The entire small intestine was obseved by balloon endoscopy. Biopsy specim
        url = 'https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE69762')

experimentData(gse69762) = infoData
```

Lo guardamos con todos los datos incorporados.

```
save(gse69762,file="../data/gse69762.rda")
```