ANÁLISIS DE DATOS DE MICROARRAY
EJERCICIOS

Los ejercicios sobre microarrays los realizaremos mediante R/BioConductor. Repasa la introducción sobre R para instalarlo y familiarizarte con los conceptos básicos.

Ejercicio 1

Preprocesamiento: lectura de datos crudos, control de calidad y normalización.

Utilizando R/BioConductor:

1. Usa la orden `library(affyPLM)` para cargar de golpe varios paquetes, entre ellos `affy`, `Biobase` y `gcrma`
2. Descarga los ficheros con instensidades (ficheros en crudo, o "raw") del experimento GSE1000 de GEO. Estos ficheros, en microarrays de Affymetrix, tienen extensión .CEL (más información sobre cómo descargar experimentos de microarray en http://www.ncbi.nlm.nih.gov/projects/geo/info/geo_paccess.html#FTP)
3. Usa la orden `abatch.raw=ReadAffy()` para leer todos los ficheros CEL en un directorio y construir un objeto de la clase `AffyBatch` de nombre `abatch.raw` (recuerda, `?ReadAffy` te muestra la ayuda sobre cómo se utiliza esta orden). ¿Qué dimensión tiene la matriz de identidad?¿Cuál es la media de las intensidades en la matriz?¿Y para cada una de las muestras?¿Qué otra información facilita el fichero `abatch.raw`?
4. Usa `MAplot(abatch.raw), hist(abatch.raw)` y `boxplot(abatch.raw)` para generar gráficas sobre el estado de los datos. ¿Qué calidad te parece que tienen los datos?¿Existe algún tipo de desviación, incongruencia o error?¿Podrías utilizarlos así, o crees que requieren normalización?
5. Preprocesa los datos mediante RMA con la orden `eset.rma=rma(abatch.raw)`. ¿Qué dimensión tiene la matriz de expresión?
6. Repite las gráficas del paso 4) pero para `eset.rma` ¿Qué pinta tienen ahora los datos?
7. Usa la ayuda (`?método`) para conocer más sobre estos métodos y las distintas opciones que permiten

Ejercicio 2

Conectividad con repositorio GEO

1. Usa la orden `library(GEOquery)` para cargar un paquete para descarga directa de datos de expresión desde GEO (sólo sirve experimentos ya procesados, no podemos descargar los ficheros CEL como hacíamos en el ejercicio 1)
2. Usa `eset.geo=getGEO("GSE1000")[[1]]` para obtener la matriz de expresión (normalizada) del mismo experimento del ejercicio anterior. El `[[1]]` es necesario porque GEO devuelve una lista de matrices (para el caso de experimentos realizados con más de una plataforma), y es el modo de acceder al primer elemento (en este caso sólo hay uno).
3. Realiza gráficas similares a las del paso 4) del ejercicio 1 para este objeto. ¿Los datos normalizados que encontramos en GEO te parecen de suficiente calidad? ¿Harías alguna transformación o normalización adicional?

Ejercicio 3

Conectividad con repositorio ArrayExpress

1. Busca en la web de ArrayExpress el experimento GSE1000. ¿Puedes encontrarlo? ¿Qué identificador tiene? ¿Qué repositorio te parece más informativo, GEO o ArrayExpress?
2. Usa la orden `library(ArrayExpress)` para cargar un paquete para descarga directa de datos de expresión desde ArrayExpress (a la inversa que GEO, sólo vamos a poder descargar experimentos en crudo, no podemos descargar los ficheros ya procesados)
3. Usa `eset.ae=ArrayExpress(id)` para obtener la matriz de expresión (en crudo) del mismo experimento del ejercicio anterior (`id` es el identificador de GSE1000 en ArrayExpress, entre comillas dobles). ¿Es igual en dimensiones y media al `abatch.raw` del ejercicio 1?¿Las condiciones están en el mismo orden?
4. Podríamos realizar el análisis exploratorio como en el paso 4 del ejercicio 1, pero vamos a utilizar en su lugar `library(arrayQualityMetrics)` para realizar todo el análisis de calidad de los datos de manera conjunta, generando un informe que nos indica las condiciones de dudosa calidad. Para realizar este análisis usa la orden `arrayQualityMetrics(eset.ae, outdir="ruta")`. Sustituye ruta por la dirección de la carpeta en la que almacenar el informe. ¿Qué ficheros genera?¿Hay condiciones con mala calidad?¿Coinciden con las conclusiones que extrajiste en el ejercicio 1?
5. Repetid el análisis de calidad pero ahora con la matriz normalizada mediante RMA en el ejericicio 1, `eset.rma`. ¿Hay diferencia?¿Alguna condición muestra signos de baja calidad?

Ejercicio 4

Explorando la información relativa al experimento y sus condiciones.

Aunque un experimento de microarray se define principalmente por los niveles de expresión, lo realmente importante son a qué corresponden esos niveles, es decir, la información que tenemos sobre las filas y columnas de la matriz.

Trabajaremos sobre los objetos `eset.rma` y `eset.ae` generados en ejercicios anteriores

1. Utiliza `eset.rma`, `annotation(eset.rma)` y `experimentData(eset.rma)`. ¿Qué información obtienes?¿Está completa? Repite con `eset.ae`, que era el mismo experimento, descargado de ArrayExpress ¿Cuál contiene más información?
2. Utiliza `sampleNames(eset.rma)` para obtener los nombres de las condiciones. ¿Qué forma tienen?¿Te dan alguna información?
3. Queremos las anotaciones de `eset.ae` (más información), pero la matriz de expresión normalizada a nuestro modo en `eset.rma` (mejor calidad). ¿Cómo podemos combinar ambas estructuras? Una posible opción:
   - `exprs(eset.ae)=exprs(eset.rma)[,sampleNames(eset.ae)]` remplazará la matriz de expresión de `eset.ae` por la de `eset.rma`. Cuidado en este caso, porque antes tenemos que asegurar que las columnas de `eset.rma` y de `eset.ae` estén en el mismo orden, es por ello que reordenamos las columnas según los nombres de las condiciones en `eset.ae` con `[,sampleNames(eset.ae)]`
   - ¿Se te ocurre alguna otra opción?
4. Los datos fenotípicos del experimento pueden consultarse mediante `pData(eset.ae)`. El objeto devuelto es un `data.frame`, cada fila contiene mucha información sobre cada una de las condiciones experimentales. ¿Qué información te parece, a priori, más relevante (indica los nombres de sus columnas)?
5. Dos tipos de información muy importantes son las características `(Characteristics..XXX.)` y los factores experimentales `(Factor.Value..XXX.)`. Las primeras indican características comunes a todas las muestras del experimento, fijando el contexto del mismo. Las segundas indican los factores que estudia el experimento. Por ejemplo, si estamos estudiando la variación en la expresión en el cerebro humano respecto al sexo, el tejido (cerebro) y el organismo (humano) son características fijas, mientras que el sexo (hombre, mujer) es el factor variable. ¿Cuáles son las características y factores experimentales en `eset.ae` (indica sus nombres tal como aparecen en el `data.frame`)?¿Qué valores toman en el experimento los factores experimentales?
6. La orden `pData(eset.ae)[,"Factor.Value..Time."]` nos devuelve los valores experimentales del factor tiempo. Para obtener las condiciones que tienen valor 6 podemos usar: `which(pData(eset.ae)[,"Factor.Value..Time."]==6)`. ¿Cuál es la media de expresión para las columnas de tiempo 6 y cuál para las de tiempo 32?

Ejercicio 5
Explorando la información relativa a las sondas y sus mapeos.

Los nombres de las 10 primeras sondas de nuestro experimento pueden obtenerse mediante `featureNames(eset.rma)[1:10]`. Los nombres de las sondas, de tipo *`xxxxx_at`*, son identificadores de Affymetrix, que generalmente se refieren a porciones de secuencias codificantes de genes (aunque también hay sondas de control, etc.). Para mapear los identificadores de sonda a los genes correspondientes tenemos los paquetes de anotación. Existe uno por cada plataforma de Affymetrix

1. Obtén el nivel de expresión de la sonda 202709_at para la condición 3.
2. Obtén la media de expresión para las columnas "GSM15791.cel" y "GSM15794.cel" de las sondas que van de la 1000 a la 2000, incluidas.
3. En nuestro caso, la plataforma es "*hgu133a*" (esta información la obtenemos con `annotation(eset.ae)`). Descarga e instala el paquete `hgu133a.db` para poder mapear sus sondas a genes.
4. Utiliza `hgu133a()` para ver todos los mapeos existentes. Verás mapeos a muchos identificadores ya conocidos, como UniGene, UniProt, Entrez, etc. ¿Cuáles te parecen más útiles?¿Hay alguno que no conozcas?
5. Para obtener el mapeo de la sonda 202709_at al nombre de su gen usamos la siguiente orden: `mget("202709_at", hgu133aGENENAME)`. De manera similar accedemos a cualquier otra sonda, o entorno, con `hgu133aNombreMapeo`. Así mismo, podemos obtener los mapeos de varias sondas a la vez usando un vector de nombres de sondas en vez de una sola. Prueba a obtener más información de esta sonda, y de las 10 primeras sondas de la matriz de expresión.

Ejercicio 6

Expresión diferencial

Vamos a probar con un experimento muy sencillo: GSE17636/E-GEOD-17636.

Es un experimento sobre una línea celular de cáncer de mama, que trata de medir la expresión en la línea tal cual (wt) y enriquecida con NF1-C2 (nf), un inhibidor tumoral. Hay tres réplicas de cada tipo, dando lugar a un total de 6 condiciones.

Ejecuta el siguiente código para tener el experimento listo para el análisis:

```
library(arrayExpress)          #adquisición
ae.raw=ArrayExpress("E-GEOD-17636")
library(affyPLM)        #normalización
ae.rma=rma(ae.raw)

#identificación de las condiciones wildtype y nf1-c2
wt=which(pData(ae.rma)[,"Factor.Value..CONDITION."]=="wildtype")
nf=which(pData(ae.rma)[,"Factor.Value..CONDITION."]!="wildtype")

#obtención de los nombres de los genes de cada sonda
library(hgu133plus2.db)
gn=unlist(mget(featureNames(ae.rma), hgu133plus2SYMBOL, ifnotfound=NA))
brca=grep("BRCA", gn)   #sondas correspondientes a BRCA
gn[brca]
```

¿Comprendes bien lo que se hace en cada paso? ¿Podrías explicarlos?

Puedes explorar los datos en mayor profundidad o hacer con ellos cualquier otro tratamiento adicional (de los vistos en anteriores ejercicios, o completamente distintos).

Realiza las siguientes tareas:

- Construye dos vectores, uno con la expresión media de cada sonda para las condiciones "wildtype" (`wt`) y otro con la expresión media de cada sonda para las condiciones nf1-c2 (`nf`).
- Calcula la expresión media para las sondas correspondientes a los genes BRCA1 y BRCA2, en ambos grupos. ¿Afecta el factor nf1-c2 a estos genes?
- Vamos ahora a ver qué genes se expresan diferencialmente en general. Para ello, lo primero que hay que calcular son los ratios de cambio. El log ratio se calcula como *ln(expresion1/expresion2).* Calculad los log ratios para todas las sondas tomando como expresión1 y expresión2 las medias obtenidas en el primer punto para nf y w2, respectivamente.
- Determina qué sondas tienen un ratio absoluto mayor que 0.8. Almacena sus posiciones en un vector llamado `degs`. ¿A qué genes mapean?
- Usa la función `t.test` para determinar la significatividad estadística de la expresión diferencial en una de las sondas obtenidas en el paso anterior. Por ejemplo, si una de las sondas es las 1061, habría que hacer: `t.test(exprs(ae.rma)[1061,wt], exprs(ae.rma)[1061,nf])` ¿Serías capaz de hacerlo para todas las sondas expresadas diferencialmente, utilizando `sapply`? Cuidado, si seleccionas muchas sondas, puede ser muy lento.

Ejercicio 7

Expresión diferencial con ajuste lineal.

Seguimos con el ejemplo anterior, ahora vamos a utilizar el paquete `limma`, que modela los datos mediante un ajuste lineal y estima la significatividad.

1. Carga el paquete `limma`
2. Utiliza `?limma` para ver cómo se utiliza este método. En especial, verás que habla de consultar la guía de usuario con `limmaUserGuide()`. En dicha guía, lee con atención la sección 8.5 (Two groups: Affymetrix) para realizar el análisis diferencial. Las secciones siguientes, con combinaciones de más grupos y análisis de series temporales, también son muy interesantes.

Ejercicio 8

Clustering jerárquico

Partimos del grupo de genes expresados diferencialmente por encima de un log ratio de 0.8 (vector `degs`) encontrado en ejercicio 6:

- Calcular la distancia entre sondas con `dr=dist(exprs(ae.rma[degs,]))`. Explora las distintas distancias que podemos utilizar (por defecto, distancia euclídea)
- Hacer un clustering jerárquico por filas con `hr=hclust(dr)`. De nuevo, podemos probar los distintos métodos de clustering jerárquico
- Imprimir el resultado con `plot(hr)`
- Repetir para las condiciones, generando `dc` y `hc`. Pista: para ello vamos a necesitar utilizar la matriz de expresión traspuesta.
- Generar un mapa de calor (heatmap) con los dendrogramas. Un mapa de calor muestra la matriz de expresión como una cuadrícula de color, cada cuadro representa la expresión de una sonda/gen para una condición, mediante una escala de colores. Pista: utilizar las funciones `heatmap` y `heatmap.2` (esta última está en el paquete `gplots`). Consultad su documentación, pues tienen una gran cantidad de argumentos configurables.

Ayuda: aquí tenéis un buen tutorial sobre clustering en R (y muchas más cosas): http://manuals.bioinformatics.ucr.edu/home/R_BioCondManual#clustering_hc

Ejercicio 9

Anotaciones funcionales: términos GO

Ya sabemos qué genes están expresados diferencialmente, su significatividad estadística, y podemos agruparlos mediante clustering.

Una última cosa que vamos a ver es si comparten alguna función biológica conocida. Para ello utilizaremos la ontología GO, a través del paquete `GOstats`. Usaremos un test hipergeométrico.

Recordad que para cualquier test de significatividad de este tipo básicamente chequeamos si el hecho que n genes (de un total de m genes diferenciados) estén anotados con un término (para el que hay un total de nt genes anotados en nuestro experimento) es significativo o no.

Primero, escoge, por ejemplo, las 50 sondas más expresadas diferencialmente según el análisis `limma` hecho en el ejercicio 7, mediante el uso de `topTable`.

- Determina los identificadores de Entrez para esas sondas, mediante el paquete `hgu133plus2.db` y la función `mget()`
- Determina los identificadores de Entrez para todas las sondas del experimento, de nuevo mediante el paquete `hgu133plus2.db` y la función `mget()`. Este conjunto se conoce como el "universo" del problema.
- Prepara un objeto de tipo `GOHyperParams`, donde geneIds será el primer grupo de identificadores que has buscado y universeGeneIds el segundo. Utiliza `ontology="BP"` y `annotation="hgu133plus2"`
- Ejecuta el test con la orden `hyperGTest` e inspecciona el resultado con `summary()`

¿Qué grupos aparecen como enriquecidos?¿Hay relación entre ellos?¿Hay relación con la razón biológica del experimento (evaluación del efecto de un represor tumoral)?¿Te parecen, viendo los números de genes anotados en el grupo respecto a todos los anotados en el universo, que el enriquecimiento de estos grupos es razonable?

Cambia el grupo de genes expresados diferencialmente, seleccionando por ejemplo aquellos por debajo de un p-valor (argumento `p.value` en `topTable`) o los que obtuvimos con un análisis más simple de la expresión diferencial en el ejercicio 6. ¿Qué relación hay entre estos grupos?¿Se parecen?¿Enriquecen los mismos términos GO?