T5_GeneSetAnalysis

Francisco Martínez Picó

7/14/2020

En esta viñeta se muestra el **análisis de grupos de genes** tanto del experimento de *Microarrays (gse69762)* como del experimento de *RNAseq (PRJNA601724)* (para obtener más información de estos estudios ir a las viñetas correspondientes). Es decir, se tratará de estudiar si hay relación entre un grupo de genes previamente definido y el fenotipo observado. En esta aproximación no usamos ningún conocimiento previo sobre relaciones conocidas entre genes que podrían ser esenciales a la hora de determinar la asociación conjunto de genes - fenotipo y no sólo gen - fenotipo

Más concretamente, se puede utilizar el concepto de **enriquecimiento** o **GSEA** (*Gene Set Enrichment Analysis*). Esto es debido a que este método trata de resumir o enriquecer la señal que da cada uno de los genes cogiendo el grupo.

Información sobre este procedimiento puede ser encontrada en:

- Aravind Subramanian y col. «Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide ex- pression profiles». En: *Proceedings of the National Academy of Sciences of the United States of America* 102.43 (2005), págs. 15545-15550. doi: 10.1073/pnas.0506580102.
- R.A. Irizarry y col. «Gene set enrichment analysis made sim- ple». En: *Statistical Methods in Medical Research* 18.6 (2009). Gene set analysis, págs. 565-575.

La información respecto a los grupos de genes han sido obtenidas de la **base de datos KEGG** (*Kyoto Encyclopedia of Genes and Genomes*).

Se utilizará el método implementado en el paquete **GSA**. Este utiliza la distribución de permutación a la que se aplica una reestandarización. La medida de enriquecimiento que proponen por defecto es la llamada **maxmean** (se toma la media de las partes positivas t+i, la media de las partes negativas t-i y nos quedamos con el máximo de ambos valores).

Cargando paquetes

Antes de empezar, cargamos los paquetes necesarios:

- *Biobase*: este paquete para trabajar con Bioconductor contiene estructuras estandarizadas de datos para representar información genómica.
- *SummarizedExperiment*: paquete utilizado para trabajar con el objeto RangedSummarizedExperiment, que contendrá toda la información de nuestro experimento de RNAseq.
- *GSA*: paquete utilizado para llevar a cabo este estudio de GSEA. Lleva implementado los métodos que utilizaremos.
- *EnrichmentBrowser*: utilizado para la descarga de grupos de genes de las BBDD que queramos (en este caso, KEGG).
- *org.Ce.eg.db*: para realizar la anotación de los genes del experimento de RNAseq. Este experimento se realiza sobre *C. elegans*. Los ID incluidos son los de WormBase y necesitamos los ENTREZ.
- *AnnotationDbi*: para llevar a cabo la anotación mencionada anteriormente de los ID de ENTREZ.
- *franciscomartinez*: el paquete que se está realizando para utilizar la función getGroupName.
- *tami*: para comparar hipótesis competitiva con hipótesis autocontenida.

Microarray

Cargamos los datos del experimento de Microarrays **gse69762**, correspondientes a la *Tarea 1*.

```
data(gse69762, package = 'franciscomartinez')
```

Descargamos los grupos de genes para *Homo sapiens* de la base de datos KEGG.

```
hsaKEGGgsc = getGenesets(org = 'hsa', db = 'kegg') # Bajamos info de grupos de genes para Homo sapiens
```

Se definen los grupos experimentales con valores 1 y 2. Esto es un requisito del paquete GSA.

```
f1 = rep.int(1,4)
f2 = rep.int(2,5)
f3 = rep.int(1,4)
f4 = rep.int(2,4)
f5 = rep.int(1,6)
f6 = rep.int(2,7)

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

Aplicando el método GSA

Y realizamos el análisis con GSA. Este método considera la **hipótesis autocontenida** (¿tiene nuestro grupo de genes una expresión diferencial asociada a las variables fenotípicas?) y fue propuesto en la siguiente referencia:

- Bradley Efron y Robert Tibshirani. «On testing the significan- ce of sets». En: *Annals of Applied Statistics* 1.1 (2007). Gene set analysis, págs. 107-129. doi: 10.1214/07-AOAS101.

```
gse69762.gsa = GSA(exprs(gse69762), grupo, genenames = fData(gse69762)[['ENTREZID']], genesets = hsaKEGG
```

Para visualizar los estadísticos enriquecidos para los grupos:

```
head(gse69762.gsa$GSA.scores)
```

Para observar los p-valores obtenidos para lo llamado **genes negativos** (genes que corresponden con genes que en la clase 2 tiene expresiones menores).

```
head(gse69762.gsa$pvalues.lo)
```

Para observar los p-valores obtenidos para lo llamado **genes positivos** (genes que corresponden con genes que en la clase 2 tiene expresiones mayor).

```
head(gse69762.gsa$pvalues.hi)
```

Los valores originales ti los tenemos con:

```
head(gse69762.gsa$gene.scores)
```

Observamos valores NA, serán genes que no aparecen en ninguno de los grupos seleccionados de la base de datos KEGG.

Analizando los resultados

Fijamos una tasa de error al 0,05. Vemos qué grupos presentan una asociación negativa o positiva con mayor diferencia significativa entre nuestros grupos.

```
ind.lo = which(gse69762.gsa$pvalues.lo < 0.05) # Negativa
ind.hi = which(gse69762.gsa$pvalues.hi <0.05) # Positiva
```

Para visualizar los identificadores KEGG para los grupos que tienen **asociación negativa**:

```
names(hsaKEGGgsc[ind.lo])
```

Para visualizar los identificadores KEGG para los grupos que tienen **asociación positiva**:

```
names(hsaKEGGgsc[ind.hi])
```

Sin embargo, si únicamente queremos saber el identificador del grupo (y no obtener con éste la información descriptiva del mismo) podemos modificar los nombres que obtiene la función getGenesets de la siguiente manera:

```
names(hsaKEGGgsc) = sapply(names(hsaKEGGgsc),getGroupName)
```

Y se vuelve a mostrar lo mismo, obteniendo ahora únicamente el identificador del grupo.

```
names(hsaKEGGgsc[ind.hi])
```

```
names(hsaKEGGgsc[ind.hi])
```

Estos son los datos que deberían ser interpretados por un especialista en esta enfermedad. Se observa una expresión diferencial en grupos de genes relacionados con el **metabolismo de diferentes metabolitos** (ciertos aminoácidos como la valina, propanoato) y algunos **procesos relacionados con el sistema inmune** (autofagia, fagosomas, activación de plaquetas, procesos inflamatorios...).

RNAseq

Cargamos los datos del experimento de RNAseq **PRJNA601724**, correspondientes a la *Tarea 3*.

```
data('PRJNA601724', package = 'franciscomartinez')
```

Como tenemos tres grupos y los métodos están preparados para trabajar únicamente con dos, decidimos trabajar arbitrariamente sólo con los grupos de *E.coli* y *Chryseobacterium*.

```
sel = colData(PRJNA601724)[,"Treatment"] == "E.coli" |
  colData(PRJNA601724)[,"Treatment"] == "Chryseobacterium"

sel = which(sel)

nuevo_se = PRJNA601724[,sel]
```

A continuación, descargamos los grupos de genes. De nuevo realizamos esta descarga utilizando la función getGenesets() del paquete EnrichmentBrowser de la base de datos KEGG, pero en esta ocasión para el organismo objeto de nuestro estudio (*C. elegans*).

```
celKEGGgsc = getGenesets(org = 'cel', db = 'kegg') # Para C. elegans y la base de datos KEGG.
```

Seguidamente pasamos a añadir los ENTREZ ID a nuestros datos, ya que con los ID de WormBase (los cuales ya están incluídos) no es suficiente. Esto es debido a que la información de grupos de genes descargada de KEGG viene con los ID de ENTREZ.

```
# Para añadir más información de anotación de los genes:
genesInfo = AnnotationDbi::select(org.Ce.eg.db, keys = rownames(nuevo_se), columns = c("ENTREZID", "SYM

# Nos quedamos con la primera coincidencia de WormBase:
posiciones = match(unique(genesInfo[,1]),genesInfo[,1]) # Para WORMBASE
genesInfo = genesInfo[posiciones,]
```

Ahora reasignamos los grupos experimentales que utilizaremos para este análisis, ya que la función GSA necesita que estén reasignados como "1" y "2".

```
f1 = rep.int(1,3)
f2 = rep.int(2,3)

grupo = c(f1,f2)
```

Finalmente, se realiza el análisis GSA.

```
PRJNA601724.gsa = GSA(assay(PRJNA601724), grupo, genenames = genesInfo[['ENTREZID']], genesets = celKEG
```

Analizando los resultados

Fijamos una tasa de error al 0,05. Vemos qué grupos presentan una asociación negativa o positiva con mayor diferencia significativa entre nuestros grupos.

```
ind.lo = which(PRJNA601724.gsa$pvalues.lo < 0.05) # Negativa
ind.hi = which(PRJNA601724.gsa$pvalues.hi <0.05) # Positiva
```

Para visualizar los identificadores KEGG para los grupos que tienen **asociación negativa**:

```
names(celKEGGgsc[ind.lo])
```

Para visualizar los identificadores KEGG para los grupos que tienen **asociación positiva**:

```
names(celKEGGgsc[ind.hi])
```

Sin embargo, si únicamente queremos saber el identificador del grupo (y no obtener con éste la información descriptiva del mismo) podemos modificar los nombres que obtiene la función getGenesets de la siguiente manera:

```
names(celKEGGgsc) = sapply(names(celKEGGgsc),getGroupName)
```

Y se vuelve a mostrar lo mismo, obteniendo ahora únicamente el identificador del grupo.

```
names(celKEGGgsc[ind.lo])
```

```
names(celKEGGgsc[ind.hi])
```

De nuevo, estos datos deberían ser interpretados por un especialista. Se puede observar que hay genes implicados en el **metabolismo general** (degradación de DNA, RNA, proteosoma, spliceosoma...) así como el **crecimiento celular** (RNA polimerasa, transporte de RNA, reparación de bases).

Hipótesis competitiva vs Hipótesis autocontenida: un ejemplo

Se define como **hipótesis competitiva** el plantear que nuestro grupo de genes tiene el mismo patrón de asociación con el fenotipo comparado con el resto de genes. Por otro lado, con la **hipótesis autocontenida** nos planteamos que nuestro grupo de genes no contiene algún gen cuyos niveles de expresión están asociados a las variables fenotípicas.

A continuación se utilizará el paquete tami para realizar el mismo anaálisis de grupos de genes pero teniendo en cuenta, por un lado, la hipótesis competitiva y, por otro lado, la hipótesis autocontenida. Los datos que se utilizarán serán los del ExpressionSet gse69762 anterior. Los grupos en este caso serán obtenidos de la base de datos *GeneOntology*.

```
hsaGOgsc = getGenesets(org = 'hsa', db = 'go') # Bajamos info de grupos de genes para Homo sapiens y la
```

Se realiza el análisis con la **hipótesis competitiva**. El estadístico será, al igual que el caso anterior, el maxmean.

```
set.seed(12345)
competitive = GeneSetTest(x = gse69762, y = "type",
                  test = rowt, association = "pvalue", correction = "BH",
                  GeneNullDistr = "randomization",
                  GeneSetNullDistr = "competitive",
                  alternative = "two-sided", nmax = 100,
                  id = "ENTREZID", gsc = hsaGOgsc, descriptive = maxmean,
                  minsize = 5, maxsize = 200,
                  foutput = "competitive")
dfcompet = tidy(competitive)
```

Para un alfa de 0,01, 39 grupos de genes tendrían una expresión diferencial signficativa bajo la hipótesis competitiva.

```
table(dfcompet$adjp < 0.01)
```

Se realiza el análisis con la **hipótesis autocontenida** El estadístico será, al igual que el caso anterior, el maxmean.

```
set.seed(12345)
self.contained = GeneSetTest(x = gse69762, y = "type",
                  test = rowt, association = "pvalue", correction = "BH",
                  GeneNullDistr = "randomization",
                  GeneSetNullDistr = "self-contained",
                  alternative = "two-sided", nmax = 100,
                  id = "ENTREZID", gsc = hsaGOgsc, descriptive = maxmean,
                  minsize = 5, maxsize = 200,
                  foutput = "self.contained")
dfself = tidy(self.contained)
```

En este caso, para un alfa de 0,01, 192 grupos de genes tendrían una expresión diferencial signficativa bajo la hipótesis autocontenida

```
table(dfself$adjp < 0.01)
```

Es decir, se puede observar que la significación de los grupos varía según la hipótesis que consideremos (39 bajo la hipótesis competitiva vs 192 bajo la hipótesis autocontenida.