

PEC2 Análisis de Datos Ómicos

Rita Ortega Vallbona

1 de junio, 2020

Contents

1	Abstract	2
2	Objetivos	2
3	Materiales y métodos	2
3.1	Los Datos	2
3.2	Preprocesado de los datos	3
3.2.1	Filtraje	3
3.2.2	Normalización	3
3.3	Identificación de genes diferencialmente expresados	3
3.4	Anotación de los resultados	3
3.5	Busca de patrones de expresión y agrupación de las muestras (comparación entre las distintas comparaciones)	3
3.6	Análisis de significación biológica (“ <i>Gene Enrichment Analysis</i> ”)	3
4	Resultados	3
5	Discusión	3
6	Apéndice	3
7	Bibliografía	3

1 Abstract

2 Objetivos

3 Materiales y métodos

3.1 Los Datos

Los datos proporcionados en el enunciado provienen del repositorio GTEx (*Genotype-Tissue Expression*), que recoge información de expresión específica de 54 tipos de tejido sano, proveniente de 1000 individuos. Este portal permite el acceso a los datos de expresión, imágenes de histología, etc.

Obtenemos los datos de targets y counts de los archivos proporcionados en el enunciado.

Con este script extraemos 10 muestras de cada grupo del archivo targets.csv y subseamos las columnas escogidas en el archivo counts.csv.

```
> # Separamos el dataframe que recoge los targets por grupos
> NIT <- subset(all_targets, Group == "NIT")
> SFI <- subset(all_targets, Group == "SFI")
> ELI <- subset(all_targets, Group == "ELI")
>
> # Seleccionamos 10 muestras de cada grupo y las unimos en un
> # único #dataframe que recoge los targets con los que
> # trabajaremos
> set.seed(params$seed.extract)
> NIT10 <- NIT[sample(nrow(NIT), size = 10, replace = FALSE), ]
> SFI10 <- SFI[sample(nrow(SFI), size = 10, replace = FALSE), ]
> ELI10 <- ELI[sample(nrow(ELI), size = 10, replace = FALSE), ]
>
> mytargets <- rbind(NIT10, SFI10, ELI10, deparse.level = 0)
>
> # Extraemos los nombres de las muestras y cambiamos los
> # guiones #por puntos para que coincidan con los nombres de
> # las muestras en #el dataframe de counts
> sample_names <- mytargets[, 3]
> s_names <- gsub("-", ".", sample_names)
>
> # Subseamos las columnas escogidas del dataframe de counts
> mycounts <- dplyr::select(all_counts, s_names)
```

3.2 Preprocesado de los datos

3.2.1 Filtrado

3.2.2 Normalización

3.3 Identificación de genes diferencialmente expresados

3.4 Anotación de los resultados

3.5 Busca de patrones de expresión y agrupación de las muestras (comparación entre las distintas comparaciones)

3.6 Análisis de significación biológica (“*Gene Enrichment Analysis*”)

4 Resultados

5 Discusión

6 Apéndice

7 Bibliografía