Deconvolución de datos de mealoma con Bisque: Datos bulk RNA-seq: GSE22155

Datos scRNA-seq: GSE72056

Elena Eyre Sánchez, PhD

2024-10-13

Contents

1 Introducción y Objetivo

2 Paquetes y datos

El paquete usado para este análisis es Bisque, el cual está diseñado para estimar proporciones celulares en datos bulk RNA-seq mediante el uso de datos scRNA-seq como referencia, cuando los datos bulk y scRNA-seq se generan con muestras con diferentes condiciones clínicas.

Repositorio GitHub de Bisque: https://github.com/cozygene/bisque

```
knitr::opts_chunk$set(warning=FALSE)
package_to_load <- c("readr", "dplyr", "ggplot2", "tidyr", "dplyr","RColorBrewer",
                     "Biobase","BisqueRNA", "gplots")
for (package in package_to_load) {
  require(package, character.only = T); packageVersion(package)
}
extra_to_load <- c("knitr","stringr", "stringi", "ggrepel","ggpubr", "ggbreak","reshape2","ggfortify","
for (package in extra_to_load) {
  require(package, character.only = T); packageVersion(package)
}
rm(package_to_load, extra_to_load)
```

#Datos

Hay dos tipos de input data: bulk RNA-seq y scRNA-seq. Los datos bulk RNA-seq son los que queremos deconvolucionar, y los datos scRNA-seq servirán como referencia de las poblaciones celulares a consultar.

`Bisque` requiere datos de expresión en formato ExpressionSet del paquete `Biobase`, así que previamente a aplicar `bisque` se necesita preparar los datos.

2.1 Bulk RNA-seq

En este análisis utilizo los datos del estudio GSE22155 descargados mediante la función `getGEO` des de la base de datos GEO, del NCBI: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE22155.

Las muestras consisten en 79 muestras analizadas con dos plataformas diferentes: GPL6102 (muestras de estadio III y IV) y GPL6947 (muestras de estadio IV). Este estudio es de especial interés para el TFM debido a que los autores también proporcionan metadata la supervivencia de los pacientes, cosa que permitirá estudiar posibles correlaciones con las poblaciones obtenidas de la deconvolución.

```
setwd("~/Desktop/ELENA_UOC/TFM")

gset <- getGEO("GSE22155", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL6102", attr(gset, "names")) else idx <- 1
gset_GPL6102 <- gset[[idx]]
#table(gset_GPL6102$characteristics_ch1.1) # OS (days)
#table(gset_GPL6102$characteristics_ch1.2) # event (0=alive, 1=dead)
#table(gset_GPL6102$characteristics_ch1.3) # sex
#table(gset_GPL6102$characteristics_ch1.4) # age at metastases
#table(gset_GPL6102$characteristics_ch1.5) # type of metastases: Lymohnode
#table(gset_GPL6102$characteristics_ch1.6) # age at primary diagnosis
#table(gset_GPL6102$characteristics_ch1.7) # localization of primary melanoma
#table(gset_GPL6102$characteristics_ch1.8) # type
#table(gset_GPL6102$characteristics_ch1.9) # breslow
#table(gset_GPL6102$characteristics_ch1.10) # clark
#table(gset_GPL6102$characteristics_ch1.11) # Stage (III and IV)
#table(gset_GPL6102$characteristics_ch1.12) # braf/nras
#table(gset_GPL6102$characteristics_ch1.13) # cdkn2a (hd=homozygous deletion, *=germline)
#table(gset_GPL6102$characteristics_ch1.14) # molecular subtype
#table(gset_GPL6102$characteristics_ch1.15)  # cd3 immunohistochemistry
#table(gset_GPL6102$characteristics_ch1.16) # cd20 immunohistochemistry
#table(gset_GPL6102$characteristics_ch1.17) # ki67 (0=<30%, 1=>30%)
#table(gset_GPL6102$`age at metastases:ch1`) # Age at metastases
#table(gset_GPL6102$`age at primary diagnosis:ch1`) # age at primary diagnosis
#table(gset_GPL6102$`localization of primary melanoma:ch1`) # localization of primary melanoma
#table(gset_GPL6102$`molecular subtype:ch1`) # molecular subtype

if (length(gset) > 1) idx <- grep("GPL6947", attr(gset, "names")) else idx <- 1
gset_GPL6947 <- gset[[idx]]
#table(gset_GPL6947$characteristics_ch1.1)# os (days)
#table(gset_GPL6947$characteristics_ch1.2) # event (0=alive, 1=dead):
#table(gset_GPL6947$characteristics_ch1.3) # sex
#table(gset_GPL6947$characteristics_ch1.4) # age at metastases
#table(gset_GPL6947$characteristics_ch1.5) # type of metastases
#table(gset_GPL6947$characteristics_ch1.6) # age at primary diagnosis
#table(gset_GPL6947$characteristics_ch1.7) # localization of primary melanoma
#table(gset_GPL6947$characteristics_ch1.8) # type
#table(gset_GPL6947$characteristics_ch1.9) # breslow
#table(gset_GPL6947$characteristics_ch1.10) # clark
#table(gset_GPL6947$characteristics_ch1.11) # stage
#table(gset_GPL6947$characteristics_ch1.12) # braf/nras
#table(gset_GPL6947$characteristics_ch1.13) # cdkn2a (hd=homozygous deletion, *=germline)
#table(gset_GPL6947$characteristics_ch1.14) # cdkn2a (hd=homozygous deletion, *=germline)
#table(gset_GPL6947$characteristics_ch1.15) # cd3 immunohistochemistry = NAs
#table(gset_GPL6947$characteristics_ch1.16) # cd20 immunohistochemistry = NAs
```

```r
#table(gset_GPL6947$characteristics_ch1.17) # ki67 (0=<30%, 1=>30%) = NAs
#table(gset_GPL6947$`localization of primary melanoma:ch1`) # localization of primary melanoma
#table(gset_GPL6947$`molecular subtype:ch1`) # molecular subtype
#table(gset_GPL6947$`tissue:ch1`) # tissue
#table(gset_GPL6947$`stage:ch1`) # All IV
#table(gset_GPL6947$`type of metastases:ch1`)# Type if metastases

# Debido a que los autores proporcionan los genes con la nomeclatura de Illumina, lo convierto a símbol
x <- illuminaHumanv4SYMBOL # cargado con el paquete illuminaHumanv4.db
mapped_probes <- mappedkeys(x) # Para sacar los símbolos
xx <- as.list(x[mapped_probes]) # Lo paso a listado
my_genes_GPL6102 <- as.data.frame(unlist(xx[(rownames(gset_GPL6102@assayData$exprs))])) # Lo convierto
my_genes_GPL6947 <- as.data.frame(unlist(xx[(rownames(gset_GPL6947@assayData$exprs))])) # Lo convierto
my_genes_GPL6102$gene <- rownames(my_genes_GPL6102)
my_genes_GPL6947$gene <- rownames(my_genes_GPL6947)

bulk_metadata_GPL6102 <- as.data.frame(gset_GPL6102@phenoData@data) # Paso la metadata disponible a una
bulk_metadata_GPL6947 <- as.data.frame(gset_GPL6947@phenoData@data) # Paso la metadata disponible a una

# Para usar los símbolos en lugar de nombres de ilumina, extraigo los datos de expresión:
bulk.mtx_GPL6102 <- as.data.frame(gset_GPL6102@assayData$exprs) # Los datos de expresión
bulk.mtx_GPL6947 <- as.data.frame(gset_GPL6947@assayData$exprs) # Los datos de expresión
bulk.mtx_GPL6102$gene <- rownames(bulk.mtx_GPL6102) # La columna que usaré para integrar
bulk.mtx_GPL6947$gene <- rownames(bulk.mtx_GPL6947) # La columna que usaré para integrar
bulk.mtx_GPL6102 <- inner_join(my_genes_GPL6102, bulk.mtx_GPL6102, by = "gene") # Integración de ambas
bulk.mtx_GPL6947 <- inner_join(my_genes_GPL6947, bulk.mtx_GPL6947, by = "gene") # Integración de ambas
bulk.mtx_GPL6102$gene <- NULL # Elimino la columna con nombres de Illumina
bulk.mtx_GPL6947$gene <- NULL # Elimino la columna con nombres de Illumina
colnames(bulk.mtx_GPL6102)[1] <- "symbols" # Nombro la columna de símbolos de los genes
colnames(bulk.mtx_GPL6947)[1] <- "symbols" # Nombro la columna de símbolos de los genes

# Agrego los posibles duplicados calculando la media:
bulk.mtx_GPL6102 <- aggregate(bulk.mtx_GPL6102, by = list(c(bulk.mtx_GPL6102$symbols)), mean) # Agregar
bulk.mtx_GPL6947 <- aggregate(bulk.mtx_GPL6947, by = list(c(bulk.mtx_GPL6947$symbols)), mean) # Agregar
rownames(bulk.mtx_GPL6102) <- bulk.mtx_GPL6102$Group.1 # Los nombres de genes únicos sin duplicados sir
rownames(bulk.mtx_GPL6947) <- bulk.mtx_GPL6947$Group.1 # Los nombres de genes únicos sin duplicados sir
bulk.mtx_GPL6102 <- bulk.mtx_GPL6102[,-c(1:2)] # Elimino las columnas usadas para conseguir los nombres
bulk.mtx_GPL6947 <- bulk.mtx_GPL6947[,-c(1:2)] # Elimino las columnas usadas para conseguir los nombres

# Convertir los datos de expresión del bulk RNA-seq a objeto ExpressionSet:
bulk.eset_GPL6102 <- Biobase::ExpressionSet(assayData = as.matrix(as.data.frame(bulk.mtx_GPL6102)))
bulk.eset_GPL6947 <- Biobase::ExpressionSet(assayData = as.matrix(as.data.frame(bulk.mtx_GPL6947)))
print("Object associated to platform GPL6102:")
```

```
## [1] "Object associated to platform GPL6102:"
```

```r
bulk.eset_GPL6102
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 16269 features, 57 samples
##   element names: exprs
## protocolData: none
## phenoData: none
## featureData: none
```

```
## experimentData: use 'experimentData(object)'
## Annotation:
```

```
print("Object associated to platform GPL6947:")
```

```
## [1] "Object associated to platform GPL6947:"
```

```
bulk.eset_GPL6947
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 19130 features, 22 samples
##   element names: exprs
## protocolData: none
## phenoData: none
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation:
```

3 scRNA-seq data

Los datos de expresión single-cell RNA de secuenciación (scRNA-seq) se recogen de muestras con una única condición, por ejemplo, sanos. Los tipos celulares del scRNA-seq son pre-determinados. Estos sirven como una referencia para estimar las proporciones del tipo celular de los datos bulk.

Para este análisis he escogido los datos procedentes del estudio GSE72056, que se encuentran células tumorales y no tumorales. Por ello, uno de los pasos a dar es la selección de las células no tumorales, entre las que se encuentran las células inmunitarias, que son de nuestro interés.

De nuevo, también procesaré los datos para obtener un objeto ExpressionSet, que en este caso también contendrá información sobre los tipos celulares con los que se identifica cada célula del estudio.

```
# Carga del fichero txt, después de descargar el archivo zip de GEO y despemaquetarlo:
GSE72056_melanoma_single_cell_revised_v2 <- read_delim("Datasets/GSE72056_melanoma_single_cell_revised_
    delim = "\t", escape_double = FALSE, trim_ws = TRUE)

# Proceso la información de la metadata, pues está en las primeras 3 líneas, con nombres de columnas mu
sc_metadata <- as.data.frame(t(GSE72056_melanoma_single_cell_revised_v2[1:3,]))
colnames(sc_metadata) <- sc_metadata[1,] # Los nombres de las columnas estan en la primera línea
sc_metadata <- sc_metadata[-1,] # Elimino línea de los nombres de columnas
colnames(sc_metadata)[2] <- "malignant" # Simplifico el nombre de la columna
sc_metadata$malignant <- sapply(sc_metadata$malignant, as.numeric) # Para poder seleccionar el valor qu
sc_metadata <- sc_metadata[sc_metadata$malignant == 1,] # Selecciono sólo las células no tumorales
colnames(sc_metadata)[3] <- "non_malignant" # Simplifico el nombre de la columna
sc_metadata$SampleID <- rownames(sc_metadata) # Genero una columna para identificar las células en el o
sc_metadata$non_malignant <- sapply(sc_metadata$non_malignant, as.numeric) # Para poder convertir los n
sc_metadata$Cell_type<- as.factor(if_else(sc_metadata$non_malignant == 1, "T_cell",
                                  ifelse(sc_metadata$non_malignant == 2, "B_cell",
                                         ifelse(sc_metadata$non_malignant == 3, "Macrophage",
                                                ifelse(sc_metadata$non_malignant == 4, "Endo_cell",
                                                       ifelse(sc_metadata$non_malignant == 5, "CAF",
                                                              ifelse(sc_metadata$non_malignant == 6, "NK"

# Preparo los apartados asociados a la metadata, y a los tipos celulares que bisque necesita saber, que
sc.pheno <- data.frame(check.names=F, check.rows=F, stringsAsFactors=F, row.names=sc_metadata$SampleID,
                       SubjectName=sc_metadata$SampleID, cellType=sc_metadata$Cell_type) # Phenodata pa
sc.meta <- data.frame(labelDescription=c("SampleID","Cell_type"),row.names=c("SampleID","Cell_type")) #
```

```
sc.pdata <- new("AnnotatedDataFrame",data=sc.pheno,varMetadata=sc.meta) # Este parámetro contiene ambos

# Procesamiento de los niveles de expresión de las células del scRNA-seq, que irán en el objeto Express
sc_gex_GPL6102 <-GSE72056_melanoma_single_cell_revised_v2[GSE72056_melanoma_single_cell_revised_v2$Cell
sc_gex_GPL6947 <-GSE72056_melanoma_single_cell_revised_v2[GSE72056_melanoma_single_cell_revised_v2$Cell
sc_gex2_GPL6102 <- sc_gex_GPL6102[rowSums(sc_gex_GPL6102[,-1]) != 0,] # También simplifico el análisis
sc_gex2_GPL6947 <- sc_gex_GPL6947[rowSums(sc_gex_GPL6947[,-1]) != 0,] # También simplifico el análisis
sc_gex2_GPL6102$Probes <- sc_gex2_GPL6102$Cell # Genero una nueva columna de los símbolos de los genes
sc_gex2_GPL6947$Probes <- sc_gex2_GPL6947$Cell # Genero una nueva columna de los símbolos de los genes
sc_gex2_GPL6102$Cell <- NULL # Elimino la columna con nombre confuso
sc_gex2_GPL6947$Cell <- NULL # Elimino la columna con nombre confuso
sc_gex2_GPL6102 <- aggregate(sc_gex2_GPL6102[,-ncol(sc_gex2_GPL6102)], by= list(c(sc_gex2_GPL6102$Probes
sc_gex2_GPL6947 <- aggregate(sc_gex2_GPL6947[,-ncol(sc_gex2_GPL6947)], by= list(c(sc_gex2_GPL6947$Probes
rownames(sc_gex2_GPL6102) <- sc_gex2_GPL6102$Group.1 # Los nombres de genes únicos sin duplicados sirve
rownames(sc_gex2_GPL6947) <- sc_gex2_GPL6947$Group.1 # Los nombres de genes únicos sin duplicados sirve
sc_gex3_GPL6102 <- sc_gex2_GPL6102[,colnames(sc_gex2_GPL6102) %in% sc_metadata$SampleID] # Me quedo con
sc_gex3_GPL6947 <- sc_gex2_GPL6947[,colnames(sc_gex2_GPL6947) %in% sc_metadata$SampleID] # Me quedo con

# Convertir los datos de expresión del scRNA-seq a objeto ExpressionSet:
sc.eset_GPL6102 <- Biobase::ExpressionSet(assayData=as.matrix(sc_gex3_GPL6102), phenoData=sc.pdata)
print("scRNA-seq objeto para analizar datos de la plataforma GPL6102:")
```

```
## [1] "scRNA-seq objeto para analizar datos de la plataforma GPL6102:"
```

```
sc.eset_GPL6102
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 14799 features, 3256 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: Cy72_CD45_H02_S758_comb CY58_1_CD45_B02_S974_comb ...
##     CY75_1_CD45_CD8_7__S274_comb (3256 total)
##   varLabels: SubjectName cellType
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation:
```

```
sc.eset_GPL6947 <- Biobase::ExpressionSet(assayData=as.matrix(sc_gex3_GPL6947), phenoData=sc.pdata)
print("scRNA-seq objeto para analizar datos de la plataforma GPL6947:")
```

```
## [1] "scRNA-seq objeto para analizar datos de la plataforma GPL6947:"
```

```
sc.eset_GPL6947
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 17344 features, 3256 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: Cy72_CD45_H02_S758_comb CY58_1_CD45_B02_S974_comb ...
##     CY75_1_CD45_CD8_7__S274_comb (3256 total)
##   varLabels: SubjectName cellType
##   varMetadata: labelDescription
## featureData: none
```

```
## experimentData: use 'experimentData(object)'
## Annotation:
```

3.1 Análisis de deconvolución

Por defecto, Bisque utiliza todos los genes para decomposición.

Hay que indicar que no hay solapamiento de muestras por tratarse de muestras diferentes en el bulk RNA-seq y el scRNA-seq, pues Bisque podría realizar un análisis usando las mismas muestras si lo quisiéramos: `use.overlap=FALSE`.

```
res_GPL6102 <- BisqueRNA::ReferenceBasedDecomposition(bulk.eset_GPL6102, sc.eset_GPL6102, markers=NULL,
```

```
## Decomposing into 7 cell types.
## Using 14799 genes in both bulk and single-cell expression.
## Converting single-cell counts to CPM and filtering zero variance genes.
## Filtered 22 zero variance genes.
## Converting bulk counts to CPM and filtering unexpressed genes.
## Filtered 0 unexpressed genes.
## Generating single-cell based reference from 3256 cells.
## Inferring bulk transformation from single-cell alone.
## Applying transformation to bulk samples and decomposing.
```

```
res_GPL6947 <- BisqueRNA::ReferenceBasedDecomposition(bulk.eset_GPL6947, sc.eset_GPL6947, markers=NULL,
```

```
## Decomposing into 7 cell types.
## Using 17344 genes in both bulk and single-cell expression.
## Converting single-cell counts to CPM and filtering zero variance genes.
## Filtered 26 zero variance genes.
## Converting bulk counts to CPM and filtering unexpressed genes.
## Filtered 0 unexpressed genes.
## Generating single-cell based reference from 3256 cells.
## Inferring bulk transformation from single-cell alone.
## Applying transformation to bulk samples and decomposing.
```

Encontramos las proporciones del bulk RNA-seq en el apartado `bulk.props`, el qual puedo integrar en la metadata que ya tenía y almacenar en un archivo para posteriores análisis.

```
ref.based.estimates_GPL6102 <- as.data.frame(t(res_GPL6102$bulk.props))
ref.based.estimates_GPL6947 <- as.data.frame(t(res_GPL6947$bulk.props))
ref.based.estimates_GPL6102$geo_accession <- rownames(ref.based.estimates_GPL6102)
ref.based.estimates_GPL6947$geo_accession <- rownames(ref.based.estimates_GPL6947)
ref.based.estimates_GPL6102 <- inner_join(ref.based.estimates_GPL6102, bulk_metadata_GPL6102, by = "geo
ref.based.estimates_GPL6947 <- inner_join(ref.based.estimates_GPL6947, bulk_metadata_GPL6947, by = "geo
knitr::kable(head(ref.based.estimates_GPL6102[,1:7]), digits=2, caption = "Sección de las primeras mues
```

Table 1: Sección de las primeras muestras como ejemplo del resultado con la plataforma GPL6102

B_cell	CAF	Endo_cell	Macrophage	NK	Other_cells	T_cell
0.57	0.00	0.00	0.00	0.00	0.00	0.43
0.11	0.00	0.00	0.00	0.00	0.40	0.49
0.11	0.00	0.00	0.06	0.05	0.00	0.78
0.14	0.00	0.03	0.00	0.02	0.25	0.56
0.20	0.08	0.01	0.00	0.01	0.16	0.54
0.16	0.00	0.00	0.00	0.00	0.30	0.53

```
knitr::kable(head(ref.based.estimates_GPL6947[,1:7]), digits=2, caption = "Sección de las primeras mues
```

Table 2: Sección de las primeras muestras como ejemplo del resultado con la plataforma GPL6947

B_cell	CAF	Endo_cell	Macrophage	NK	Other_cells	T_cell
0.17	0.03	0.10	0.01	0.07	0.00	0.62
0.22	0.00	0.00	0.00	0.00	0.17	0.61
0.12	0.02	0.03	0.05	0.01	0.18	0.59
0.20	0.00	0.00	0.00	0.00	0.28	0.53
0.12	0.01	0.00	0.00	0.00	0.42	0.45
0.37	0.00	0.00	0.00	0.00	0.00	0.63

```
write.csv(ref.based.estimates_GPL6102,"./bisque_GSE22155_GPL6102.csv", row.names = FALSE)
write.csv(ref.based.estimates_GPL6947,"./bisque_GSE22155_GPL6947.csv", row.names = FALSE)
```