

Deconvolución de datos de melanoma con quantiseqr:

Datos bulk RNA-seq: GSE35640

Datos scRNA-seq: GSE72056

Elena Eyre Sánchez, PhD

2024-10-27

Contents

1	Introducción y Objetivo	1
2	Paquetes y datos	1

1 Introducción y Objetivo

2 Paquetes y datos

Repositorio GitHub de : https://github.com/Danko-Lab/quantiseqr/blob/main/tutorial_deconvolution.pdf

```
knitr::opts_chunk$set(warning=FALSE)
# if (!requireNamespace("BiocManager", quietly = TRUE)) {
#   install.packages("BiocManager")
# }
# BiocManager::install("quantiseqr")
package_to_load <- c("readr", "dplyr", "ggplot2", "tidyr", "dplyr", "RColorBrewer",
                    "Biobase", "quantiseqr", "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", "cowplot", "GEOquery", "Seurat", "data.table",
                  "limma", "illuminaHumanv4.db", "SummarizedExperiment", "tibble")
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.

#Deconvolución

En este análisis utilizo los datos del estudio GSE35640 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=GSE35640>.

Las muestras consisten en 65 muestras analizadas con la plataformas GPL570, y con varios tratamientos: dabrafenib + trametinib.

Este estudio es de especial interés para el TFM debido a que los autores también proporcionan metadata la respuesta de los pacientes, cosa que permitirá estudiar posibles correlaciones con las poblaciones obtenidas de la deconvolución.

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

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

gset <- getGEO("GSE35640", GSEMatrix = TRUE, getGPL = FALSE)
if (length(gset) > 1) idx <- grep("GPL570", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
#table(gset$characteristics_ch1) # treatment response for gene profiling:
#table(gset$characteristics_ch1.1) # response
#table(gset$characteristics_ch1.2) # tissue

# Debido a que los autores proporcionan los genes con la nomenclatura de Affymetrix, lo convierto a símbolos
library(biomaRt)
my_genes <- as.data.frame((rownames(gset@assayData$exprs))) # Lo convierto en tabla para poder consultar
colnames(my_genes) <- "Aff_IDs" # Etiqueto la columna de los nombres de Affymetrix
mart <- biomaRt::useEnsembl(biomart="ensembl", dataset="hsapiens_gene_ensembl") # Recopilo los nombres
biomart_genes <- biomaRt::getBM(attributes=c("hgnc_symbol", "affy_hg_u133_plus_2"), # conversión que necesito
                              filters = "affy_hg_u133_plus_2", values = my_genes$Aff_IDs, mart = mart) # Biomart me proporciona

bulk_metadata <- as.data.frame(gset@phenoData@data) # Paso la metadata disponible a una tabla

# Para usar los símbolos en lugar de nombres de illumina, extraigo los datos de expresión:
bulk.mtx <- as.data.frame(gset@assayData$exprs) # Los datos de expresión
bulk.mtx$affy_hg_u133_plus_2 <- rownames(bulk.mtx) # La columna que usaré para integrar
bulk.mtx <- inner_join(biomart_genes, bulk.mtx, by = "affy_hg_u133_plus_2") # Integración de ambas tablas
bulk.mtx$affy_hg_u133_plus_2 <- NULL # Elimino la columna con nombres de Affymetrix
colnames(bulk.mtx)[1] <- "symbols" # Cambio la columna de símbolos de los genes

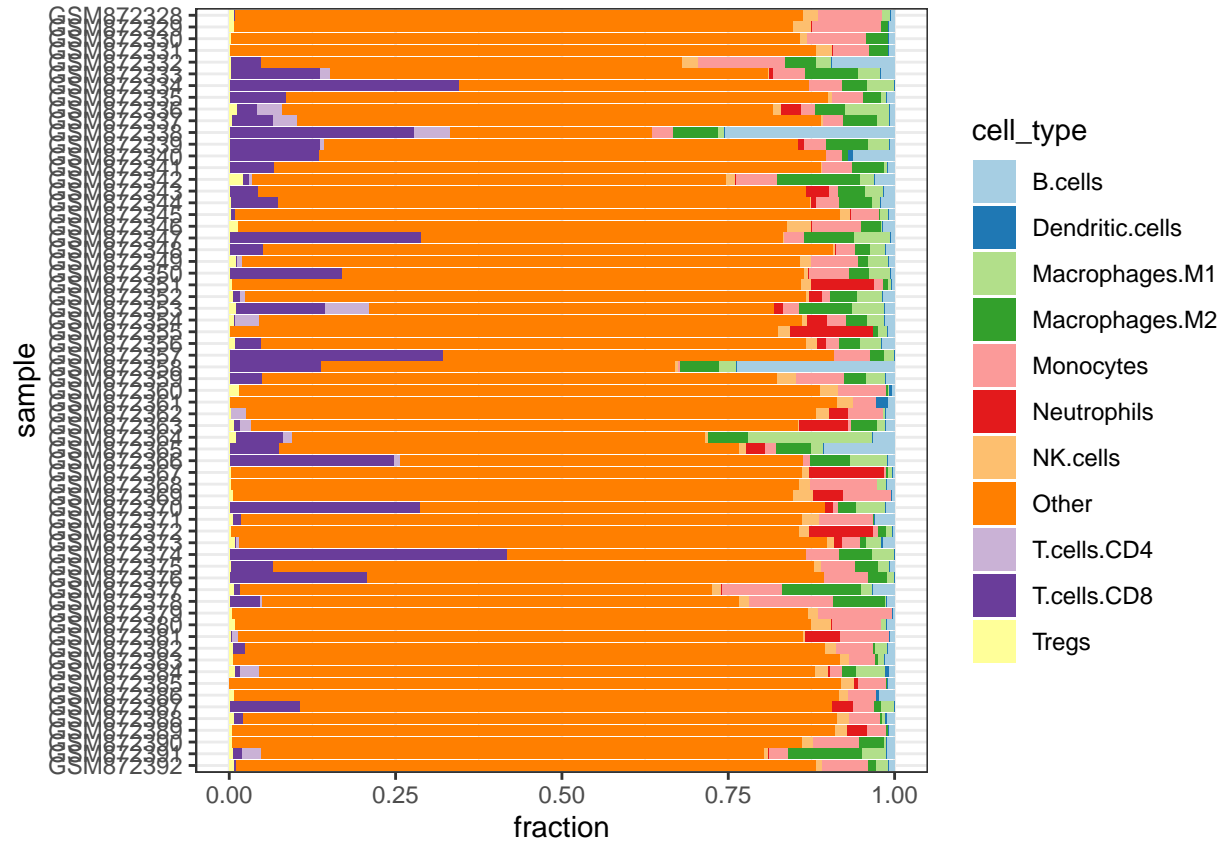
# Agrego los posibles duplicados calculando la media:
bulk.mtx <- aggregate(bulk.mtx, by = list(c(bulk.mtx$symbols)), mean) # Agregar
rownames(bulk.mtx) <- bulk.mtx$Group.1 # Los nombres de genes únicos sin duplicados sirven para dar nombre
bulk.mtx <- bulk.mtx[,-c(1:2)] # Elimino las columnas usadas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

# Convertir los datos de expresión del bulk RNA-seq a objeto ExpressionSet:
bulk.eset <- Biobase::ExpressionSet(assayData = as.matrix(as.data.frame(bulk.mtx)))
print("ExpressionSet object:")

## [1] "ExpressionSet object:"
bulk.eset

## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 22384 features, 65 samples
## element names: exprs
## protocolData: none
## phenoData: none
## featureData: none
```

```
## experimentData: use 'experimentData(object)'
## Annotation:
ti_racle <- quantiseqr::run_quantiseqr(
  expression_data = bulk.eset@assayData$exprs,
  signature_matrix = "TIL10",
  is_arraydata = FALSE,
  is_tumordata = TRUE,
  scale_mRNA = TRUE
)
quantipLOT(ti_racle)
```



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

```
ref.based.estimates <- as.data.frame(ti_racle)
ref.based.estimates$geo_accession <- rownames(ref.based.estimates)
ref.based.estimates <- inner_join(ref.based.estimates, bulk_metadata, by = "geo_accession")
knitr::kable(head(ref.based.estimates[,1:7]), digits=2, caption = "Sección de las primeras muestras y c
```

Table 1: Sección de las primeras muestras y columnas como ejemplo del resultado

Sample	B.cells	Macrophages.M1	Macrophages.M2	Monocytes	Neutrophils	NK.cells
GSM872328	0.01	0.01	0.00	0.10	0.00	0.02
GSM872329	0.01	0.00	0.01	0.10	0.00	0.03
GSM872330	0.01	0.00	0.03	0.09	0.00	0.01
GSM872331	0.01	0.00	0.03	0.06	0.00	0.02

Sample	B.cells	Macrophages.M1	Macrophages.M2	Monocytes	Neutrophils	NK.cells
GSM872332	0.10	0.02	0.05	0.13	0.00	0.02
GSM872333	0.02	0.03	0.08	0.05	0.01	0.00

```
write.csv(ref.based.estimateds, "./quantiseq_GSE35640.csv", row.names = FALSE)
```