

Deconvolución de datos de melanoma con quantiseqr: Datos bulk RNA-seq: GSE50509

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

#Deconvolución

En este análisis utilizo los datos del estudio GSE50509 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=GSE50509>.

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")

gset <- getGEO("GSE50509", GSEMatrix = TRUE, getGPL = FALSE)
if (length(gset) > 1) idx <- grep("GPL10558", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
#table(gset$characteristics_ch1) # genotype: BRAF V600E, BRAF V600K, and BRAF V600R
#table(gset$characteristics_ch1.2) # location
table(gset$characteristics_ch1.3) # treatment: dabrafenib, none, or vemurafenib

##
## treatment: dabrafenib          treatment: none treatment: vemurafenib
##                25                28                8
# Debido a que los autores proporcionan los genes con la nomenclatura de Illumina, lo convierto a símbolos
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 <- as.data.frame(unlist(xx[(rownames(gset@assayData$exprs))])) # Lo convierto en tabla para poder usarlos
my_genes$gene <- rownames(my_genes)

bulk_metadata <- as.data.frame(gset@phenoData@data) # Paso la metadata disponible a una tabla
bulk_metadata_descr <- data.frame(stringr::str_split_fixed(bulk_metadata$description, " ", 3)) # Separo la descripción en 3 partes
bulk_metadata_descr$X1 <- NULL # Descarto la columna que sólo menciona la anotación "Patient"
colnames(bulk_metadata_descr) <- c("Patient_ID", "Description_2") # Nombro las nuevas columnas
bulk_metadata_descr$geo_accession <- bulk_metadata$geo_accession # Genero la columna de los IDs de los genes
bulk_metadata <- inner_join(as.data.frame(bulk_metadata), as.data.frame(bulk_metadata_descr), by = "geo_accession")

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

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

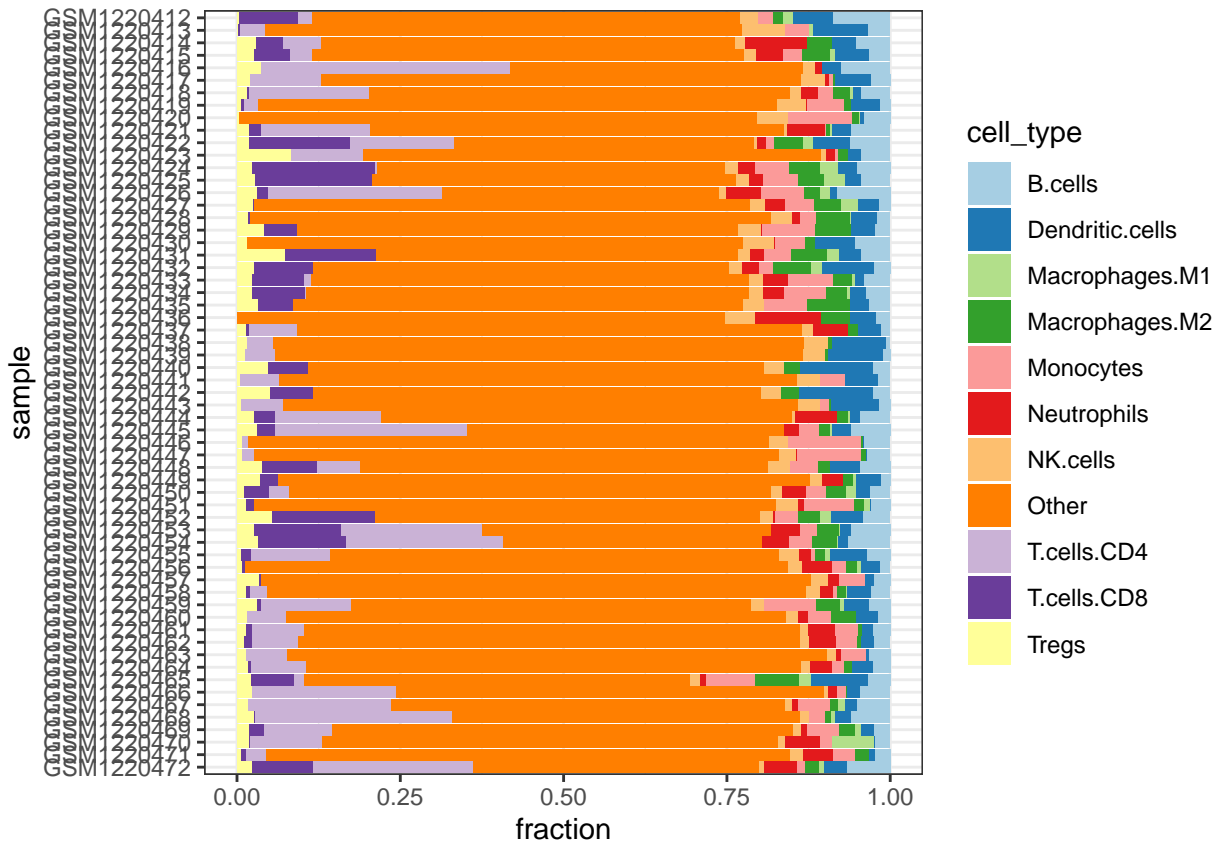
# Convertir los datos de expresión del bulk RNA-seq a objeto ExpressionSet:
bulk.eset <- Biobase::ExpressionSet(assayData = as.matrix(as.data.frame(bulk.mtx)))
print("ExpressionSet object:")

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

## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 20909 features, 61 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
GSM1220412	0.09	0.02	0.02	0.02	0.00	0.03
GSM1220413	0.03	0.01	0.00	0.04	0.00	0.07
GSM1220414	0.05	0.00	0.04	0.00	0.09	0.02
GSM1220415	0.03	0.01	0.04	0.03	0.04	0.02
GSM1220416	0.08	0.00	0.00	0.00	0.01	0.02

Sample	B.cells	Macrophages.M1	Macrophages.M2	Monocytes	Neutrophils	NK.cells
GSM1220417	0.03	0.00	0.01	0.01	0.01	0.04

```
write.csv(ref.based.estimateds,"./quantiseq_GSE50509.csv", row.names = FALSE)
```