GSE156902

Mattia Manna

2025-01-19

Indice

1 Importare dati

1.1 GSE68086 (Dataset BestTal 285 pazienti 6 tipi tumori + sani) TRAIN

1.1.1 Importare dati sui pazienti GSE68086

```
patientsGSE68086 <- read.csv("~/Desktop/Università/Magistrale/Tesi/R/GSE68086/patients.csv", row.names=
dim(patientsGSE68086)
```

```
## [1] 285  51
```

1.1.2 Importare dati sulle conte GSE68086

```
tep.exprGSE68086 <- read.delim("~/Desktop/Università/Magistrale/Tesi/R/GSE68086/GSE68086_TEP_data_matri
dim(tep.exprGSE68086)
```

```
## [1] 57736   285
```

```
colnames(patientsGSE68086)
```

```
##  [1] "title"                   "geo_accession"
##  [3] "status"                  "submission_date"
##  [5] "last_update_date"        "type"
##  [7] "channel_count"           "source_name_ch1"
##  [9] "organism_ch1"            "characteristics_ch1"
## [11] "characteristics_ch1.1"   "characteristics_ch1.2"
## [13] "characteristics_ch1.3"   "characteristics_ch1.4"
## [15] "characteristics_ch1.5"   "molecule_ch1"
## [17] "extract_protocol_ch1"    "extract_protocol_ch1.1"
## [19] "taxid_ch1"               "description"
## [21] "data_processing"         "data_processing.1"
## [23] "data_processing.2"       "data_processing.3"
## [25] "data_processing.4"       "data_processing.5"
## [27] "platform_id"             "contact_name"
## [29] "contact_email"           "contact_laboratory"
## [31] "contact_department"      "contact_institute"
## [33] "contact_address"         "contact_city"
## [35] "contact_zip.postal_code" "contact_country"
## [37] "data_row_count"          "instrument_model"
## [39] "library_selection"       "library_source"
## [41] "library_strategy"        "relation"
## [43] "relation.1"              "supplementary_file_1"
## [45] "batch.ch1"               "cancer.type.ch1"
## [47] "cell.type.ch1"           "mutational.subclass.ch1"
## [49] "patient.id.ch1"          "tissue.ch1"
## [51] "sample_name"
```

1.2 GSE183635 (2000 e oltre pazienti 18 tipi tumori + sani) TEST

1.2.1 Importare dati sui pazienti GSE183635

```
patientsGSE183635 <- read_csv("/Users/mattia/Desktop/Università/Magistrale/Tesi/R/Risoluzione Data Leaka
```

```
## Rows: 1646 Columns: 43
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (40): title, geo_accession, status, submission_date, last_update_date, t...
## dbl  (3): channel_count, taxid_ch1, data_row_count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(patientsGSE183635)
```

```
## [1] 1646   43
```

1.2.2 Importare dati sulle conte GSE183635

```
load("~/Desktop/Università/Magistrale/Tesi/R/Risoluzione Data Leakage/Dati/GSE183635_TEP_Count_Matrix.Rd
tep.exprGSE183635 <- as.data.frame(TEP_Count_Matrix)
rm(TEP_Count_Matrix)
```

1.3 GSE156902 (GBM nel tempo) TEST

1.3.1 Importare dati sui pazienti GSE156902

```
library(GEOquery)
library(readr)

dataset <- "GSE156902"

# Scaricare informazioni riguardo ai geni
getGEOSuppFiles(dataset) # Check for available supplementary files

# Scaricare informazioni riguardanti i samples
patients <- getGEO(dataset,GSEMatrix=T)
patients<- pData(phenoData(patients[[1]]))
#write_csv(patients,file = "/Users/mattia/Desktop/Università/Magistrale/Tesi/R/GSE156902/patientGSE1569
```

```
#p <- pData(phenoData(patients$`GSE156902-GPL20301_series_matrix.txt.gz`))

#pp <- experimentData(patients$`GSE156902-GPL20301_series_matrix.txt.gz`)
```

```
patientsGSE156902 <- read_csv("/Users/mattia/Desktop/Università/Magistrale/Tesi/R/GSE156902/patientGSE15
```

```
## Rows: 600 Columns: 40
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (37): title, geo_accession, status, submission_date, last_update_date, t...
## dbl  (3): channel_count, taxid_ch1, data_row_count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(patientsGSE156902)
```

```
## [1] 600  40
```

```
table(patientsGSE156902$`group:ch1`)
```

```
##
## asymptomaticControls            brainMeta                  GBM
##                  252                  125                  156
##     multipleSclerose
##                   67
```

```
library(dplyr)
library(stringr)

patientsGSE156902_GBM <- patientsGSE156902[patientsGSE156902$`group:ch1` == "GBM",]
nrow(patientsGSE156902_GBM)
```

```
## [1] 156

# Filtrare le righe in cui X contiene "t0"
t0 <- patientsGSE156902_GBM %>% filter(!str_detect(title, "t1|t2|t3|t4|t5|t6|t7|t8|t9|t10"))
nrow(t0)
## [1] 70

t1 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t1"))
nrow(t1)
## [1] 32

t2 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t2"))
nrow(t2)
## [1] 18

t3 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t3"))
nrow(t3)
## [1] 8

t4 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t4"))
nrow(t4)
## [1] 13

t5 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t5"))
nrow(t5)
## [1] 7

t6 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t6"))
nrow(t6)
## [1] 4

t7 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t7"))
nrow(t7)
## [1] 2

t8 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t8"))
nrow(t8)
## [1] 1

t9 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t9"))
nrow(t9)
## [1] 1

t10 <- patientsGSE156902_GBM %>% filter(str_detect(title, "t10"))
nrow(t10)
## [1] 1
```

```
nrow(t0) +nrow(t1) + nrow(t2) + nrow(t3) + nrow(t4) + nrow(t5) + nrow(t6) + nrow(t7) + nrow(t8) + nrow(
```

```
## [1] 157
```

1.3.2 Importare dati sulle conte GSE156902

```
load("~/Desktop/Università/Magistrale/Tesi/R/GSE156902/GSE156902_TEP_Count_Matrix.RData")
tep.exprGSE156902.raw <- as.data.frame(dgeIncludedSamples$raw.counts)
tep.exprGSE156902 <- as.data.frame(dgeIncludedSamples$counts)
dim(tep.exprGSE156902)
```

```
## [1] 4487  805
```

```
rm(dgeIncludedSamples)
```

2 Controllo indipendenza pazienti tra dataset

2.1 GSE68086 vs GSE156902

```
intersect(patientsGSE68086$geo_accession,patientsGSE183635$geo_accession)
```

```
## character(0)
```

Nessun paziente in comune. Per sicurezza si controlli la submission date.

```
table(patientsGSE68086$submission_date)
##
## Apr 21 2015 Jul 10 2015
##         245          40

table(patientsGSE156902$submission_date)
##
## Aug 26 2020 Sep 15 2020
##         598           2

table(patientsGSE183635$submission_date)
##
## Sep 07 2021
##        1646
```

Diversi.

2.2 GSE183635 vs GSE156902

```
intersect(patientsGSE183635$geo_accession,patientsGSE156902$geo_accession)
```

```
## character(0)
```

2.3 GSE68086 vs GSE183635

```
intersect(patientsGSE68086$geo_accession,patientsGSE183635$geo_accession)
```

```
## character(0)
```

3 Analisi dati preliminare

3.1 Ricodificare

Modificare i nomi dei dataset dei patients e delle conte per renderli più facili da utilizzare.

```
patients <- patientsGSE156902
counts <- tep.exprGSE156902.raw
rm(patientsGSE156902,tep.exprGSE156902,tep.exprGSE156902.raw)
```

Modificare i nomi dei pazienti/sample renderli univoci per i due dataset (count e samples).

```
patients$name <- sub(".*\\[(.*?)\\].*", "\\1", patients$title)

colnames(counts) <- sub("^[^-]*-(.*)", "\\1", colnames(counts))
```

3.2 Osservare le patologie per pazienti ed estrarre GBM e HC

```
table(patients$`group:ch1`)
```

```
##
## asymptomaticControls            brainMeta                  GBM
##                  252                  125                  156
##     multipleSclerose
##                   67
```

```
patientsGBM <- patients[patients$`group:ch1` == "GBM",]
patientsGBM <- patientsGBM %>% filter(!str_detect(title, "t1|t2|t3|t4|t5|t6|t7|t8|t9|t10"))

patientsHC <- patients[patients$`group:ch1` == "asymptomaticControls",]

patients.names.GBM <- patientsGBM$name
patients.names.HC <-  patientsHC$name
```

```
countsGBM <- counts[,colnames(counts) %in% patients.names.GBM]
countsHC <- counts[,colnames(counts) %in% patients.names.HC]
```

```
# Per risolvere il problema colnames(counts[colnames(counts) == "Maas.GBM.NICT.035G.TR2170"])
# runnare due volte
colnames(counts[colnames(counts) == "Maas-GBM-NICT-035G-TR2170"])
## [1] "Maas-GBM-NICT-035G-TR2170"   "Maas-GBM-NICT-035G-TR2170.1"
counts <-  counts[,colnames(counts) %in% c(patients.names.HC,patients.names.GBM)]

dim(counts)
## [1] 4487  323
colnames(counts[colnames(counts) == "Maas-GBM-NICT-035G-TR2170"])
## [1] "Maas-GBM-NICT-035G-TR2170"
counts <-  counts[,colnames(counts) %in% c(patients.names.HC,patients.names.GBM)]
dim(counts)
## [1] 4487  322
```

4 Normalizzare

4.1 Normalizzazione CPM

```
# Inizializzare il dataset con la prima iterazione, in questo modo sarà più facile aggiungere i risulta
oggettoDGEList <- DGEList(counts = counts[,1,drop = FALSE])
normalized_counts.test <- cpm(oggettoDGEList,normalized.lib.sizes = F,prior.count= 1 ,log = T)
normalized_counts.test <- as.data.frame(normalized_counts.test)

for (i in 2:ncol(counts)){
  # Estrarre la iesima riga di TEST e farne un oggetto edgeR
  oggettoDGEList <- DGEList(counts = counts[,i,drop = FALSE])
  ## drop = FALSE permette di continuare a considerare l'oggetto come un dataframe e quindi mantenere i

  # Normalizzare la iesima riga di test
  normalized_counts.test_row <- cpm(oggettoDGEList,normalized.lib.sizes = F,prior.count= 1 ,log = T)

  # Rendere la iesima riga normalizzata un dataframe
  normalized_counts.test_row <- as.data.frame(normalized_counts.test_row)

  # Salvare la iesima riga
  normalized_counts.test <- cbind(normalized_counts.test, normalized_counts.test_row)

}
counts <- normalized_counts.test
counts[1:5,1:5]
```

```
##                 Vumc-GBM-306-TR1346 Vumc-GBM-378-TR1347 Vumc-GBM-402-TR1348
## ENSG00000000419            3.402664            3.801715            1.067340
## ENSG00000000938            4.746618            4.879718            4.949983
## ENSG00000001036            3.509579            4.464680            4.237265
## ENSG00000001461            0.702224            6.041181            5.682050
## ENSG00000001629            5.768313            4.216753            3.874695
##                 Vumc-GBM-406-TR1349 Vumc-GBM-408-TR1350
## ENSG00000000419            1.256324            4.862543
## ENSG00000000938            4.343786            6.435122
## ENSG00000001036            4.130793            4.621535
## ENSG00000001461            5.947486            4.100702
## ENSG00000001629            3.194923            4.786594
```

5 Creazione dati di TEST

```
dataframe.formato.classificazione <- function(dataframe){
  conditions <- patients[,c("name","group:ch1")]


  # Trasporre il dataframe delle conte di test, in questo modo si hanno le conte
  # disposte nel modo giusto per la classificazione: pazienti sulle righe e geni sulle colonne
  dataframe <- as.data.frame(t(dataframe))

  # Aggiungere le label y (cancer sano) alle conte
  dataframe <- merge(dataframe,conditions,by.x="row.names",
                                    by.y="name",all = FALSE)

  # Sistemare il dataframe dopo l'operazione di merge, rimettere i rownames al posto giusto
  rownames(dataframe) <- dataframe$Row.names

  # Eliminare la colonna rownames
  dataframe <- dataframe %>% select(-"Row.names")
  dataframe$cancer.type.ch1 <- dataframe$`group:ch1`
  dataframe <- dataframe %>% select(-"group:ch1")

  return(dataframe)
}
```

```
# Mettere nel formato adatto alla classificazione le conte
countsGBM <- dataframe.formato.classificazione(counts)

# Estrarsi il vettore dei vecchi nomi
nuovi.nomi.patologia <- countsGBM$cancer.type.ch1

# Sostituire Glioma con GBM
#nuovi.nomi.patologia <- gsub("Glioma", "GBM", nuovi.nomi.patologia)

# Sostituire Asymptomatic Controls con  HC
nuovi.nomi.patologia <- gsub("asymptomaticControls", "HC", nuovi.nomi.patologia)

# Assegnare la nuova nomenclatura
countsGBM$cancer.type.ch1 <- nuovi.nomi.patologia

table(countsGBM$cancer.type.ch1)
```

```
##
## GBM  HC
##  70 252
```

```
#write.csv(countsGBM,file="/Users/mattia/Desktop/Università/Magistrale/Tesi/Python Data Leakage GSE1836
```