pubh6886_proj

2024-11-13

```
library(dplyr)
```

1. Data Import

```
# RNA-expression count; sample metadata; survival data
counts_raw <- read.table('tcga_brca_counts.tsv', sep = '\t', header = TRUE)
metadata_raw <- read.table('tcga_brca_metadata.tsv', sep = '\t', header = TRUE)
survival <- read.table('tcga_brca_survival.tsv', sep = '\t', header = TRUE)

# Glycoenzyme gene list, glycoenzyme metadata, patient metadata selected col
gene_list <- read.csv('glycoenzyme_gene_list.csv') %>% unlist
gene_metadata <- read.csv('glycoenzyme_genes.csv')
sel_metadata <- read.csv('metadata_colnames.csv') %>% unlist
```

2. Data Wrangling

2.1. Expression Counts Data

```
# Filter counts data to contain only gene names in gene_list
counts <- counts_raw[counts_raw$sample %in% gene_list, ]

# List of glycoenzyme genes unsuccessfully mapped (n = 21)
setdiff(gene_list, counts_raw$sample) %>% sort
```

```
##  [1] "B3GLCT"   "B4GAT1"   "COLGALT1" "COLGALT2" "EOGT"     "GALNT15"
##  [7] "GALNT16"  "GALNT17"  "GALNT18"  "GALNT20"  "GCNT2A"   "GCNT2B"
## [13] "GCNT2C"   "LARGE1"   "LARGE2"   "MGAT4D"   "POGLUT1"  "POMGNT2"
## [19] "RTFDC1"   "UGT2B17"  "XXYLT1"
```

```
# Fix column names (gene) and transpose the count matrix
rownames(counts) <- counts$sample
counts <- counts[ ,-1] %>% t %>% as.data.frame### RUN ONLY ONCE ###

# Fix row names (sampleID)
counts$sampleID <- gsub("\\.", "-", rownames(counts))
counts <- counts[, c("sampleID", colnames(counts)[-183])]### RUN ONLY ONCE ###
counts <- counts[order(counts$sampleID), ] # sort sampleID alphabetically
rownames(counts) <- NULL # remove row names
```

2.2. Patient Metadata

```
# Subset metadata to contain columns of interest
metadata <- metadata_raw[ ,sel_metadata]
```

```r
# Sort sampleID alphabetically
metadata <- metadata[order(metadata$sampleID), ]
rownames(counts) <- NULL
```

2.3. Survival Data

```r
# Remove patient ID column
survival <- survival[ , -2]

# Rename column header "sample" to "sampleID"
colnames(survival)[1] <- "sampleID"

# Sort sampleID alphabetically
survival <- survival[order(survival$sampleID), ]
rownames(survival) <- NULL
```

3. Merging Data

```r
# 2. Merge Data for Modeling
# Merge counts, metadata, and survival on "sampleID"
merged_data <- counts %>%
  inner_join(metadata, by = "sampleID") %>%
  inner_join(survival, by = "sampleID")

# Remove rows where PAM50Call_RNAseq is an empty string
merged_data <- merged_data %>% filter(PAM50Call_RNAseq != "")
```

4. Data Export

```r
write.csv(counts, "./r_output/counts.csv", row.names = FALSE)
write.csv(metadata, "./r_output/metadata.csv", row.names = FALSE)
write.csv(survival, "./r_output/survival.csv", row.names = FALSE)
write.csv(merged_data, "./r_output/merged_data.csv", row.names = FALSE)
```