Metabric EDA

Paul K. Yu

2024-11-19

Exploratory data analysis

Do this once (set eval to TRUE only once)

```
# Install required libraries if not already installed
if (!requireNamespace("pheatmap", quietly = TRUE)) install.packages("pheatmap")
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")
```

Get data

```
library(data.table)

# Get RNA data
epxression <- as.data.frame(fread("metabric/data_mrna_illumina_microarray.txt", header = TRUE, sep = "\t

# Remove NA
epxression <- na.omit(epxression)

# Remove duplicate genes
epxression <- epxression[!duplicated(epxression$Entrez_Gene_Id), ]

# Make gene names as row name
rownames(epxression) <- epxression[[1]]  # Set the first column as row names

# Remove gene name and ID columns
epxression <- epxression[, -c(1, 2)]

n <- ncol(epxression) ## number of subjects
g <- nrow(epxression) ## number of gene features

# Calculate row averages
row_averages <- rowMeans(epxression)

# Determine the cutoff for the lowest 40%
cutoff <- quantile(row_averages, probs = 0.4)

# Filter rows
epxression <- epxression[row_averages > cutoff, ]

# Get outcome data
outcome <- as.data.frame(fread("metabric/brca_metabric_clinical_data.tsv", header = TRUE, sep = "\t"))
```

```r
outcome <- outcome[, c("Patient ID", "Overall Survival Status")]
outcome$`Overall Survival Status` <- as.factor(outcome$`Overall Survival Status`)
levels(outcome$`Overall Survival Status`) <- c(1,2)

# Remove NA
outcome <- na.omit(outcome)
matches <- colnames(epxression) %in% outcome$`Patient ID`
epxression <- epxression[, matches]
matches <- outcome$`Patient ID` %in% colnames(epxression)
outcome <- outcome[matches,]
```

Summary of the dataset

Get summary of outcome data

```r
summary(outcome)
```

```
##   Patient ID        Overall Survival Status
##  Length:1980        1: 837
##  Class :character   2:1143
##  Mode  :character
```

Check for missing values

```r
sum(is.na(epxression))
```

```
## [1] 0
```

```r
sum(is.na(outcome))
```

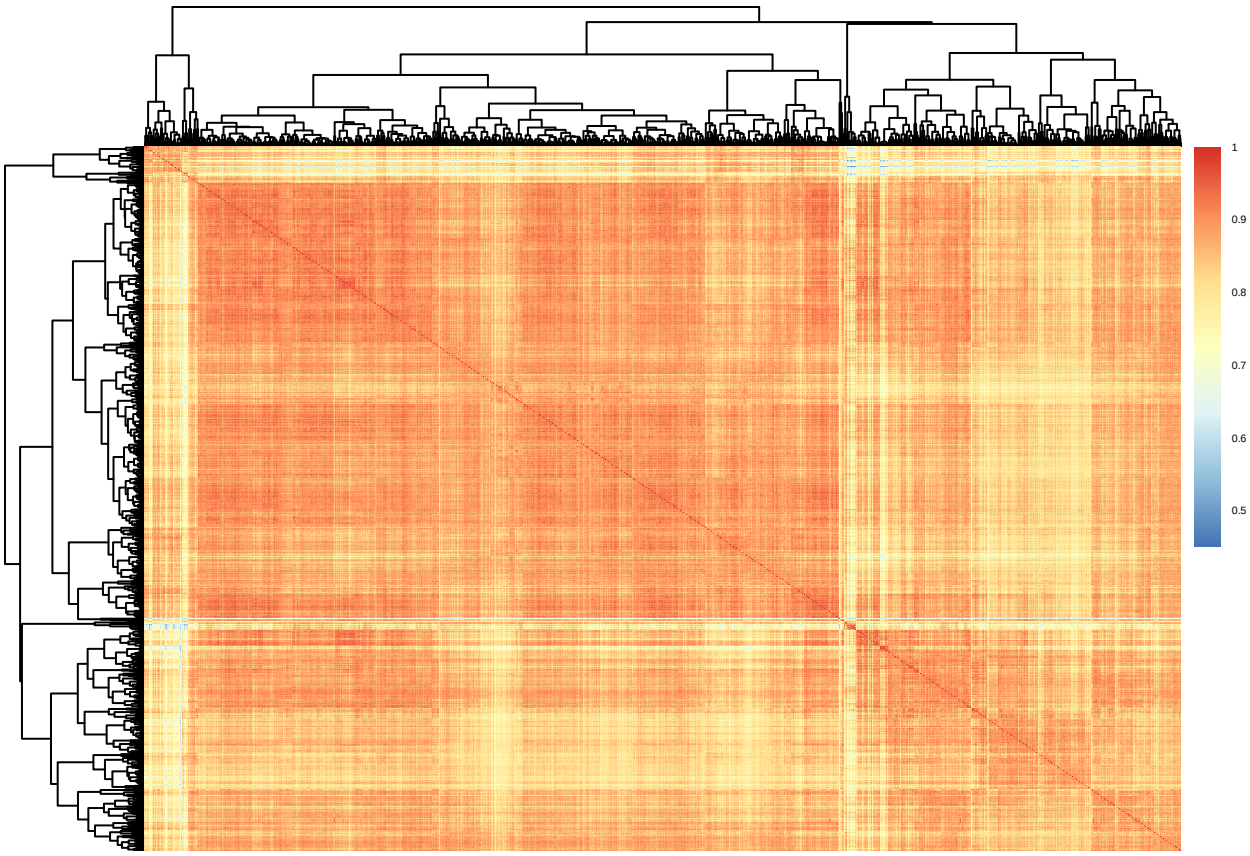
```
## [1] 0
```

Correlation analysis

```r
library(pheatmap)

# Compute correlations
cor_matrix <- cor(epxression, use = "pairwise.complete.obs")

# Visualize correlations as a heatmap
pheatmap(cor_matrix,
         show_rownames = FALSE,
         show_colnames = FALSE,
         clustering_distance_rows = "correlation",
         clustering_distance_cols = "correlation",
         fontsize = 4)
```


1
0.9
0.8
0.7
0.6
0.5