Clustering

Akanksha Garlapati

2024-09-30

### 1. **Load the tissue\_gene\_expression dataset, remove row means, and compute distances**

We load the dataset, remove row means, and compute the distance.

# Load the necessary libraries  
library(dslabs)  
library(tidyverse)

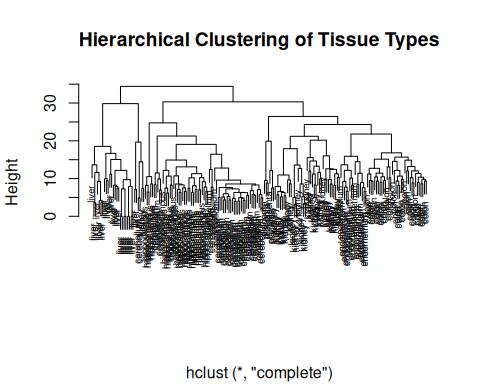
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Load the tissue\_gene\_expression dataset  
data("tissue\_gene\_expression")  
  
# Remove the row means  
x <- sweep(tissue\_gene\_expression$x, 1, rowMeans(tissue\_gene\_expression$x))  
  
# Compute the distance between each observation  
d <- dist(x)

### 2. **Make a hierarchical clustering plot and label the tissue types**

We perform hierarchical clustering and plot it with labels for the tissue types.

# Perform hierarchical clustering  
h <- hclust(d)  
  
# Plot the dendrogram with tissue types as labels  
plot(h, labels = tissue\_gene\_expression$y, main = "Hierarchical Clustering of Tissue Types", xlab = "", cex = 0.6)



### 3. **Run k-means clustering with K = 7 and compare to actual tissue types**

We run k-means clustering with 7 clusters, compare it to the actual tissue types, and repeat the algorithm to observe changes.

# Set K = 7 for k-means clustering  
set.seed(123) # Set seed for reproducibility  
k <- kmeans(x, centers = 7)  
  
# Compare the identified clusters with actual tissue types  
table(k$cluster, tissue\_gene\_expression$y)

##   
## cerebellum colon endometrium hippocampus kidney liver placenta  
## 1 0 0 0 0 0 7 0  
## 2 0 0 0 0 0 17 0  
## 3 31 0 0 0 0 0 0  
## 4 2 0 0 0 2 2 0  
## 5 0 34 15 0 1 0 6  
## 6 5 0 0 31 0 0 0  
## 7 0 0 0 0 36 0 0

# Run the k-means algorithm multiple times  
set.seed(456)  
k\_repeated <- kmeans(x, centers = 7)  
table(k\_repeated$cluster, tissue\_gene\_expression$y)

##   
## cerebellum colon endometrium hippocampus kidney liver placenta  
## 1 0 0 0 0 0 24 0  
## 2 6 0 0 17 0 0 0  
## 3 30 0 0 0 0 0 0  
## 4 0 0 0 14 0 0 0  
## 5 2 0 0 0 2 2 0  
## 6 0 0 0 0 36 0 0  
## 7 0 34 15 0 1 0 6

### 4. **Select the 50 most variable genes, center predictors, and add color bars for tissue types**

We select the 50 most variable genes, center the predictors, and plot a heatmap with color bars for tissue types.

library(matrixStats)

##   
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':  
##   
## count

library(RColorBrewer)  
  
# Select the 50 most variable genes  
sds <- rowSds(tissue\_gene\_expression$x)  
top\_genes <- order(sds, decreasing = TRUE)[1:50]  
  
# Create a subset of data with the 50 most variable genes  
x\_subset <- tissue\_gene\_expression$x[top\_genes, ]  
  
# Center the predictors (rows)  
x\_subset <- sweep(x\_subset, 1, rowMeans(x\_subset))  
  
# Create a color bar based on tissue types  
# Ensure the number of colors matches the number of columns (samples) in x\_subset  
tissue\_colors <- brewer.pal(7, "Set1")[as.numeric(factor(tissue\_gene\_expression$y))]  
  
# Now ensure tissue\_colors matches the number of columns in x\_subset  
# The number of tissue types (samples) should match the number of columns  
if(length(tissue\_colors) != ncol(x\_subset)) {  
 tissue\_colors <- tissue\_colors[1:ncol(x\_subset)]  
}  
  
# Plot the heatmap with color bar  
heatmap(x\_subset, col = brewer.pal(11, "RdBu"), ColSideColors = tissue\_colors, scale = "row")

