Mixed Thermogram Cluster Analysis

Avery Bell

2024-02-26

Introduction

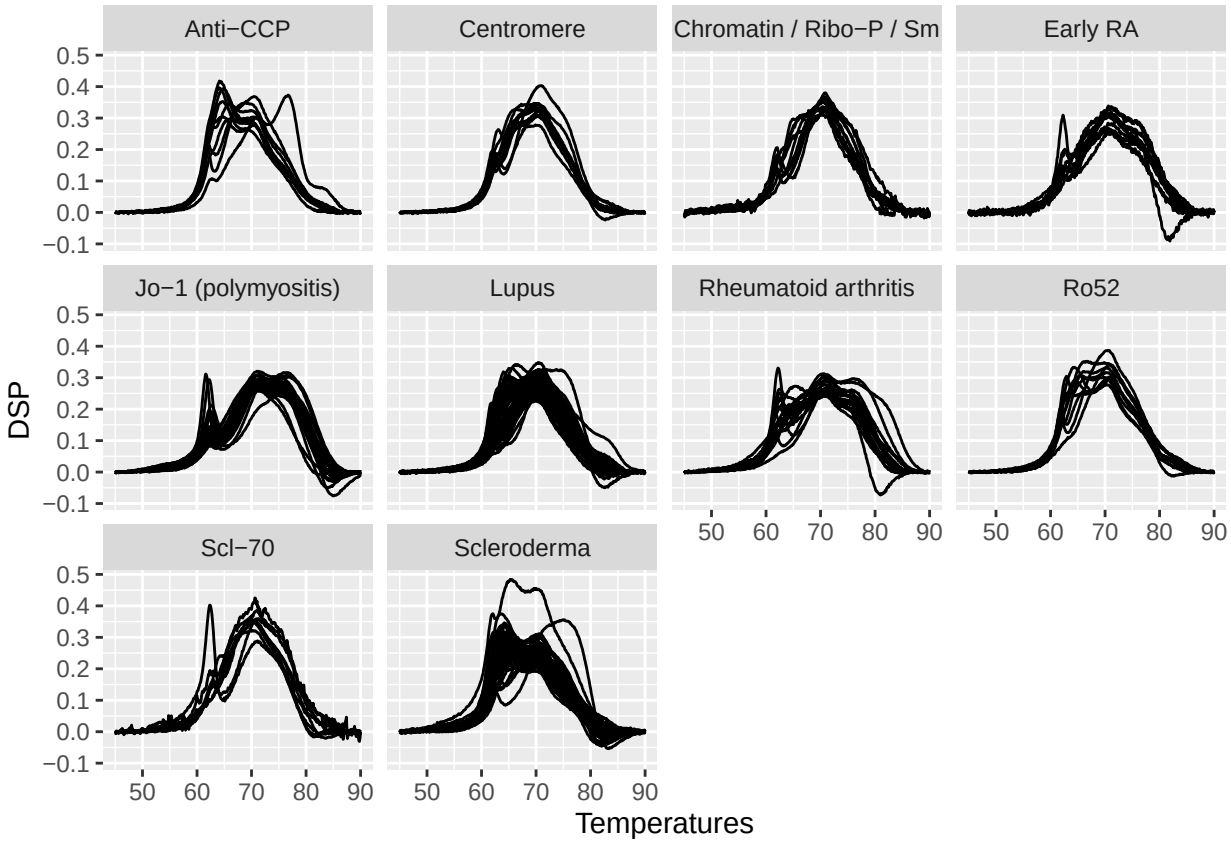
The report represents the results of a cluster analysis on the Fritzler thermograms present in the Mixed Thermogram data set. The goal of this analysis was to utilize unsupervised techniques to determine if thermogram characteristics differ by disease.Table 1 represents the diseases and thermogram frequencies included in this analysis.

Table 1: Disease Groups and Frequencies

DiseaseGroup	Count
Anti-CCP	10
Centromere	10
Chromatin / Ribo-P / Sm	10
Early RA	10
Jo-1 (polymyositis)	25
Lupus	50
Rheumatoid arthritis	18
Ro52	10
Scl-70	9
Scleroderma	50

It can be seen that a broad range of diseases are present in this analysis with a mix of frequencies, ranging from 9 - 50. Figure 1 plots each thermogram curve according to it's disease group. It can be seen that the thermograms differ by disease, but also contain variability in presentation within a disease. This may impact a clustering algorithm's ability to identify differences between curves that are distinct across diseases.

Figure 1: Thermograms by Diesease Group



Methods

Divisive and agglomerative hierarchical clustering was used to analyze these thermogram curves. Distances between samples were calculated using Gower Distance. The specific excess heat capacities observed from 50 - 80 C were used to train the models. The average silhouette width was calculated for k = 2,. . . ,10 and was used to evaluate cluster . The values of k = 2, . . . , 10 were uses because 2 is the minimum number of clusters that can be used in an analysis, and there are 10 disease groups present in the data used. The values for k for the divisive and agglomerative method with the lowest average silhouette width observed were used were explored and visualized.

Results

The results for k = 2, . . . , 10 for the divisive and agglomerative analysis are shown in table 2. For agglomerative clustering, the ideal number of clusters is 5, and for divisive clustering the ideal number of clusters is 6.

Table 2: Clustering Metrics Results

k	method	silhouette
5	agglomerative	0.1399324
7	agglomerative	0.1881143
6	divisive	0.1904221
5	divisive	0.1941881
10	divisive	0.1972642

Table 3 and figure 2 represent the results for the top agglomerative results. Table 3 demonstrates that there is a large mixture of different diseases in clusters 1 and 3, and sparse clusters in groups 2, 4, and 5. Figure 2 shows that the thermograms have been grouped by shape. There is a potential outlier in the data present in in cluster 4, as the curve so so distinct that it is grouped on its own.

Table 3: Porportions of each Disease across Agglomerative Clusters

	Clusters				
Disease	1	2	3	4	5
Anti-CCP	0.90	0.00	0.10	0.00	0.00
Centromere	0.90	0.00	0.00	0.00	0.10
Chromatin / Ribo-P / Sm	0.80	0.00	0.00	0.00	0.20
Early RA	0.30	0.00	0.70	0.00	0.00
Jo-1 (polymyositis)	0.00	0.04	0.88	0.00	0.08
Lupus	0.70	0.00	0.30	0.00	0.00
Rheumatoid arthritis	0.00	0.28	0.72	0.00	0.00
Ro52	0.90	0.10	0.00	0.00	0.00
Scl-70	0.56	0.11	0.00	0.00	0.33
Scleroderma	0.78	0.10	0.10	0.02	0.00

Figure 2: Top Agglomerative Result

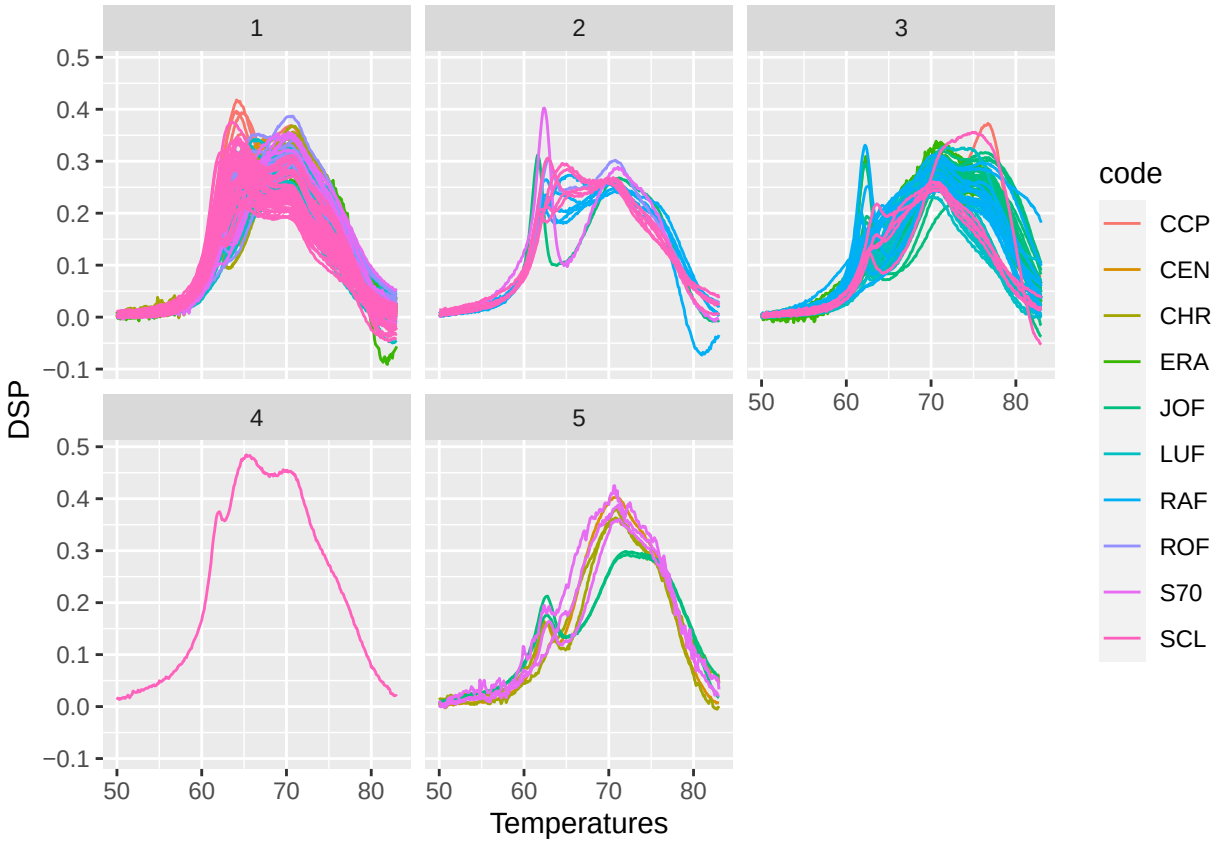
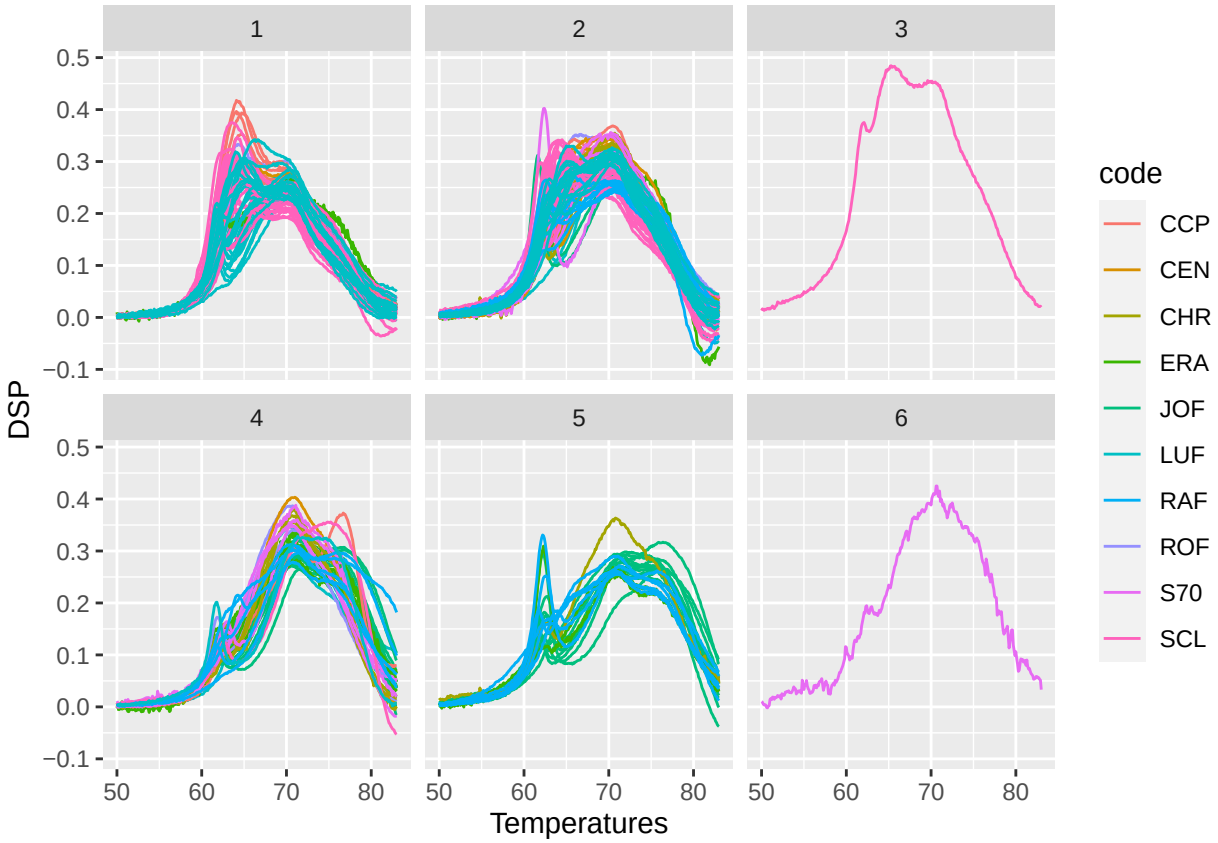


Table 4 and figure 3 represent the results for the top divisive results. Table 4 demonstrates that there is a large mixture of different diseases in clusters 1, 2, 4, and 5 and sparse clusters in groups 3 and 6. Figure 3 shows that the thermograms have been grouped by shape. There are potential outliers in the data present in in clusters 3 and 6, with the curve in cluster 3 also being grouped by itself in the agglomerative clustering.

Table 4: Porportions of each Disease across Divisive Clusters

	Clusters					
Disease	1	2	3	4	5	6
Anti-CCP	0.50	0.40	0.00	0.10	0.00	0.00
Centromere	0.10	0.80	0.00	0.10	0.00	0.00
Chromatin / Ribo-P / Sm	0.00	0.60	0.00	0.30	0.10	0.00
Early RA	0.20	0.10	0.00	0.50	0.20	0.00
Jo-1 (polymyositis)	0.00	0.08	0.00	0.48	0.44	0.00
Lupus	0.54	0.38	0.00	0.08	0.00	0.00
Rheumatoid arthritis	0.00	0.39	0.00	0.28	0.33	0.00
Ro52	0.20	0.60	0.00	0.20	0.00	0.00
Scl-70	0.00	0.44	0.00	0.44	0.00	0.11
Scleroderma	0.56	0.40	0.02	0.02	0.00	0.00

figure 3: top divisive result



Unsupervised methods are sensitive to the data they are trained on. Due to this, the two outlier curves present in the above clusterings may have a strong effect on the results. The above methodology was repeated after removing those two thermograms.

Removing outlier curves

table 4: Reclustering Metrics Evaluation

k	method	silhouette
7	agglomerative	0.1675227
5	agglomerative	0.1706187
9	agglomerative	0.1730010
8	agglomerative	0.1758041
4	divisive	0.1776854

figure 4: top agglomerative result after reclustering

```
##                            Clusters
## Disease                       1    2    3    4    5    6    7
##   Anti-CCP                 0.60 0.00 0.10 0.30 0.00 0.00 0.00
##   Centromere               0.10 0.00 0.30 0.60 0.00 0.00 0.00
##   Chromatin / Ribo-P / Sm  0.00 0.00 0.00 0.90 0.10 0.00 0.00
##   Early RA                 0.00 0.00 0.70 0.10 0.00 0.20 0.00
##   Jo-1 (polymyositis)      0.00 0.00 0.20 0.04 0.52 0.16 0.08
##   Lupus                    0.34 0.00 0.40 0.24 0.02 0.00 0.00
##   Rheumatoid arthritis     0.00 0.11 0.44 0.00 0.11 0.28 0.06
##   Ro52                     0.20 0.10 0.20 0.50 0.00 0.00 0.00
##   Scl-70                   0.00 0.00 0.00 0.75 0.12 0.12 0.00
##   Scleroderma              0.49 0.29 0.08 0.12 0.02 0.00 0.00
```

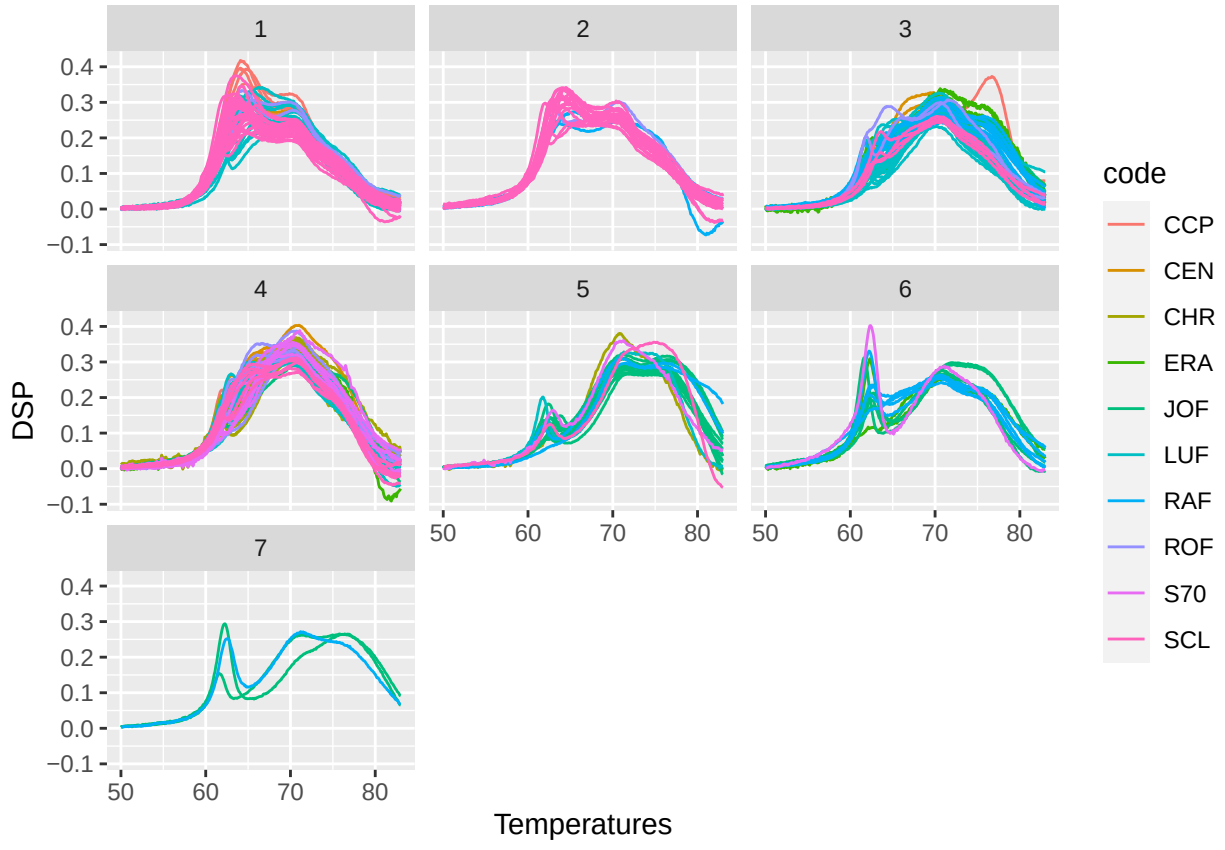
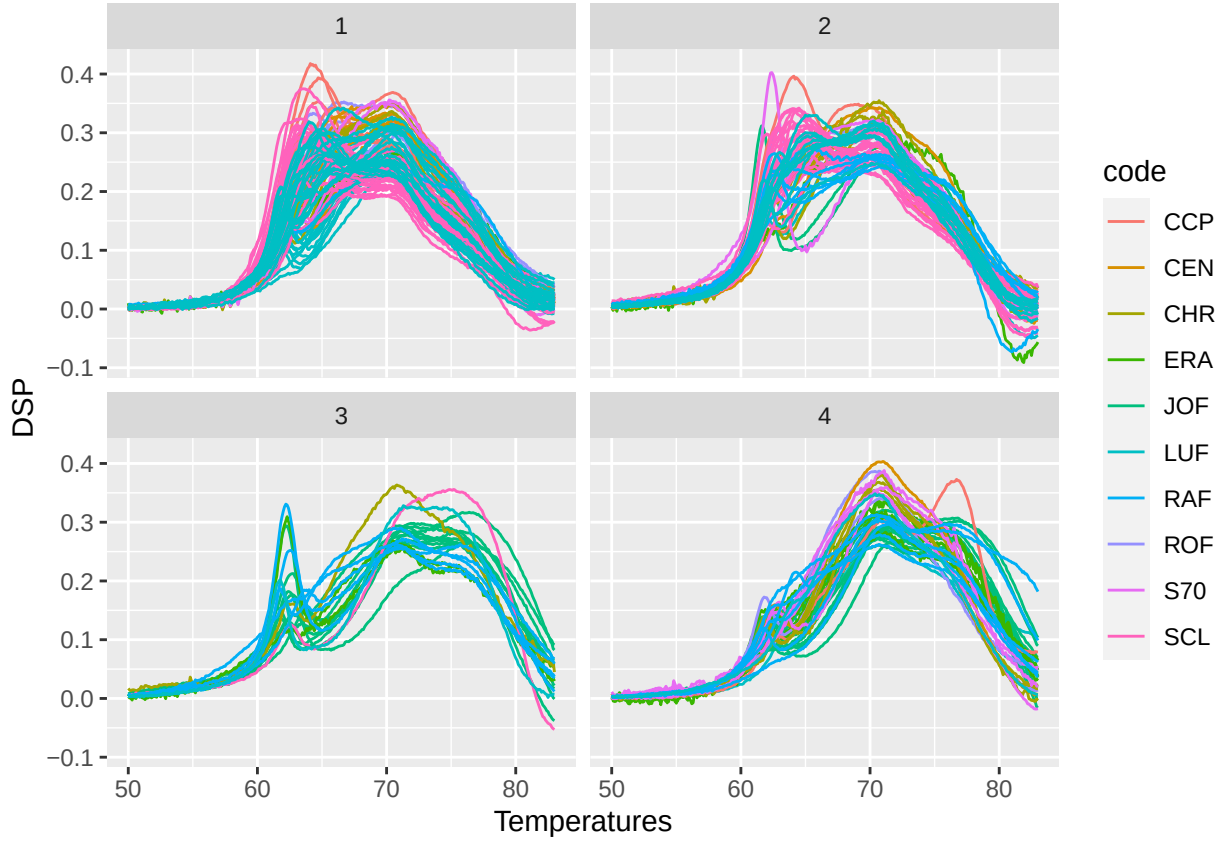


figure 5: top divisive result after reclustering

```
##                            Clusters
## Disease                       1    2    3    4
```

```
##    Anti-CCP                0.60 0.30 0.00 0.10
##    Centromere              0.60 0.30 0.00 0.10
##    Chromatin / Ribo-P / Sm 0.30 0.30 0.10 0.30
##    Early RA                0.20 0.10 0.20 0.50
##    Jo-1 (polymyositis)     0.00 0.08 0.40 0.52
##    Lupus                   0.72 0.20 0.02 0.06
##    Rheumatoid arthritis    0.06 0.33 0.28 0.33
##    Ro52                    0.60 0.20 0.00 0.20
##    Scl-70                  0.25 0.25 0.00 0.50
##    Scleroderma             0.61 0.37 0.02 0.00
```



Chi Square Test ???

```
## Warning in chisq.test(contingency_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 119.6, df = 27, p-value = 1.27e-13
```

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(tidyverse)
library(clinicalDSCdata)
```

```r
library(fpc)
library(cluster)
library(factoextra)
library(ClusterR)

codes <- c('SCL', 'RAF', 'JOF', 'ERA', 'ROF', 'CCP', 'CEN', 'CHR', 'S70', 'LUF') #LUF


cluster_df <- MixedThermograms %>%
  filter(code %in% codes)

graph_df <- cluster_df

counts <- cluster_df %>%
  group_by(DiseaseGroup) %>%
  summarize(Count = n())

knitr::kable(counts)
mixed_long <- pivot_longer(
                           cluster_df,
                           cols = T45:T90,
                           names_to = "Temperatures",
                           values_to = "DSP"
                            )

mixed_long <- mixed_long %>%
  mutate(Temperatures = as.numeric(str_remove(Temperatures, "T")))

ggplot(mixed_long, aes(x = Temperatures))+
  geom_line(aes(y = DSP, group = sampleID))+
  facet_wrap( . ~ DiseaseGroup )

cluster_df <- MixedThermograms %>%
  filter(code %in% codes)%>%
  select(c(3, T50:T83))

labs <- MixedThermograms %>%
  filter(code %in% codes)%>%
  select(DiseaseGroup)

labs_1 <- as.factor(labs$DiseaseGroup)


labs <- as.numeric(as.factor(labs$DiseaseGroup))

cluster_df <- column_to_rownames(cluster_df, 'sampleID')

# cluster_df_1_1 <- t(cluster_df)
#
# cluster_df_1_2 <- diff(cluster_df_1_1)
#
# cluster_df_1 <- as.data.frame(t(cluster_df_1_2))
```

```r
gower.dist <- daisy(cluster_df, metric = 'gower')
#class(gower.dist)

# Using "complete" linkage - agglomerative
agg_clust_c <- hclust(gower.dist, method = "complete")

# Using "complete" linkage - divisive
divisive_clust <- diana(as.matrix(gower.dist),
                        diss = TRUE, keep.diss = TRUE)


result_df <- data.frame(k = numeric(), purity = numeric(), silhouette = numeric(), twss = numeric())

for (k in 2:length(unique(labs))) {

  # Agglomerative clustering
  clusters_agg <- cutree(agg_clust_c, k = k)

  # Divisive clustering
  clusters_div <- cutree(divisive_clust, k = k)

  # Calculate cluster statistics for agglomerative clustering
  cluster_stats_agg <- cluster.stats(gower.dist, clusters_agg)

  # Calculate purity for agglomerative clustering
  purity_result_agg <- external_validation(clusters_agg, labs, method = "purity")

  # Calculate silhouette for agglomerative clustering
  silhouette_result_agg <- cluster_stats_agg$avg.silwidth

  # Calculate TWSS for agglomerative clustering
  twss_result_agg <- cluster_stats_agg$within.cluster.ss

  # Calculate cluster statistics for divisive clustering
  cluster_stats_div <- cluster.stats(gower.dist, clusters_div)

  # Calculate purity for divisive clustering
  purity_result_div <- external_validation(clusters_div, labs, method = "purity")

  # Calculate silhouette for divisive clustering
  silhouette_result_div <- cluster_stats_div$avg.silwidth

  # Calculate TWSS for divisive clustering
  twss_result_div <- cluster_stats_div$within.cluster.ss

  # Combine all results into a single dataframe for agglomerative clustering
  cluster_results_agg <- data.frame(k = k, method = "agglomerative",
                                    purity = purity_result_agg, silhouette = silhouette_result_agg,
                                    twss = twss_result_agg)

  # Combine all results into a single dataframe for divisive clustering
  cluster_results_div <- data.frame(k = k, method = "divisive",
                                    purity = purity_result_div, silhouette = silhouette_result_div,
```

```
                                    twss = twss_result_div)

  # Append the results to result_df
  result_df <- rbind(result_df, cluster_results_agg, cluster_results_div)

}



rclusters <- cutree(agg_clust_c, k = 9)
# Create a contingency table
contingency_table <- table(Disease = labs_1, Clusters = rclusters)

# Calculate proportions
proportions <- prop.table(contingency_table, margin = 1)
result_df %>%
  select(k, method, silhouette) %>%
  arrange(silhouette) %>%
  head(5) %>%
  knitr::kable()
groups <- 5
rclusters <- cutree(agg_clust_c, k = groups)
# Create a contingency table
contingency_table <- table(Disease = labs_1, Clusters = rclusters)

# Calculate proportions
round(prop.table(contingency_table, margin = 1), digits = 2)

clust_graph <- cbind(graph_df, rclusters)

clust_graph <- rownames_to_column(clust_graph, "sample_id")

clust_graph_1 <- pivot_longer(
                           clust_graph,
                           cols = T50:T83,
                           names_to = "Temperatures",
                           values_to = "DSP"
                            )

clust_graph_2 <- clust_graph_1 %>%
  mutate(Temperatures = as.numeric(str_remove(Temperatures, "T")))

ggplot(clust_graph_2, aes(x = Temperatures))+
  geom_line(aes(y = DSP, group = sample_id, color = code))+
  facet_wrap( . ~ rclusters )




groups <- 6
rclusters <- cutree(divisive_clust, k = groups)
# Create a contingency table
contingency_table <- table(Disease = labs_1, Clusters = rclusters)
```

```r
# Calculate proportions
round(prop.table(contingency_table, margin = 1), digits = 2)

clust_graph <- cbind(graph_df, rclusters)

clust_graph <- rownames_to_column(clust_graph, "sample_id")

clust_graph_1 <- pivot_longer(
                            clust_graph,
                            cols = T50:T83,
                            names_to = "Temperatures",
                            values_to = "DSP"
                             )

clust_graph_2 <- clust_graph_1 %>%
  mutate(Temperatures = as.numeric(str_remove(Temperatures, "T")))

ggplot(clust_graph_2, aes(x = Temperatures))+
  geom_line(aes(y = DSP, group = sample_id, color = code))+
  facet_wrap( . ~ rclusters )



cluster_df <- MixedThermograms %>%
  filter(sampleID != 'SCL25' & sampleID != 'S70F5') %>%
  filter(code %in% codes)%>%
  select(c(3, T50:T83))

graph_df <- MixedThermograms %>%
  filter(sampleID != 'SCL25' & sampleID != 'S70F5') %>%
  filter(code %in% codes)


labs <- MixedThermograms %>%
  filter(sampleID != 'SCL25' & sampleID != 'S70F5') %>%
  filter(code %in% codes)%>%
  select(DiseaseGroup)

labs_1 <- as.factor(labs$DiseaseGroup)


labs <- as.numeric(as.factor(labs$DiseaseGroup))

cluster_df <- column_to_rownames(cluster_df, 'sampleID')

# cluster_df_1_1 <- t(cluster_df)
#
# cluster_df_1_2 <- diff(cluster_df_1_1)
#
# cluster_df_1 <- as.data.frame(t(cluster_df_1_2))

gower.dist <- daisy(cluster_df, metric = 'gower')
#class(gower.dist)
```

```r
# Using "complete" linkage - agglomerative
agg_clust_c <- hclust(gower.dist, method = "complete")

# Using "complete" linkage - divisive
divisive_clust <- diana(as.matrix(gower.dist),
                        diss = TRUE, keep.diss = TRUE)


result_df <- data.frame(k = numeric(), purity = numeric(), silhouette = numeric(), twss = numeric())

for (k in 2:length(unique(labs))) {

  # Agglomerative clustering
  clusters_agg <- cutree(agg_clust_c, k = k)

  # Divisive clustering
  clusters_div <- cutree(divisive_clust, k = k)

  # Calculate cluster statistics for agglomerative clustering
  cluster_stats_agg <- cluster.stats(gower.dist, clusters_agg)

  # Calculate purity for agglomerative clustering
  purity_result_agg <- external_validation(clusters_agg, labs, method = "purity")

  # Calculate silhouette for agglomerative clustering
  silhouette_result_agg <- cluster_stats_agg$avg.silwidth

  # Calculate TWSS for agglomerative clustering
  twss_result_agg <- cluster_stats_agg$within.cluster.ss

  # Calculate cluster statistics for divisive clustering
  cluster_stats_div <- cluster.stats(gower.dist, clusters_div)

  # Calculate purity for divisive clustering
  purity_result_div <- external_validation(clusters_div, labs, method = "purity")

  # Calculate silhouette for divisive clustering
  silhouette_result_div <- cluster_stats_div$avg.silwidth

  # Calculate TWSS for divisive clustering
  twss_result_div <- cluster_stats_div$within.cluster.ss

  # Combine all results into a single dataframe for agglomerative clustering
  cluster_results_agg <- data.frame(k = k, method = "agglomerative",
                                    purity = purity_result_agg, silhouette = silhouette_result_agg,
                                    twss = twss_result_agg)

  # Combine all results into a single dataframe for divisive clustering
  cluster_results_div <- data.frame(k = k, method = "divisive",
                                    purity = purity_result_div, silhouette = silhouette_result_div,
                                    twss = twss_result_div)

  # Append the results to result_df
```

```
  result_df <- rbind(result_df, cluster_results_agg, cluster_results_div)

}
result_df %>%
  select(k, method, silhouette) %>%
  arrange(silhouette) %>%
  head(5) %>%
  knitr::kable()
groups <- 7
rclusters <- cutree(agg_clust_c, k = groups)
# Create a contingency table
contingency_table <- table(Disease = labs_1, Clusters = rclusters)

# Calculate proportions
round(prop.table(contingency_table, margin = 1), digits = 2)


clust_graph <- cbind(graph_df, rclusters)

clust_graph <- rownames_to_column(clust_graph, "sample_id")

clust_graph_1 <- pivot_longer(
                           clust_graph,
                           cols = T50:T83,
                           names_to = "Temperatures",
                           values_to = "DSP"
                            )

clust_graph_2 <- clust_graph_1 %>%
  mutate(Temperatures = as.numeric(str_remove(Temperatures, "T")))

ggplot(clust_graph_2, aes(x = Temperatures))+
  geom_line(aes(y = DSP, group = sample_id, color = code))+
  facet_wrap( . ~ rclusters )



groups <- 4
rclusters <- cutree(divisive_clust, k = groups)
# Create a contingency table
contingency_table <- table(Disease = labs_1, Clusters = rclusters)

# Calculate proportions
round(prop.table(contingency_table, margin = 1), digits = 2)


clust_graph <- cbind(graph_df, rclusters)

clust_graph <- rownames_to_column(clust_graph, "sample_id")

clust_graph_1 <- pivot_longer(
                           clust_graph,
                           cols = T50:T83,
```

```
                          names_to = "Temperatures",
                          values_to = "DSP"
                           )

clust_graph_2 <- clust_graph_1 %>%
  mutate(Temperatures = as.numeric(str_remove(Temperatures, "T")))

ggplot(clust_graph_2, aes(x = Temperatures))+
  geom_line(aes(y = DSP, group = sample_id, color = code))+
  facet_wrap( . ~ rclusters )



# Create a contingency table
contingency_table <- table(labs, rclusters)

# Perform chi-square test
chi_square_test <- chisq.test(contingency_table)

# Print the results
print(chi_square_test)
```