Preliminaries

The `CircadianTools` package was loaded first. The ANOVA and t-test filtered datasets from the previous chapter were read in. Both datasets were scaled and centered.

```
library(CircadianTools)
a.filter <- read.csv(
  "~/MEGA/Uni/Masters/Diss/Stats/R Markdown/filtering/a_filter.csv",
  stringsAsFactors=FALSE)
a.filter <-GeneScale(a.filter)
t.filter <- read.csv(
  "~/MEGA/Uni/Masters/Diss/Stats/R Markdown/filtering/t_filter.csv",
  stringsAsFactors=FALSE)
t.filter <-GeneScale(t.filter)
```

The seed was set to 123 for reproducibility purposes.

```
set.seed(123)
```

Cluster Validation

Plots of cluster validation metric scores against k (the number of clusters in the partition) were plotted for PAM, agglomerative hierarchical clustering and DIANA for the ANOVA and t-test filtered datasets. Both Euclidean distance and absolute Pearson's correlation were considered.

Due to the huge amount of computational power required to create the cluster validation plots, the plots were generated using Google Cloud virtual machines with 24 cores. As a result, the cluster validation code will not be run again for this appendix. However, the code used has been included.

The values of k considered were first created.

```
k.params <- c(seq(2,5), seq(10, 300, 5))
```

ANOVA Filtered with Euclidean Distance

```
# Use the maximum number of processor threads for calculations
nthreads <- parallel::detectCores()

ClusterParamSelection(a.filter, k = k.params, metric="euclidean",
                      nthreads = nthreads, path = "anova_euclid_validation")
```

ANOVA Filtered with Absolute Pearson's Correlation

```
ClusterParamSelection(a.filter, k = k.params, metric="abs.correlation",
                      nthreads = nthreads, path = "anova_abscor_validation")
```

T-Test Filtered with Euclidean Distance

```
ClusterParamSelection(t.filter, k = k.params, metric="euclidean",
                      nthreads = nthreads, path = "ttest_euclid_validation")
```

T-Test Filtered with Absolute Pearson's Correlation

```
ClusterParamSelection(t.filter, k = k.params, metric="abs.correlation",
                      nthreads = nthreads, path = "ttest_abscor_validation")
```

Generating the Clusters For Use in Future Methods

After consulting the cluster validation plots, k = 95 was decided to be the best value to use. Clustering partitions were then generated and saved to csv files.

ANOVA Filtered

The distance matrices using either Euclidean distance or absolute Pearson's Correlation were calculated for the ANOVA filtered dataset.

```
a.euc <- DistanceGen(a.filter, metric = "euclidean")
a.cor <- DistanceGen(a.filter, metric = "abs.correlation")
```

PAM Clustering

```
system.time(a.pam.euc <- PamClustering(distance = a.euc, k = 95, scale = TRUE)
            )
```

```
##     user   system  elapsed
## 4877.809    0.418 4888.552
```

```
a.pam.euc <- cbind(a.filter, cluster = a.pam.euc$cluster)
write.csv(a.pam.euc, file = "a_pam_euc.csv", row.names = FALSE)
a.pam.cor <- PamClustering(distance = a.cor, k = 95, scale = TRUE)
a.pam.cor <- cbind(a.filter, cluster = a.pam.cor$cluster)
write.csv(a.pam.cor, file = "a_pam_cor.csv", row.names = FALSE)
```

Agglomerative Hierarchical Clustering

```
system.time(a.agglom.euc <- AgglomClustering(distance = a.euc, k = 95,
                                             scale = TRUE)
            )
```

```
##    user  system elapsed
##   2.032   0.155   7.634
```

```
a.agglom.euc <- cbind(a.filter, cluster = a.agglom.euc$cluster)
write.csv(a.agglom.euc, file = "a_agglom_euc.csv", row.names = FALSE )
a.agglom.cor <- AgglomClustering(distance = a.cor, k = 95, scale = TRUE)
a.agglom.cor <- cbind(a.filter, cluster = a.agglom.cor$cluster)
write.csv(a.agglom.cor, file = "a_agglom_cor.csv", row.names = FALSE)
```

DIANA Clustering

```
system.time(a.diana.euc <- DianaClustering(distance = a.euc, k = 95,
                                           scale = TRUE)
                  )
```

```
##    user  system elapsed
## 628.248   0.342 629.667
```

```
a.diana.euc <- cbind(a.filter, cluster = a.diana.euc$cluster)
write.csv(a.diana.euc, file = "a_diana_euc.csv", row.names = FALSE)
a.diana.cor <- DianaClustering(distance = a.cor, k = 95, scale = TRUE)
a.diana.cor <- cbind(a.filter, cluster = a.diana.cor$cluster)
write.csv(a.diana.cor, file = "a_diana_cor.csv", row.names = FALSE)
```

T-Test Filtered

Similar calculations were then run for t-test filtered dataset. The distance matrices for Euclidean distance and absolute Pearson's Correlation were calculated.

```
t.euc <- DistanceGen(t.filter, metric = "euclidean")
t.cor <- DistanceGen(t.filter, metric = "abs.correlation")
```

PAM

```
t.pam.euc <- PamClustering(distance = t.euc, k = 95, scale = TRUE)
t.pam.euc <- cbind(t.filter, cluster = t.pam.euc$cluster)
write.csv(t.pam.euc, file = "t_pam_euc.csv", row.names = FALSE)
t.pam.cor <- PamClustering(distance = t.cor, k = 95, scale = TRUE)
t.pam.cor <- cbind(t.filter, cluster = t.pam.cor$cluster)
write.csv(t.pam.cor, file = "t_pam_cor.csv", row.names = FALSE)
```

Agglomerative Hierarchical

```
t.agglom.euc <- AgglomClustering(distance = t.euc, k = 95, scale = TRUE)
t.agglom.euc <- cbind(t.filter, cluster = t.agglom.euc$cluster)
write.csv(t.agglom.euc, file = "t_agglom_euc.csv", row.names = FALSE)
t.agglom.cor <- AgglomClustering(distance = t.cor, k = 95, scale = TRUE)
t.agglom.cor <- cbind(t.filter, cluster = t.agglom.cor$cluster)
write.csv(t.agglom.cor, file = "t_agglom_cor.csv", row.names = FALSE)
```

DIANA

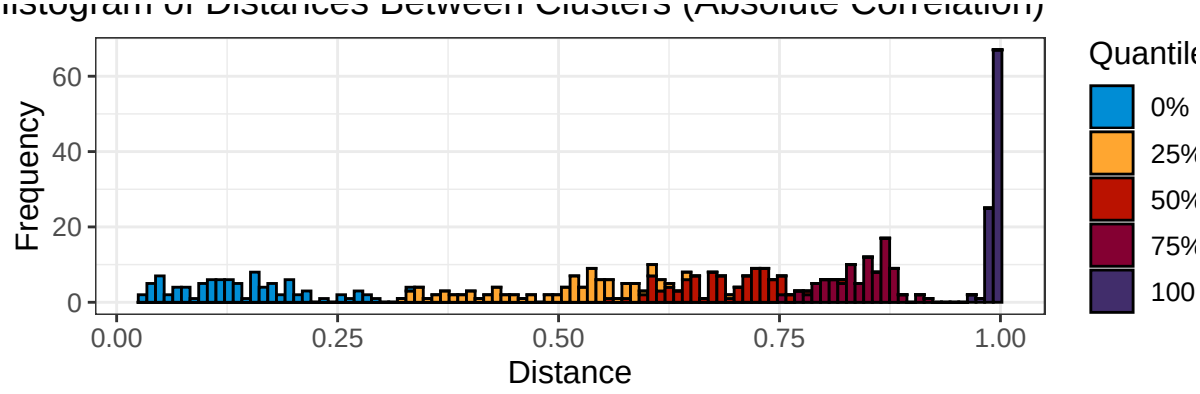
```
t.diana.euc <- DianaClustering(distance = t.euc, k = 95,
                                          scale = TRUE)
t.diana.euc <- cbind(t.filter, cluster = t.diana.euc$cluster)
write.csv(t.diana.euc, file = "t_diana_euc.csv", row.names = FALSE)
t.diana.cor <- DianaClustering(distance = t.cor, k = 95, scale = TRUE)
t.diana.cor <- cbind(t.filter, cluster = t.diana.cor$cluster )
write.csv(t.diana.cor, file = "t_diana_cor.csv", row.names = FALSE)
```

Comparing Distance Metrics

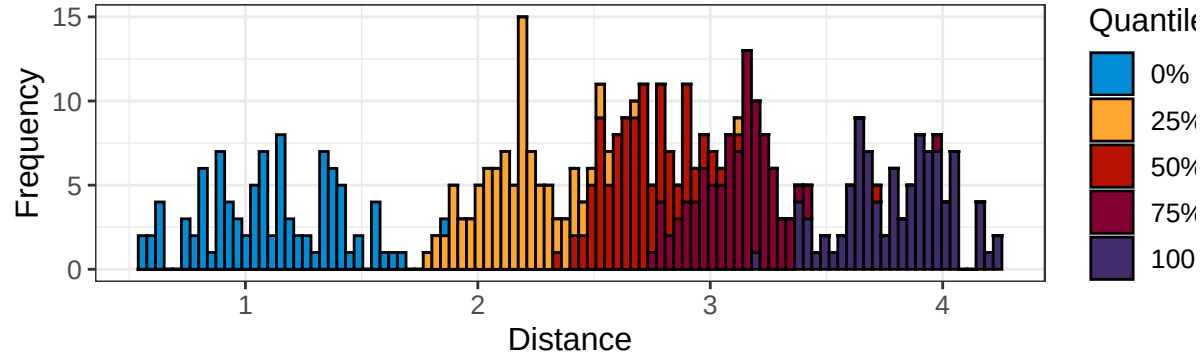
In order to compare the effect of using different distance metrics, the distances between the centers of clusters generated via using the two metrics were plotted used histograms.

```
quantiles <- CircadianTools::FindClusterDistanceQuantiles(
  cluster.dataset = a.agglom.cor, metric = "abs.correlation",
  nthreads = 12)

quantiles.plot <- reshape2::melt(data = quantiles,
                                 measure.vars = c("0%", "25%", "50%", "75%", "100%")
                                 )
colnames(quantiles.plot) <- c("results", "Quantile", "Distance")
# Vector of colours used in package
colours.vector <- c("#008dd5", "#ffa630", "#ba1200", "#840032", "#412d6b")

p1 <- ggplot2::ggplot(data = quantiles.plot, ggplot2::aes(x = Distance,
                                                          fill = Quantile)
                      )
p1 <- p1 + ggplot2::geom_histogram(color = "black", bins = 100)
```

```
p1 <- p1 + ggplot2::scale_fill_manual(values = colours.vector)
p1 <- p1 + ggplot2::theme_bw() + ggplot2::ylab("Frequency")
p1 <- p1 + ggplot2::ggtitle("Histogram of Distances Between Clusters (Absolute Correlation)")
p1 <- p1 + ggplot2::theme(plot.title = ggplot2::element_text(hjust = 1))
p1 <- p1 + ggplot2::theme(text = ggplot2::element_text(size = 12))




quantiles <- CircadianTools::FindClusterDistanceQuantiles(
  cluster.dataset = a.agglom.euc, metric = "euclidean", nthreads = 12)

quantiles.plot <- reshape2::melt(data = quantiles,
                                 measure.vars = c("0%", "25%", "50%",
                                                  "75%", "100%")
                                 )
colnames(quantiles.plot) <- c("results", "Quantile", "Distance")


p2 <- ggplot2::ggplot(data = quantiles.plot, ggplot2::aes(x = Distance,
                                                          fill = Quantile)
                      )
p2 <- p2 + ggplot2::geom_histogram(color = "black", bins = 100)
p2 <- p2 + ggplot2::scale_fill_manual(values = colours.vector)
p2 <- p2 + ggplot2::theme_bw() + ggplot2::ylab("Frequency")
p2 <- p2 +  ggplot2::ggtitle("Histogram of Distances Between Clusters (Euclidean Distance)")
p2 <- p2 + ggplot2::theme(plot.title = ggplot2::element_text(hjust = 1))
p2 <- p2 + ggplot2::theme(text = ggplot2::element_text(size = 12))

p <- gridExtra::grid.arrange(p1,p2, nrow = 2)
```



Histogram of Distances Between Clusters (Euclidean Distance)




```
ggplot2::ggsave("Quantile_comp.png", plot = p, width = 8.27,
                height = 11.69, units = "in")  # Save the plot
```