K-means Clustering

R script for k-means clustering

This is a markdown document to accompany the paper describing the clustering of H7 into lineages using k-means clustering.

Firstly we need to clear R to remove any old data and then we need to include the libraries needed for calculating the nucleotide distance matrix (ape), the bootstrap clustering (kmed) and for plotting advanced graphics (ggplot2)

```
rm(list=ls())
## Library attachment
library(kmed)
```

```
## Warning: package 'kmed' was built under R version 3.5.3
```

```
library(ape)
```

```
## Warning: package 'ape' was built under R version 3.5.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

In this case there were some warnings about the version build but these do not affect the results.

The next chunk of code reads in the sequence data and calculates the nucleotide distance matrix using the defauly Kitano 1980 evolutionary model.

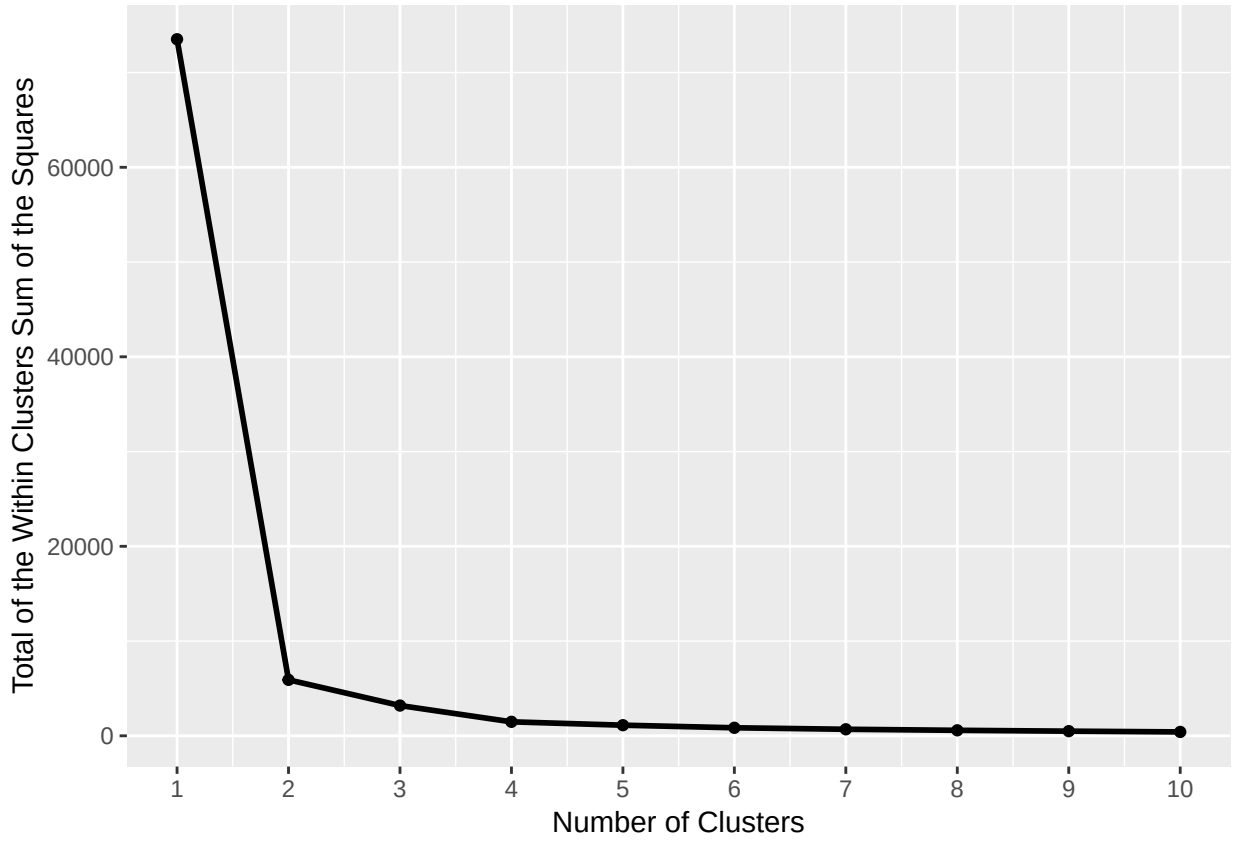
```
x <- read.FASTA("H7_HA_IRDB_2019_1_14_aligned_muscle_cleaned.fas", type="DNA")
y <- dist.dna(x)
```

Plotting the total within cluster sum of the squares of the difference

The next code segment calculates the total within cluster sum of the squares of the difference for the range of 1 to 10 clusters. This graph can be used to decide on the optimum number of clusters for the clustering.The resulting data is then plotted with ggplot2.

```
wss <- vector()
for (i in 1:10) wss[i] <- sum(kmeans(y, iter.max=1000, nstart=100,
                                     centers=i)$tot.withinss)
Clusters <- c(1:10)

wss_data <- data.frame(cbind(Clusters,wss))
ggplot(data=wss_data, aes(x=Clusters, y=wss))+
  geom_line(size=1)+
  geom_point(size=1.5)+
  labs(x="Number of Clusters", y="Total of the Within Clusters Sum of the Squares")+
  scale_x_continuous(breaks=c(1:10))
```



The k-means clustering analysis

The next section of code actually carries out a k-means clustering for 3 to 7 clusters using 100 starting points for each clustering and setting a maximum number of iteractions of 1000. These clusters are then written to a csv file along with the sequence names. The final part of the code gives the summary statistics for the 3 cluster solution.

```
# K-Means Cluster Analysis for 3 to 7 clusters
fit7 <- kmeans(y, 7, iter.max=1000, nstart=100) # 7 cluster solution
fit6 <- kmeans(y, 6, iter.max=1000, nstart=100) # 6 cluster solution
fit5 <- kmeans(y, 5, iter.max=1000, nstart=100) # 5 cluster solution
fit4 <- kmeans(y, 4, iter.max=1000, nstart=100) # 4 cluster solution
fit3 <- kmeans(y, 3, iter.max=1000, nstart=100) # 3 cluster solution
# append cluster assignment to the sequence names and save as a cvs file
mydata <- data.frame(cbind(fit3$cluster,fit4$cluster,fit5$cluster,fit6$cluster,fit7$cluster))
write.csv(mydata, "clustered_HA_kmeans_RNA.csv")
# Show the summary statistics for the 3 cluster solution
fit3$size
```

```
## [1] 1043  718  884
```

```
fit3$iter
```

```
## [1] 2
```

```
fit3$betweenss
```

```
## [1] 70321.61
```

```
fit3$withinss
```

```
## [1]   69.46099 2198.77436  925.48590
```

The bootstrap diagnostics of the clustering

This code fragment creates a function kmboot for bootstrapping the k-means clustering. This is the 4 cluster solution. Each time the clustering is run the cluster numbers are assigned differently and so a hierarchical clustering has to be run on the results of the 1000 boostraps in order to identify the patterns of clustering and to put them in cluster order. That is the function of the wordorder subroutine. The final bootstrap consensus clusters are then plotted as a heatmap.

```
kmboot <- function(x, nclust) {
  res <- kmeans(x, nclust)
  return(res$cluster)
}
kmeansboot <- clustboot(y, nclust=4, kmboot, nboot=1000, diss = FALSE)

wardorder <- function(x, nclust) {
  res <- hclust(x, method = "ward.D2")
  member <- cutree(res, nclust)
  return(member)
}
consensuskmeans <- consensusmatrix(kmeansboot, nclust=4, wardorder)
clustheatmap(consensuskmeans, "HA Clustering")
```

The USEARCH results

The final code segment shows the code used to generate the plot of the cluster numbers from USEARCH. Firstly the csv file containing the data has to be read in and then the data is plotted using ggplot2.

```
usearch <- read.csv("usearch_clustering.csv", header=TRUE)

ggplot(data=usearch, aes(x=Identity, y=Clusters, group=Alignment))+
  geom_line(aes(colour=Alignment), size=1.5)+
  geom_point(aes(shape=Alignment), size=2)
```