**Finding the most Similar Species from species samples and their genetic markers**

By

JaivigneshRameshbabu

Aegis School of Data Science

Mumbai.

Problem Statement:

The dataset provided here (<https://drive.google.com/file/d/0B_lEnc_EG8wHWWJHS2xfbUo5OFE/view?usp=sharing>) consists of 30 species samples and their genetic markers (over 22,000). Use two or more ML techniques to find the most similar species.

Data Exploration

The first step is to look at the data and try to collect the information from the given data

#Clearing the Environment Memory  
remove(list =ls())  
#Importing the data Set  
SD=read.csv("C:/Users/Jaivignesh/Downloads/Species\_data.csv")  
#EDA  
summary(SD)

## entrez symbol S1   
## Min. : 11287 Gm8297 : 3 Min. : 4.403   
## 1st Qu.: 50790 LOC100861615 : 3 1st Qu.: 4.403   
## Median : 75828 1-Mar : 2 Median : 4.721   
## Mean : 6521874 1700106J16Rik: 2 Mean : 5.611   
## 3rd Qu.: 245276 2-Mar : 2 3rd Qu.: 6.510   
## Max. :105247288 4-Sep : 2 Max. :17.088  
## (Other) :22397   
## S2 S3 S4 S5   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.755 Median : 4.759 Median : 4.701 Median : 4.734   
## Mean : 5.619 Mean : 5.606 Mean : 5.644 Mean : 5.627   
## 3rd Qu.: 6.506 3rd Qu.: 6.517 3rd Qu.: 6.520 3rd Qu.: 6.547   
## Max. :16.464 Max. :16.356 Max. :17.250 Max. :15.996  
##   
## S6 S7 S8 S9   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.744 Median : 4.687 Median : 4.710 Median : 4.677   
## Mean : 5.650 Mean : 5.642 Mean : 5.647 Mean : 5.649   
## 3rd Qu.: 6.523 3rd Qu.: 6.563 3rd Qu.: 6.530 3rd Qu.: 6.510   
## Max. :15.881 Max. :15.983 Max. :16.278 Max. :16.200  
##   
## S10 S11 S12 S13   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.696 Median : 4.707 Median : 4.722 Median : 4.403   
## Mean : 5.645 Mean : 5.652 Mean : 5.660 Mean : 5.625   
## 3rd Qu.: 6.432 3rd Qu.: 6.448 3rd Qu.: 6.443 3rd Qu.: 6.392   
## Max. :16.455 Max. :16.506 Max. :17.112 Max. :16.834  
##   
## S14 S15 S16 S17   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.403 Median : 4.403 Median : 4.403 Median : 4.403   
## Mean : 5.617 Mean : 5.631 Mean : 5.623 Mean : 5.629   
## 3rd Qu.: 6.389 3rd Qu.: 6.364 3rd Qu.: 6.408 3rd Qu.: 6.402   
## Max. :17.195 Max. :17.010 Max. :17.231 Max. :16.902  
##   
## S18 S19 S20 S21   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.757 Median : 4.403 Median : 4.716 Median : 4.403   
## Mean : 5.643 Mean : 5.612 Mean : 5.632 Mean : 5.616   
## 3rd Qu.: 6.402 3rd Qu.: 6.419 3rd Qu.: 6.428 3rd Qu.: 6.421   
## Max. :16.505 Max. :17.218 Max. :16.838 Max. :17.272  
##   
## S22 S23 S24 S25   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.403 Median : 4.403 Median : 4.691 Median : 4.403   
## Mean : 5.600 Mean : 5.612 Mean : 5.628 Mean : 5.616   
## 3rd Qu.: 6.440 3rd Qu.: 6.397 3rd Qu.: 6.432 3rd Qu.: 6.455   
## Max. :17.100 Max. :16.766 Max. :16.574 Max. :17.075  
##   
## S26 S27 S28 S29   
## Min. : 4.403 Min.: 4.403 Min.: 4.403 Min. : 4.403   
## 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403 1st Qu.: 4.403   
## Median : 4.718 Median : 4.707 Median : 4.724 Median : 4.741   
## Mean : 5.624 Mean : 5.625 Mean : 5.639 Mean : 5.625   
## 3rd Qu.: 6.456 3rd Qu.: 6.444 3rd Qu.: 6.455 3rd Qu.: 6.459   
## Max. :16.803 Max. :16.849 Max. :17.447 Max. :17.492  
##   
## S30   
## Min. : 4.403   
## 1st Qu.: 4.403   
## Median : 4.720   
## Mean : 5.630   
## 3rd Qu.: 6.434   
## Max. :17.058  
##

str(SD)

## 'data.frame': 22411 obs. of 32 variables:  
## $ entrez: int 14679 54192 12544 107815 11818 67608 12390 23849 29871 12858 ...  
## $ symbol: Factor w/ 22329 levels "0610005C13Rik",..: 8579 14975 3791 17476 2168 12901 3405 10347 17475 4549 ...  
## $ S1 : num 7.99 4.4 5.34 4.72 4.4 ...  
## $ S2 : num 8.65 4.4 5.62 5.13 4.4 ...  
## $ S3 : num 7.85 4.4 5.89 5.02 4.4 ...  
## $ S4 : num 9.43 4.4 5.53 5.13 4.4 ...  
## $ S5 : num 8.48 4.4 5.95 5.14 4.4 ...  
## $ S6 : num 8.43 4.4 5.8 5.08 4.4 ...  
## $ S7 : num 8.79 4.4 5.75 4.8 4.4 ...  
## $ S8 : num 8.23 4.4 5.76 4.4 4.4 ...  
## $ S9 : num 8.61 4.4 6.19 5.01 4.4 ...  
## $ S10 : num 8.75 4.4 5.68 4.91 4.4 ...  
## $ S11 : num 8.81 4.4 5.62 4.93 4.4 ...  
## $ S12 : num 8.99 4.4 6.09 4.79 4.4 ...  
## $ S13 : num 9.33 4.4 5.13 4.4 4.4 ...  
## $ S14 : num 9.29 4.4 5.27 4.91 4.4 ...  
## $ S15 : num 8.86 4.4 5.3 4.78 4.4 ...  
## $ S16 : num 8.61 4.4 5.31 4.4 4.4 ...  
## $ S17 : num 8.54 4.4 4.76 4.4 4.4 ...  
## $ S18 : num 9.09 4.4 4.9 4.4 4.4 ...  
## $ S19 : num 9.52 4.4 4.84 4.4 4.4 ...  
## $ S20 : num 8.74 4.4 4.94 4.4 4.4 ...  
## $ S21 : num 8.99 4.4 5.07 4.4 4.4 ...  
## $ S22 : num 8.86 4.4 5.2 4.4 4.4 ...  
## $ S23 : num 8.7 4.4 5.21 4.4 4.4 ...  
## $ S24 : num 8.21 4.4 5.16 4.4 4.4 ...  
## $ S25 : num 8.93 4.4 4.77 4.4 4.4 ...  
## $ S26 : num 8.52 4.4 5.03 4.4 4.4 ...  
## $ S27 : num 8.26 4.4 5.39 4.4 4.4 ...  
## $ S28 : num 8.44 4.4 5.01 4.4 4.4 ...  
## $ S29 : num 8.68 4.4 5.37 4.4 4.4 ...  
## $ S30 : num 8.15 4.4 5.03 4.4 4.4 ...

dim(SD)

## [1] 22411 32

Data Cleaning

1. As per the Data Dimension, it has 22411Records with 32 columns But Species Data columns  $symbol: Factor variable: 22329 Unique Values . Hence Duplication is Possible.

Hence checking for the duplicates and removing it.

#Removing the Duplicates  
Rmdentrez=SD[!duplicated(SD$entrez),]  
Rmdsymbol=Rmdentrez[!duplicated(Rmdentrez$symbol),]

1. Missing Values are not found in the data Set.

#Missing values  
sum(is.na(SD)

Data Processing and Feature Engineering

1. There is no need of unique or reference ids like symbol and entrez columns for computation. Hence dropping it from dataset.

#Removing the entrez and symbols since they are unique ids  
SD\_Cont\_data=subset(Rmdsymbol, select = -c(entrez,symbol))

2.Normalizing the data set.

Even if variables are of the same units but it show quite different variances it is still a good idea to standardize before applying clustering Algorithms.

* Mean and StdDev Method

#normalizing the data using mean and SD  
m=apply(SD\_Cont\_data,2,mean)  
s=apply(SD\_Cont\_data,2,sd)  
NormalizeSD=scale(SD\_Cont\_data,m,s)

* *Using min max values in data set.*

scaling = function(y)

{

y = (y-min(y))/(max(y)-min(y))

}

I used mean and Standard deviation method in this problem since min and max values are almost similar in the data set.

3*.* Transposing the dataset

For applying the clustering techniques across the dataset we need to transpose the dataset.

#Transpose is required for clustering the data  
TransposeSD=t(NormalizeSD)

Model building:

Since I need to find the most Similar Species from species samples and their genetic markers(30 Species sample and 22329 genetic markers).

It clearly shows that it can be solved using clustering and segmentation algorithms in machine learning

 So I have used two popular clustering algorithms to find the similar species and reason being it revolves around centroids and Eu*clidean*

*Distances in the dataset.*

1. Hierarchical clustering
2. K-Means clustering.

**Hierarchical clustering**

* Partitions can be visualized using a tree structure (a dendrogram)
* Does not need the number of clusters as input
* Possible to view partitions at different levels of granularities using different K

#cacluating Euclidean distance  
distance=dist(TransposeSD)

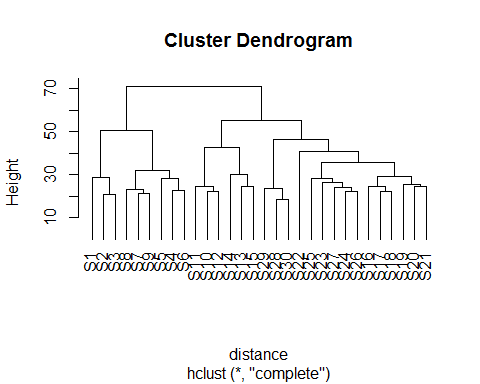
The algorithm works as follows:

* Put each data point in its own cluster.
* Identify the closest two clusters and combine them into one cluster.
* Repeat the above step till all the data points are in a single cluster.

Hclust function can be done in two methods,

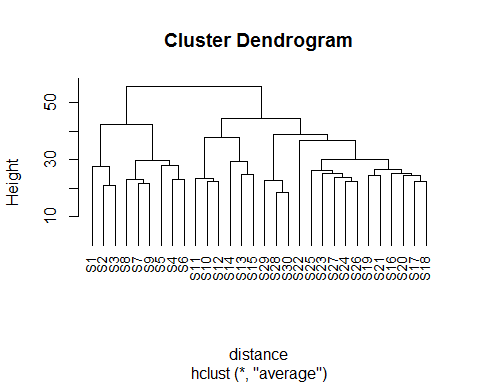
* Complete linkage clustering: Find the maximum possible distance between points belonging to two different clusters.

#Cluster Dendrogram with complete Linkage  
hc.c=hclust(distance)  
plot(hc.c,hang = -1)



* Mean linkage clustering: Find all possible pairwise distances for points belonging to two different clusters and then calculate the average.

#Cluster Dendrogram with Average Linkage  
hc.a=hclust(distance,method ="average")  
plot(hc.a,hang = -1,cex=0.9)



Creating the table to find the difference in cluster formation.

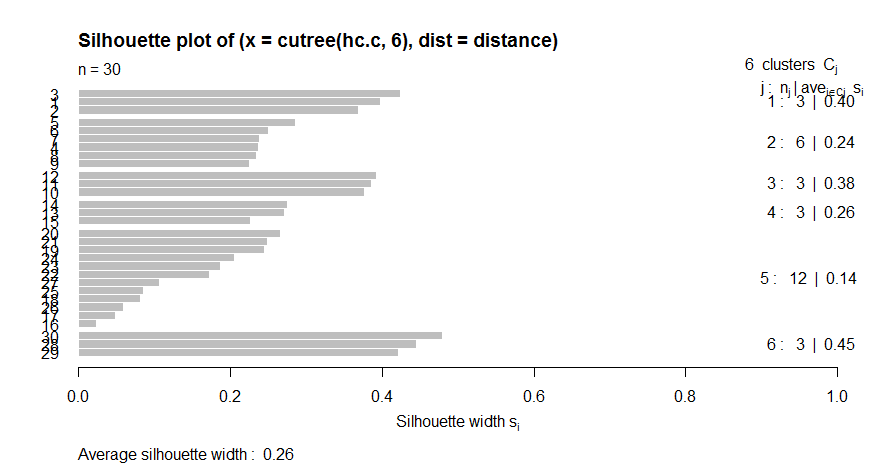
#cluster members  
member.c =cutree(hc.c,6)  
member.a =cutree(hc.a,6)  
table(member.a,member.c)

## member.c  
## member.a 1 2 3 4 5 6  
## 1 3 0 0 0 0 0  
## 2 0 6 0 0 0 0  
## 3 0 0 3 0 0 0  
## 4 0 0 0 3 0 0  
## 5 0 0 0 0 12 0  
## 6 0 0 0 0 0 3

Since there is no mismatch in above two method in clustering the species.This model works well for this Dataset.

**Silhouette plot for Hclust:**

#silhouette Plot  
library(cluster)  
plot(silhouette(cutree(hc.c,9),distance))



**K-Means Clustering:**

It is one of the popular algorithms used for segmenting the similar values.

The algorithm works as follows:

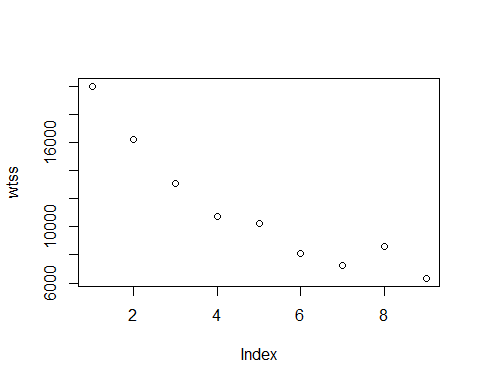
* K-means picks k number of points for each cluster known as centroids.
* Each data point forms a cluster with the closest centroids i.e. k clusters.
* Finds the centroid of each cluster based on existing cluster members. Here we have new centroids.
* As we have new centroids, repeat step 2 and 3. Find the closest distance for each data point from new centroids and get associated with new k-clusters. Repeat this process until convergence occurs i.e. centroids does not change.

**For finding the optimum k value,**

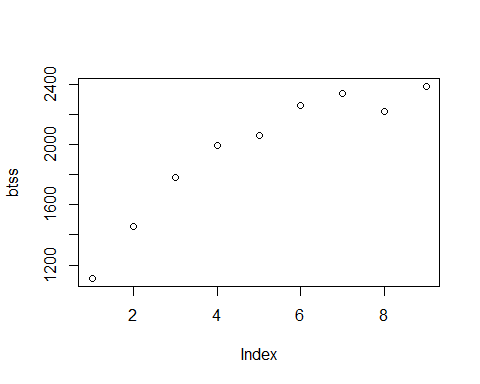
In K-means, we have clusters and each cluster has its own centroid. Sum of square of difference between centroid and the data points within a cluster constitutes within sum of square value for that cluster. Also, when the sum of square values for all the clusters are added, it becomes total within sum of square value for the cluster solution.

As the number of cluster increases, this value keeps on decreasing but if you plot the result you may see that the sum of squared distance decreases sharply up to some value of k, and then much more slowly after that. Here, we can find the optimum number of cluster

#Apply kmeans  
wtss =c()  
btss =c()  
for(i in 2:10)  
{  
kmeanscluster =kmeans(TransposeSD,i)  
wtss =c(wtss,kmeanscluster$tot.withinss)  
btss =c(btss,kmeanscluster$betweenss)  
}  
plot(wtss)



plot(btss)



As per the plot after 6 the data starts smoothen. Hence I choose( k=6 ).

**Applying Algorithm:**

#kmeans  
kc=kmeans(TransposeSD,6)  
summary(kc)

## Length Class Mode  
## cluster 30 -none- numeric  
## centers 133974 -none- numeric  
## totss 1 -none- numeric  
## withinss 6 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 6 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

kc$cluster

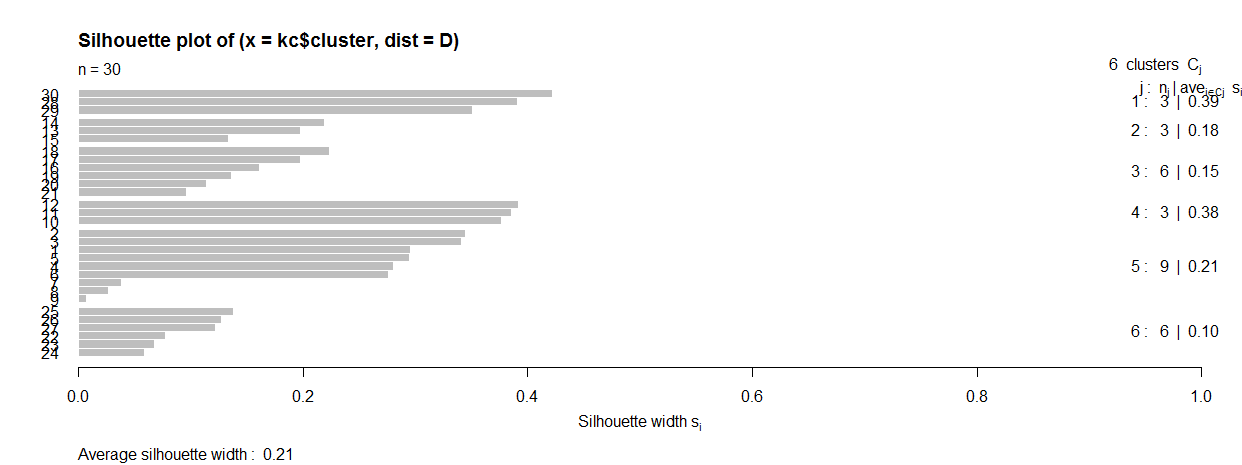
## S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18   
## 5 5 5 5 5 5 5 5 5 4 4 4 2 2 2 3 3 3   
## S19 S20 S21 S22 S23 S24 S25 S26 S27 S28 S29 S30   
## 3 3 3 6 6 6 6 6 6 1 1 1

kc$size

## [1] 3 3 6 3 9 6

**Silhouette plot for Kmeans**

D=daisy(TransposeSD)  
plot(silhouette(kc$cluster, D))



**Conclusion:**

So in this dataset both the model worked well but Hierarchical Clustering is slightly better than kmeans. This reason is below.

**Silhouette plot interpretation:**

* silhouette plot is used to understand the cluster visually.
* If the cluster formation is good and members in the clusters are closer to each other then (si) value will be high otherwise (si) will be low.

Since Average Si Value of Hclust (0.26) > Average Si value of Kmeans (0.21) .

When you look at individual value Si values of Hclust is much better than kmeans And also in the Silhouette plot for kmeans have Null values is higher its clearly indicates it’s not in that part of clusters.

Hence going with Hierarchical Clustering.

**Table for most Similar groups.**

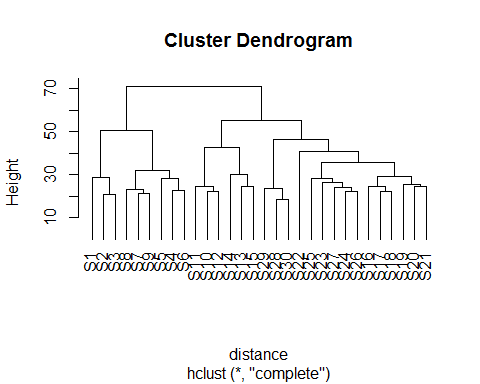
|  |  |  |
| --- | --- | --- |
| Species | K-Means | Hierarchical Clustering |
| Similar Species group 1 | S1 to S9 | S1 to S3 |
| Similar Species group 2 | S16 to S21 | S4 to S9 |
| Similar Species group 3 | S10 to S12 | S10 to S12 |
| Similar Species group 4 | S13 to S15 | S13 to S15 |
| Similar Species group 5 | S22 to S27 | S16 to S27 |
| Similar Species group 6 | S28 to S30 | S28 to S30 |

Note:

Species Similarities are matched exactly represented in Green,

And minor variablites are captured in yellow and blue .

**Hierarchical Tree Representaion.**



Note: Corresponding R files is also Attached in mail.