Unsupervised Learning in COVID DNA Sequences

Nitharsan Sivakanthan

6/1/22

Abstract:

Principal Component Analysis and K-means clustering can both be used to explore data and uncover patterns. Here we look at several thousand COVID DNA sequences collected primarily from Washington state. We uncover that there are certain locations and values on these sequences that account for most of the variation in all the data and that there is potential to differentiate these sequences into groups based on COVID variant or even origin.

Background:

Principal Component Analysis (PCA)-

PCA is a dimensionality reduction technique that can be used to approximate data by creating principal components that best describe the data.

PCA computes principal components Z_n using the following linear combination.

$$Z_n = \phi_{1n}x_1 + \phi_{2n}x_2 + \cdots + \phi_{p_n}x_P$$

Each ϕ_ρ is a linear combination of features p that correspond to the direction of greatest variance and the observations x.

In performing PCA, our principal components now give the location of observations in a reduced dimensional space with the principal components having a hierarchical order, the first principal component being the direction of greatest variance.

Furthermore, a small number of principal components that account for much of the variance in the data can be used to approximate the data.

Singular Value Decomposition (SVD)-

Another method for dimensionality reduction is SVD. When the data is scaled by mean and variance, this method is equivalent to PCA. This method breaks down our observations into three matrices: a matrix of the principal components, a matrix of the scaling of all principal components called the singular values, and a matrix mapping the data to the principal components.

Imputation of missing data can be performed using SVD to estimate the true values of the missing data. First, the average is used to impute the missing data and then SVD is iteratively used to obtain best estimates for the missing data.

K-means Clustering-

This method clusters mainly numerical data into k groups of data that are closest together in terms of Euclidian distance. Each of the k groups has a centroid and membership of each group is determined by a points distance to each centroid. The centroids serve as a computationally convenient way to minimize the distance between points within a group.

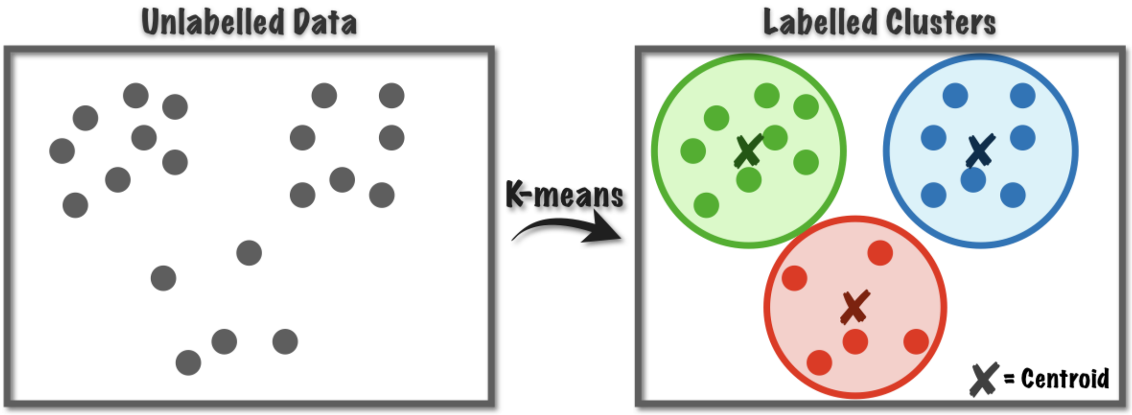


Fig. 1.: "K-Means: A Complete Introduction." *Medium*, Towards Data Science, 19 Nov. 2019, https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c.

Performing k-means clustering requires initialization of k random points for the initial centroids. Then, we determine which groups points are allocated to based on their distance to these centroids. New centroids are then computed based on the average of all the points in a group. This process of allocating points to groups and computing new centroids is repeated until points are no longer assigned to new centroids or equivalently when the centroids do not change.

Methodology:

Data –

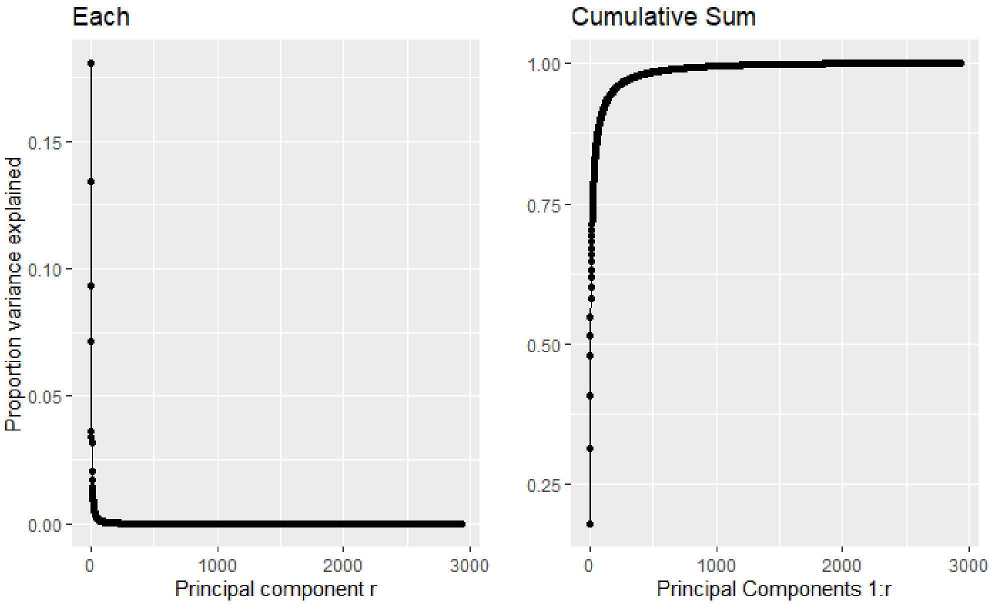
The data consists of genetic sequences of COVID mainly from Washington state collected from GISAID. The sequences originally indicated the nucleobases that make up DNA denoted by the characters A, C, G, and T. Our data has been processed to be numerical such that A = 1, C = 2, G = 3, and T = 4. There are many values in the genetic sequences that are missing.

Process –

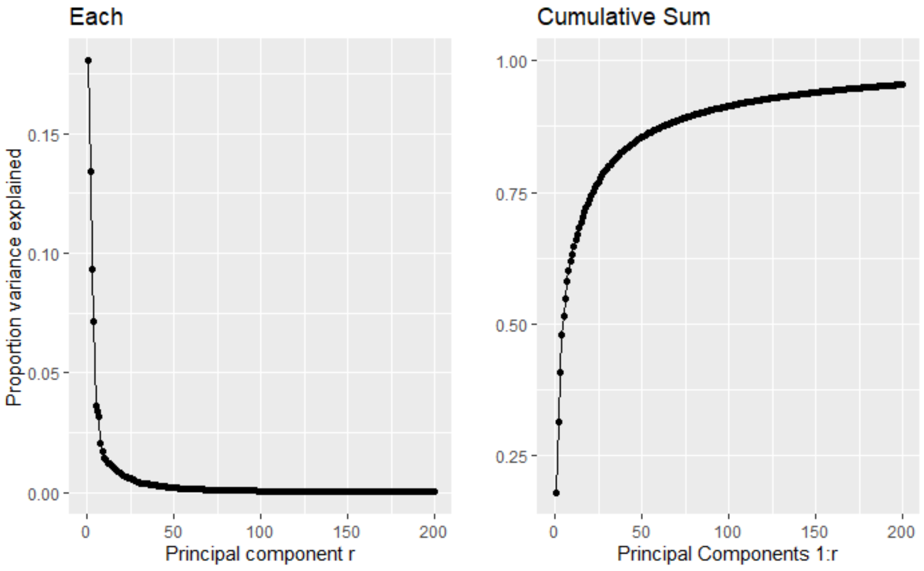
To perform PCA, the missing data must be dealt with. SVD imputation is used to estimate the true values of the missing data. Once the imputation is complete, PCA is run on the imputed data. To further explore the data, K-means is run on the imputed data to discover clusters of genetic sequences. Analysis is performed on these results to find patterns and potential relationships.

Results:

Once the data is prepped with the missing data imputed, PCA can be performed. Here are plots of the proportion of variance explained by the data per principal component and a cumulative sum plot, respectively.



We will zoom in to get a better idea of the proportion of variance explained for the first 200 principal components.



Below is a table of some numbers of principal components and their corresponding cumulative sum of proportion of variance explained by the data:

Number of Principal Components	Cumulative sum of proportion of variance explained
20	.7376
50	.8551
200	.9548
1000	.9952

These plots and table indicate a significant portion of the variance can be explained by 20-50 principal components, but for more accuracy a greater number of principal components can be considered.

Below is a table of the weights of the first 20 observations for the first three principal components:

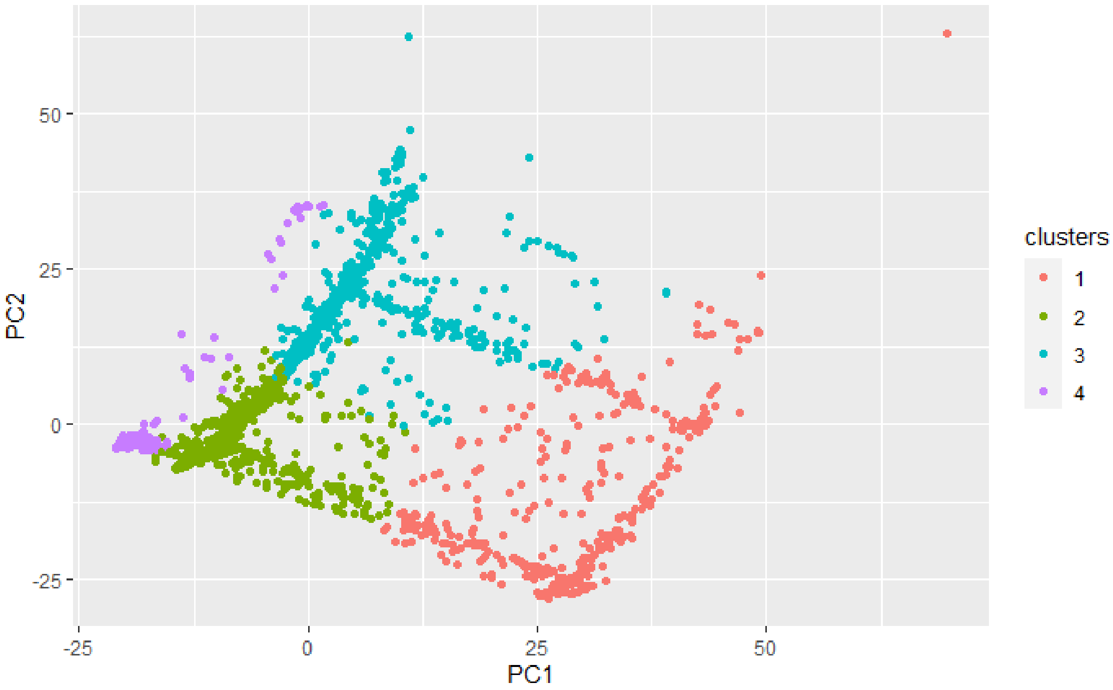
COVID Sequence	PC1	PC2	PC3
1	0.000803	0.000521	0.000958
2	0.004009	0.002344	0.004759
3	0.009508	0.006903	0.008789
4	0.002997	0.00205	0.002537
5	0.003384	0.001971	0.001321
6	0.003792	0.001961	0.000495
7	0.011744	0.006216	0.001455
8	0.013414	0.006181	0.001379
9	0.020293	0.008594	0.002741
10	0.020448	0.009589	0.003393

This table shows us that there are certain positions and values on these COVID genetic sequences that could potentially indicate which variant of COVID the individual had or even help trace the origins of their virus.

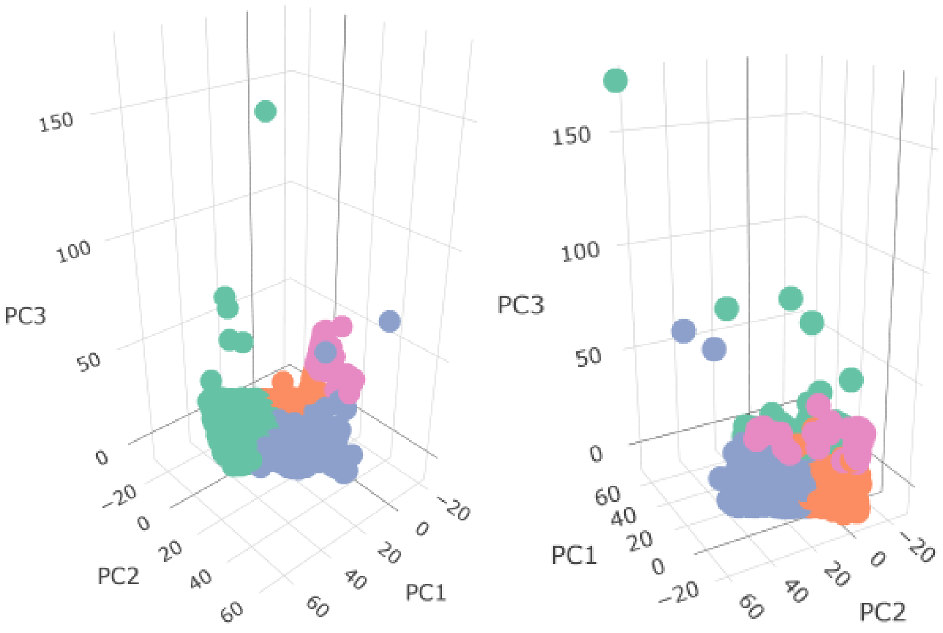
Next, k-means is used to cluster the data into groups. For the following analysis we will perform k-means on the first 50 principal components, as this captures a large portion of the variance explained.

Conceptually here, it would make sense to cluster the data into groups based on the number of COVID variants that are prevalent in Washington state. We will start with k = 4.

The following is a graph of the first two principal components and the four clusters resulting from k-means each labeled by color:



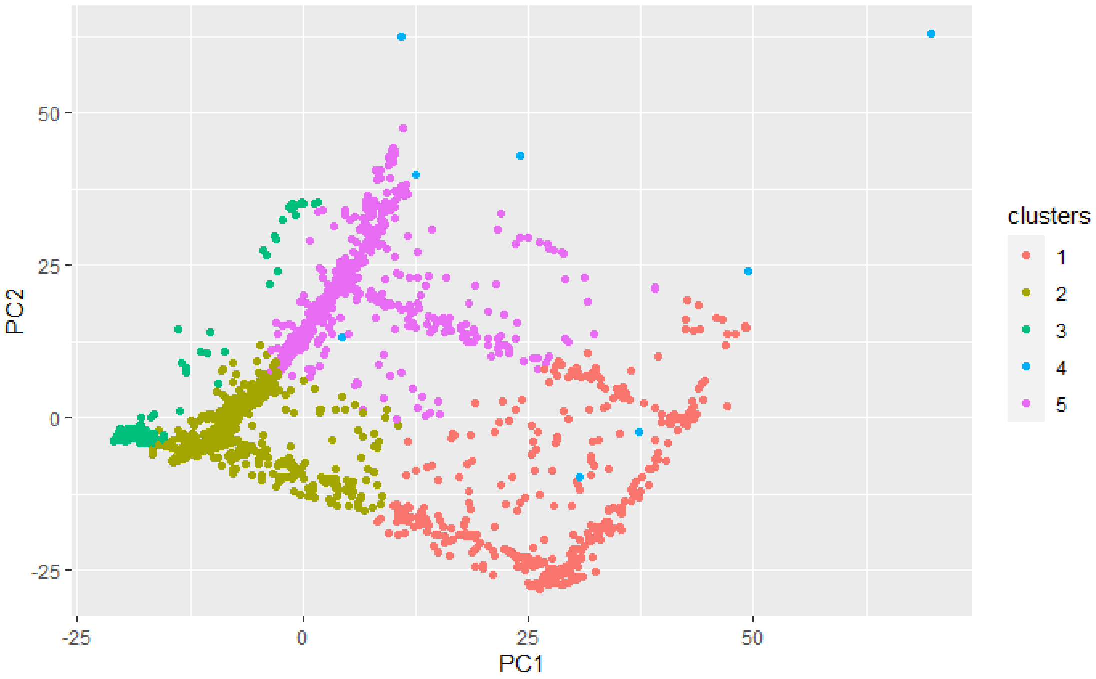
From only two principal components, it appears that the purple cluster does not seem to be much different from the other clusters. However, if we also plot PC3, we get a better picture of the differences.



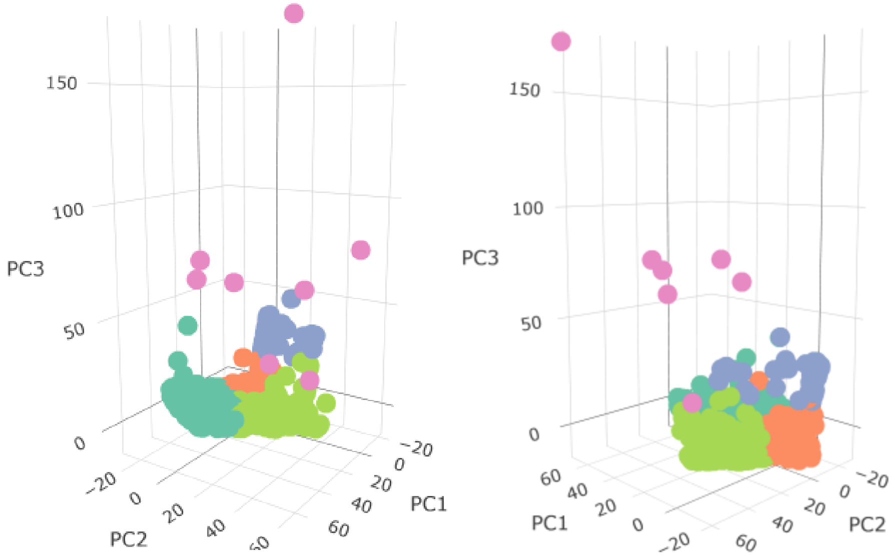
These are different orientations of the same plot of the first three principal components and the four clusters resulting from k-means.

From these plots the differences between each cluster are more distinguishable, however it does appear that maybe a different number of clusters could be used to differentiate some of the points farther away.

Next, we will try k-means clustering with k = 5.



Here, some of the observations that appeared as outliers before are now grouped together in their own cluster. Again, it will be easier to distinguish these points when viewing the first three principal components.



Again, these are different orientations of the same plot of the first three principal components, but now for five clusters resulting from k-means. Here it is easier to see the sequences that appear far away from the other clusters are grouped together in their own cluster.

Conclusion:

It appears that there are significant patterns between COVID genetic sequences from different individuals in the Washington state area. When performing PCA, we observe certain positions and values on the genetic sequences have a high influence on the variation of the sequences. Clustering on many of the principal components from PCA, we can further distinguish sequences potentially by their origin or variant of COVID.

Appendix:

```
# import libraries

library(tidyverse)

library(plotly)

library(softImpute)

library(plotly)
```

```
# read in data, drop date column, create matrix

data = read.csv('covid_vals.csv',header = TRUE)

x = data %>%

  select(!date) %>%

  as.matrix()
```

```
# change into sparce matrix, scale, perform soft impute, impute the entire matrix

xs = as(x,"Incomplete")

xsc = biScale(xs, col.scale = FALSE, row.scale = FALSE)

fit = softImpute(xsc, type ='svd', rank.max = 20, trace.it = FALSE)

dataimp = complete(xsc, fit)
```

```
# perform PCA, graph variance explained
pr.out <- prcomp(dataimp)
pve <- data.frame(var = pr.out$sdev^2/sum(pr.out$sdev^2))
pve$id <- as.integer(row.names(pve))
p1 <- ggplot(pve, aes(x=id, y=var)) +
   geom_point()+
   geom_line()+
   labs(x='Principal component r',
      y='Proportion variance explained',
      title='Each')
p2 <- ggplot(pve, aes(x=id, y=cumsum(var)))+
   geom_point()+geom_line()+
   labs(x='Principal Components 1:r',
      y='',
      title='Cumulative Sum')
grid.arrange(p1, p2, ncol=2)

# show weights from PCA
pr.out$rotation[1:20,1:3]

# calculate cumulative variance explained for range of principal components
sum(pve$var[1:20])

# perform k-means on principal components and graph clusters
pca <- data.frame(pr.out$x)
pca_km <- pca[,1:50]
km_out <- kmeans(pca_km, 4, nstart = 20)
ggplot(data = pca_km, mapping = aes(x = PC1, y = PC2, col = as.factor(km_out$cluster))) +
  geom_point()+
  labs(col = 'clusters')
fig = plot_ly()
fig = fig %>% add_markers(data = pca_km, x = ~PC1, y = ~PC2, z = ~PC3, color =
~as.factor(km_out$cluster), showlegend=FALSE)
fig = fig %>% layout(scene = list(xaxis = list(title = 'PC1'),
                   yaxis = list(title = 'PC2'),
                   zaxis = list(title = 'PC3')))
fig
```