Comparison of hierarchical clustering algorithms (single linkage, complete linkage and centroid) and k-means.

Rohith Pulluri

Abstract

In the sea of DNA and protein sequences, most sequences remain unknown to people. One of the fundamental needs in bioinformatics today is, given a number of query sequences, to group them based on their similarity to sequences in an external reference database or one another. The former one is known as sequence mapping by finding sequences' best match against a reference database, such as genome-resequencing. When no reference database is available, sequences can be clustered based on similarity to one another using sequence alignment.

Different clustering algorithms will produce different result on the same data. In this paper, I investigate difference between partitional clustering and hierarchical clustering and go further on proximity use in hierarchical clustering, i.e. single linkage, complete linkage and centroid linkage.

Introduction

Clustering is the process of partitioning a set of objects into subsets, such that objects in one cluster are like one another and different from objects in other clusters. Based on whether the cluster sets are nested or not, clustering algorithms can be categorized into 2 type: partitional clustering and hierarchical clustering. Partitional clustering simply divides dataset into non-overlapping subset, while hierarchical clustering organizes nested clusters using tree structure.

As one of partitional clustering algorithms, K-means (often referred as Lloyd's algorithm) aims to partition N objects into k groups of equal variances. With strengths of easy implementation and computational efficiency, K-means is popular for clustering analysis in data mining. On the other hand, as the name of K-means suggests, a priori k should be specified beforehand. This problem is troublesome since it is difficult to predict the number of clusters and there is no general theoretical solution to find the optimal k value for any given dataset. Also, K-means is sensitive to training data. Initial seeds, outliers, data scaling and order of input training data, all these have strong impact on the final cluster

results. For example, with different initial seeds (cluster centers) at random, K-means may generate different outcome each run time.

Hierarchical clustering has an alternative approach which does not require to prespecify a choice of k. It aims to output a hierarchy, a structure that is more informative than unstructured set of partitional clusters. Hierarchical clustering algorithm are either top-down (agglomerate) or bottom up (divisive). In agglomerative clustering method, each object is treated as a singleton cluster at outset and successively merge pairs of clusters until all clusters have been merged into a single cluster which contains all objects. The merging criteria is based on similarities between new cluster and each of old cluster. Depending on different similarity measures, agglomerative hierarchical clustering can be further categorized into single-linkage, complete-linkage and centroid linkage.

Data and Feature

The data I use for this project is 16S rRNA reference gene sequences (RDP) dataset, available from Source Forge or mothur project wiki page. This public training set is released in February 2016 and consists of a collection of 12681 bacterial and 531 archaeal 16S rRNA gene sequences with an improved taxonomy label.

```
Sequence
------------------------------------------------------------------------
Metadata:
    'description': 'Root;Bacteria;Firmicutes;Bacilli;Bacillales;Staphyl
                    ococcaceae;Staphylococcus'
    'id': 'D83355|S000413956'
Stats:
    length: 1476
------------------------------------------------------------------------
0    aggatgaacg ctggcggcgt gcctaataca tgcaagtcga gcgaacggac gagaagcttg
60   cttctctgat gttagcggcg gacgggtgag taacacgtgg ataacctacc tataagactg
...
1380 caccgcccgt cacaccacga gagtttgtaa cacccgaagc cggtggagta acctttagg
1440 agctagccgt cgaaggtggg acaaatgatt ggggtg
```

For the sake of runtime, I work with a small random subset these sequences, around 100 sequences each time.

Methods

1) Pairwise Sequence Local Alignment

   Sequence alignment is the procedure of comparing two or more multiple sequences by searching for a series of individual characters or patterns which are in the same order in the sequences. In this project, in order to speed up alignment process, I use optimized Striped Smith Waterman algorithm to compute the pairwise optimal local alignment. Finally, I get an $N \times N$ alignment score matrix which will be used as data features to train clustering

models, and an $N \times N$ distance matrix which is used for visualization demonstration. The distance matrix is converted into a 2-dimensional array using multidimensional scaling technique to make all DNA sequences into a point on 2-d plane.

2) K-means

I use K-means clustering algorithm which divides objects into prespecified k clusters based on scoring matrix which is computed from previous alignment process. For each sequence, it first iterates the distance matrix of its row to find k highest-scored "neighbors" and assign itself into the cluster containing the majority of his k "neighbors".

The k value I used in the experiment is k = 3, 5, 7, 9, 11, 13, 15, 17, 19.

3) Hierarchical clustering

In this project, using the same scoring matrix computed in alignment process, I compare 3 type of agglomerative hierarchical clustering with different similarity measure: single, complete and centroid.

In the single-link clustering, the similarity of two clusters is decided by the similarity of their most similar member. In the complete-link clustering, such similarity is decided by the similarity of their most dissimilar members. In the centroid clustering, the similarity of two clusters is decided by the similarity of their centroids.

Experiment and Analysis

1) Data preprocessing

There are 140 labeled DNA sequences in 1 category:

"Acidobacteria" : 2, "Actinobacteria" : 29, "Bacteroidetes" : 21, "Chloroflexi" : 1, "BRC1" : 1, "Armatimonadetes" : 3, "Euryarchaeota" : 8, "Proteobacteria" : 50, "Tenericutes" : 3, "Firmicutes": 16 "Spirochaetes" : 1, "Synergistetes" : 2, "Marinimicrobia" : 1, "Cyanobacteria/Chloroplast" : 2

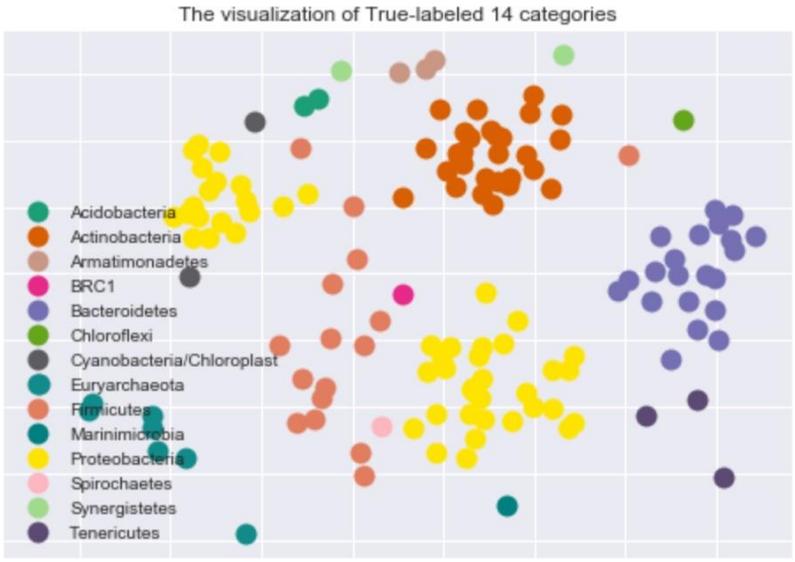


Fig 1 True labeled DNA sequence distribution

From the above figure, we can see that generally sequences in the same category are grouped closely with the same color.

2) K-means with $k = \{3, 5, 7, 9, 11, 13, 15, 17, 19\}$

I use Silhouette Coefficient to evaluation. The Silhouette Coefficient is calculated using intra distance a and the mean nearest cluster distance b for each sample.

$$\frac{b - a}{\max(a, b)}$$

Here is the Silhouette Coefficient result with above k value:

```
For n_clusters=3, The Silhouette Coefficient is 0.4007769691786452
For n_clusters=5, The Silhouette Coefficient is 0.41120944025499645
For n_clusters=7, The Silhouette Coefficient is 0.455802510941555
For n_clusters=9, The Silhouette Coefficient is 0.4083965582424128
For n_clusters=11, The Silhouette Coefficient is 0.35557203864308057
For n_clusters=13, The Silhouette Coefficient is 0.33912314896850826
For n_clusters=15, The Silhouette Coefficient is 0.3621948752381777
For n_clusters=17, The Silhouette Coefficient is 0.28064625654883446
For n_clusters=19, The Silhouette Coefficient is 0.29686224447482434
```

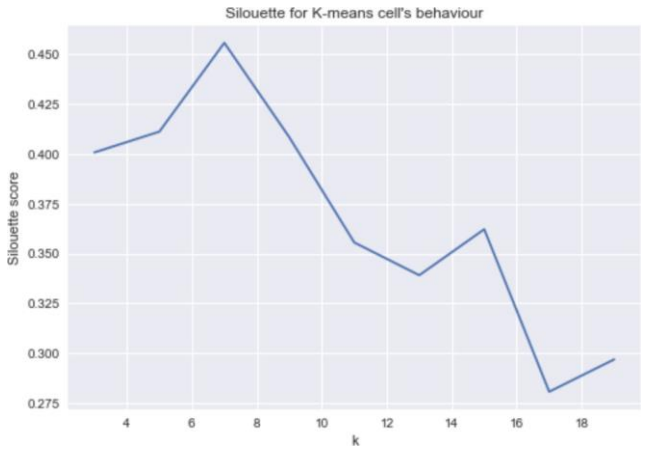


Fig 2 Silhouette scores for different k values

The above Silhouette Coefficient scores show that the optimal Silhouette Coefficient (0.4558) can be reached when k=7.

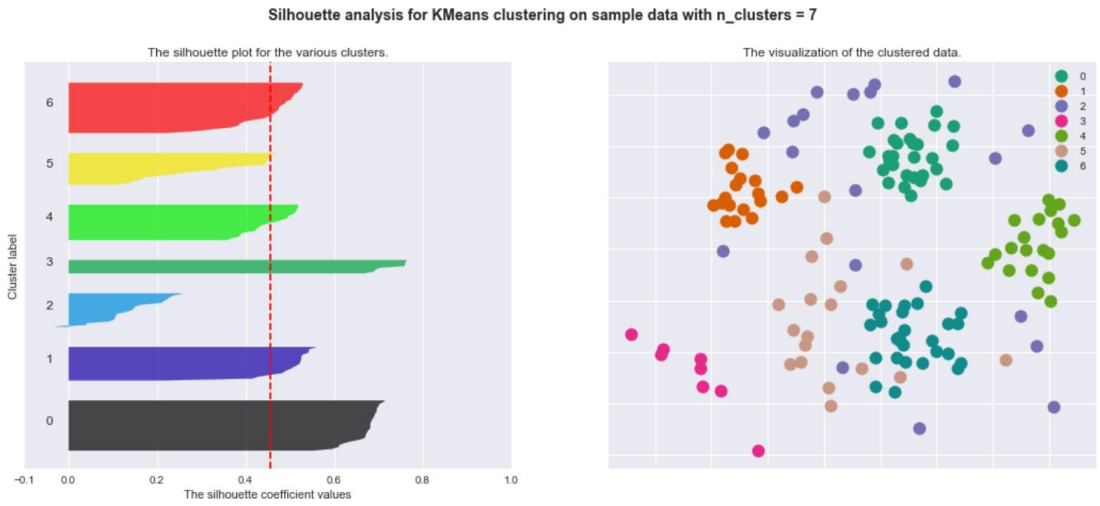


Fig 3 Silhouette analysis for k-means cluster (k=7)

Comparing the k-means clustering result with true-labeled visualization above, we can say that k-means can cluster the dataset almost same to their true labels. Meanwhile, an interesting point is that sequences which belong to 'Proteobacteria' class is divided into two different clusters: Cluster 0 and Cluster 6. In fact, in the true-labeled visualization, the Proteobacteria' class (colored yellow) is distributed into two clusters. A possible explanation for this could be that Proteobacteria has evolved into two sub-species.

3) Hierarchical clustering

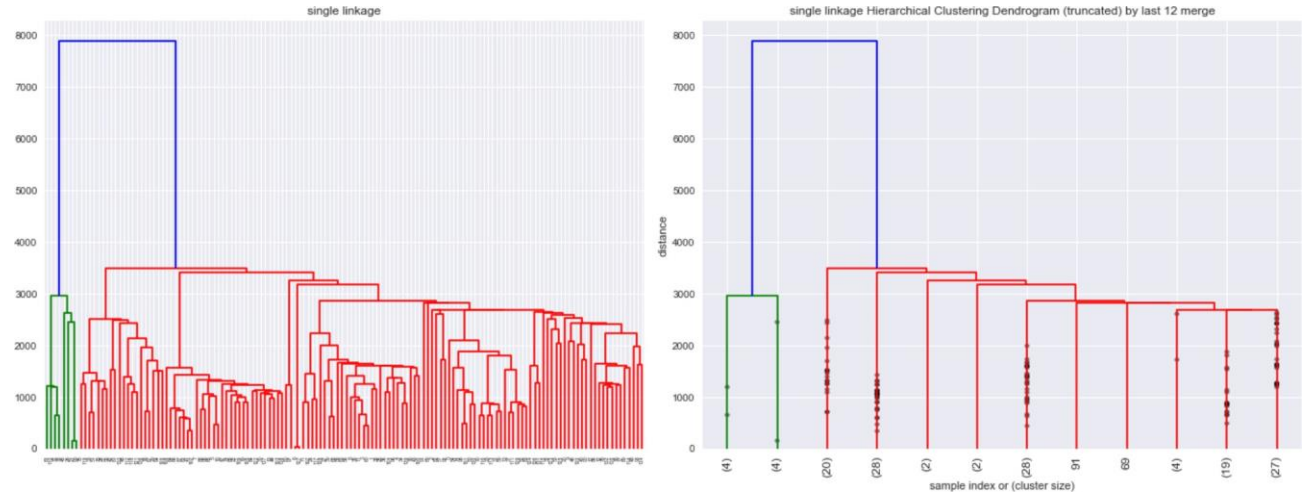


Fig 4 Single-link hierarchical clustering

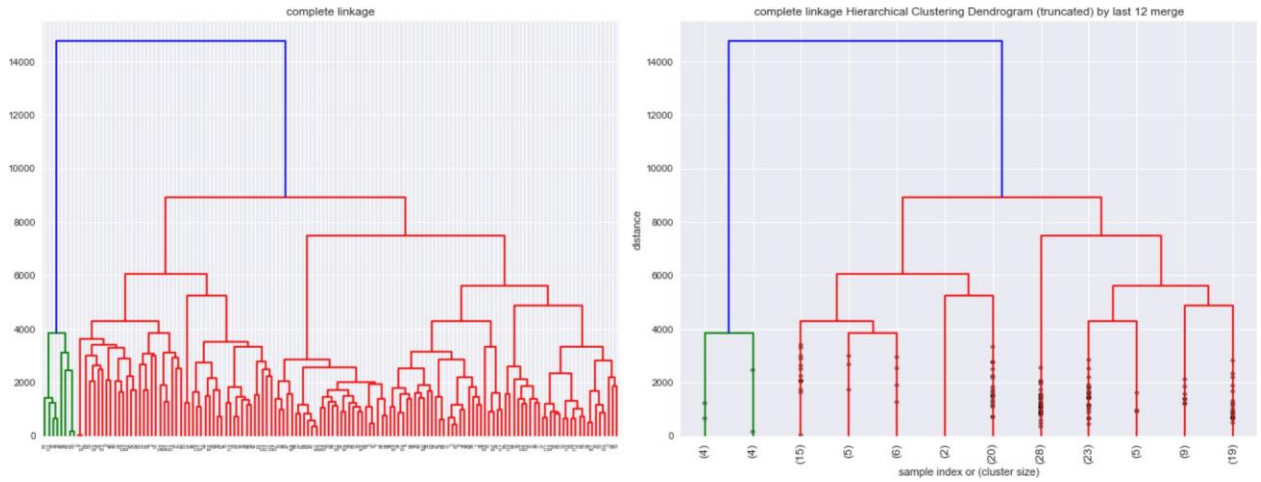


Fig 5 Complete-link hierarchical clustering

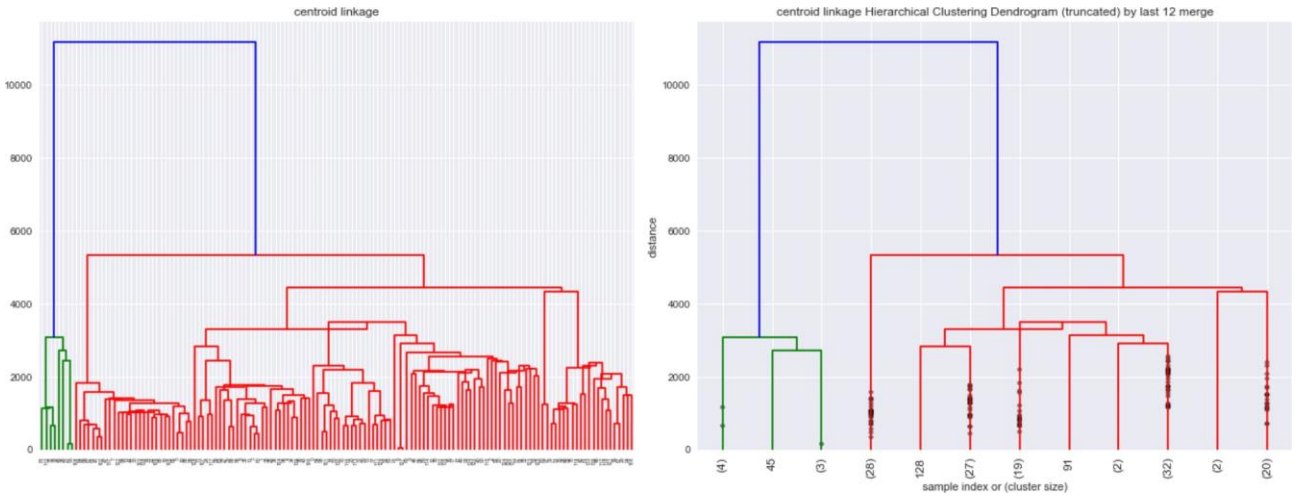


Fig 6 Centroid-link hierarchical clustering

The left panel shows the full dendrogram of clustering results while the right panel show a truncated dendrogram with only last 12 merges.

To make better understanding of clustering result, try example with a specific number of clusters by cutting dendrogram with 7.

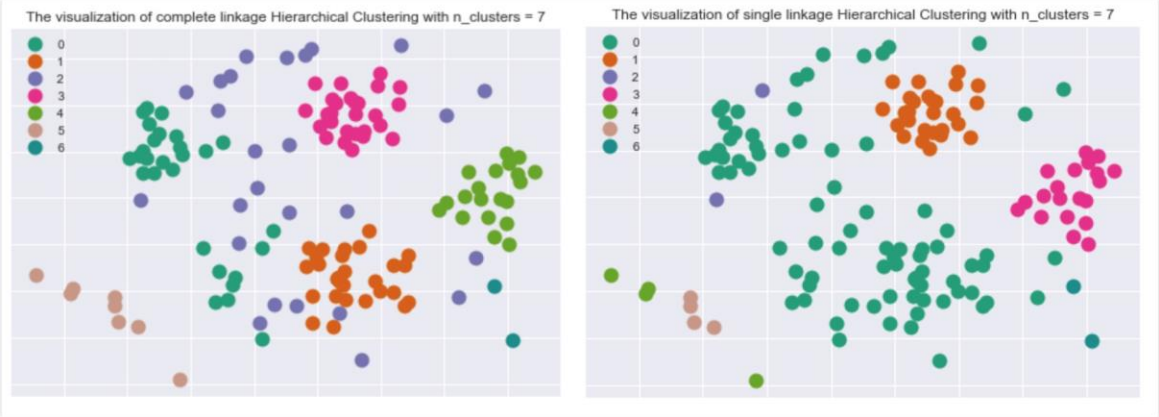


Fig 7 Complete and Single linkage clustering with 7

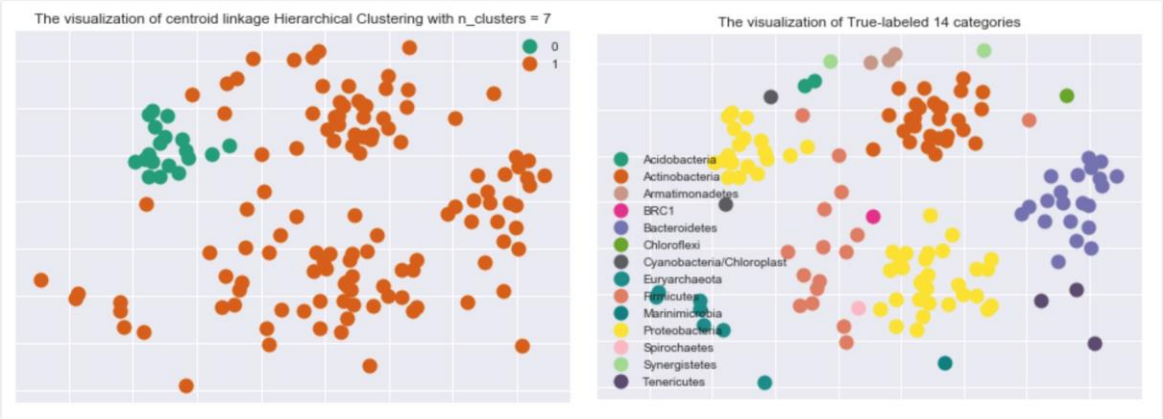
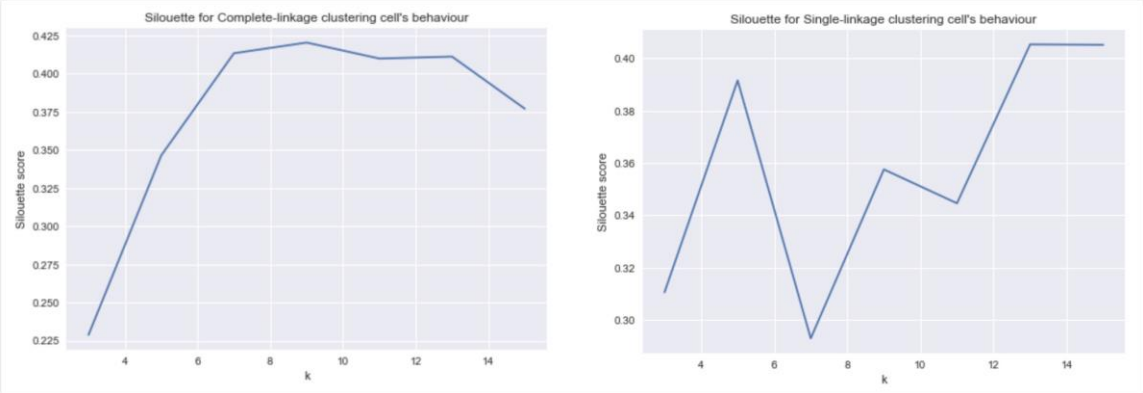


Fig 8 Centroid linkage clustering with 7 and True-labeled visualization

```
For single linkage Hierarchical Clustering, the Silhouette Coefficient with 7 is 0.29302476911079
For complete linkage Hierarchical Clustering, the Silhouette Coefficient with 7 is 0.41345886046628455
For centroid linkage Hierarchical Clustering, the Silhouette Coefficient with 7 is 0.04876989286122942
```

Notice that when we set to remain 7 clusters, we can safely get 7 clusters from both complete and single linkage clustering, but only get 2 clusters from centroid linkage clustering. This conforms with Silhouette coefficient score as centroid linkage clustering gets very low Silhouette coefficient score with 0.0488. When I set to remain 11 clusters, centroid clustering return 5 clusters, which is obviously less than required number. (please refer to ipython notebook)

Meanwhile, comparing complete linkage and single linkage with true-labeled visualization, we can find that complete link clustering, I find it is different to say which one is better. Although the cluster result of complete linkage is much similar to the true-labeled visualization, single linkage clustering groups Proteobacteria' class correctly.



Last, I compare Silhouette coefficient of complete and single clustering algorithm with different specified remaining clusters. Generally, with the increase of clusters remain, the Silhouette coefficient also increases. Particularly for complete linkage, when k = 7, the Silhouette coefficient gradually increase in a decreasing speed. However, k = 7 will make single linkage clustering algorithm score at the lowest level.

Conclusion

In this project, I compared partitional clustering and hierarchical clustering using 16S rRNA sequence dataset. For partial clustering, I investigate the choice of prespecified k value and evaluate the clustering results using Silhouette coefficient score. Compared with true-labeled visualization, the optimal clustering result can be reached when $k = 7$.

Besides I compare the hierarchical clustering result using different similarity measure of single linkage, complete linkage and centroid linkage. The experiment shows that even after building hierarchical dendrogram, it is still important to decide on the number of clusters k. For the same k clusters to remain, complete and single linkage shows different performance on cluster result.