Cluster Validation using the clValid R Package

Jill R

2019-12-04

## Introduction

Cluster analysis is a powerful, unsupervised data-mining technique that is used to elucidate patterns and/ or the underlying structure of high dimensional datasets, which may not be intuitively obvious. Fundamentally, the key objective of clustering is to split the data into groups so that highly similar observations are tightly clustered from those that are highly distinct in order to minimize the average within cluster distance while maximizing the average distance between clusters [1].

Not surprisingly, cluster analysis is not represented by a single algorithm, but rather, a multitude of algorithms exist[1-3]. Although designed to achieve the same goal (i.e., clustering of the data), these algorithms differ in both the way clusters are identified and defined, resulting in vastly different groupings of the data [1-3]. Unfortunately, there is no single clustering algorithm that consistently outperforms the rest, and therefore, deciding on the most appropriate method for a given dataset can be a fundamentally challenging and overwhelming task [1,3].

When selecting a clustering algorithm, it is common practice for researchers to rely on their prior knowledge and expertise, however, this process can be highly subjective. For example, a researcher may be biased towards selecting an algorithm that validates their own assumptions while disregarding alternative algorithms that may reveal contradicting results [1]. Such an approach is nonetheless counterproductive as it undermines the key purpose of conducting a cluster analysis, which is to discover novel patterns in the data for hypothesis generation and further study [1].

To obtain the most meaningful results, a researcher should always select an algorithm that generates compact, well-separated and biologically relevant clusters [3]. Likewise, it is also important for the researcher to keep in mind that clustering algorithms will attempt to find clusters even when they may not be present in the data. If this caveat is not taken into consideration, non-significant clustering could lead to misinterpretations of the data, potentially impacting downstream analyses and decision-making [1-2]. How can one be certain that the most appropriate clustering algorithm and number of clusters has been selected for a given dataset?

To ensure the results of the cluster analysis are of the highest quality, validation has become an increasingly important step in the cluster analysis workflow [1,3]. Essentially, the role of cluster validation is to evaluate both the appropriateness of the clustering algorithm and the optimal number of clusters for a given dataset [1,3]. When conducting cluster validation, it is recommended to use a variety of validation indices as these metrics can report different optimal clustering solutions for the same dataset, and therefore, using a single index could lead to incorrect assumptions about the validation analysis [4].

As a means for providing researchers with the necessary tools to efficiently validate the results of a cluster analysis, a number of R packages have been developed [5], including fpc [6], factoextra [7], NbClust [8], clv [9], and clValid [3], to name a few. However, unlike most currently available packages, *clValid* offers nine cluster validation measures grouped into three categories (i.e., internal, stability and biological) that can be used to evaluate ten different clustering algorithms (i.e., “hierarchical”, “agnes”, “diana”, “*k*-means”, “pam”, “clara”, “fanny”, “model”, “som”, and “sota”) [3]. As reported by the authors, the most notable strength of the *clValid* package has the ability to perform simultaneous validations of multiple clustering algorithms for a range of clusters and validation measures using a single line of code, which cannot currently be achieved by other programs [3]. As well, the interpretation of the results returned by *clValid* appear to be intuitive and straightforward as the “optimal” clustering algorithm and number of clusters can be returned using the *optimalScores* function based on the overall results of the validation metrics that were assessed [3].

**OBJECTIVE**

The objective of the current study is to determine the optimal clustering method for two high-dimensional datasets (for which the ground truth is known), using the *clValid* R package. Since the datasets contain a maximum of two clusters, the selected clustering algorithms will be assessed using a range of clusters from two to seven. Although a number of clustering algorithms exist within the package, only Hierarchical, *k*-means and PAM (i.e., *k*-medoids) will be investigated as these methods were explicitly discussed in class, and therefore, it is assumed that the intended audience will have a baseline knowledge of their inner workings. As a refresher, a brief overview of each methods as previously described in Holmes and Huber [2] is included below.

* *Partition-based*
  + Subdivide observations into k groups based on similarity to cluster centroids
  + The number of clusters (k) must be pre-specified by the researcher
    - Observations are placed in the cluster containing the cluster center it is closest to
    - Cluster centers are updated and the process is repeated until the clusters stabilize
  + Example algorithms include *k*-means and *k*-medoids (i.e., Partitioning Around Medoids; PAM)
    - Centroid is not represented by a single point in *k*-means, but rather, a collection of points (i.e., average distance of all points in a single cluster)
    - Centroid, or more appropriately termed, Medoid is the most central point in the cluster for *k*-medoids (i.e. PAM)
* *Hierarchical*
  + Tree-based (i.e., dendrogram) visualization of the observations
  + Does not require prior knowledge of the number of clusters (k)
  + Bottom-up approach
    - Agglomerative
      * Most common hierarchical clustering method
      * Used to find small clusters
      * Single observation clusters are iteratively merged based on similarity until all observations are contained in a single, large cluster.
  + Top-down approach
    - Divisive
      * All observations begin in a single large cluster and are iteratively separated based on dissimilarity until all observations are in their own cluster.
      * Not commonly used

## Important Note

The default distance metric for the available clustering algorithms and validation measures in the *clValid* package is set to “euclidean”. Additionally, the agglomerative method for hierarchical clustering is defaulted to “average linkage”. However, these parameters can be modified using the “metric” and method" switch, respectively, in the *clValid* function [3]. For the purposes of this study, all default parameters will be used unless otherwise noted.

## Datasets

The datasets that have been selected for analysis include two genomic datasets from the PulseNet Canada, which is the national molecular subtyping network for foodborne disease surveillance in Canada [9]. More specifically, the datasets contain allele data derived from whole-genome multilous sequence typing (wgMLST), which is a gene-by-gene approach that detects variation within a set of microbial genomes using a predefined set of pan-genomic loci (i.e., genes) [10].

**Outbreak 1**

The Outbreak 1 dataset contains whole-genome multilocus sequencing type (wgMLST) allele data for 74 *Listeria monocytogenes* isolates associated with two outbreaks, one of which was associated with ready to eat (RTE) meat products (n = 62), and the other, was linked to the consumption of contaminated leafy greens (n = 12).

**Sporadic**

Based on the fact that clustering algorithms are designed to find clusters, even if they do not actually exist within the data, a sporadic dataset (i.e., contains ONLY unrelated isolates) was selected to investigate the cluster validation results returned by the *clValid* package. This dataset contains wgMLST allele data for 26 *Listeria monocytogenes* isolates that were referred to the PNC network within a similar time frame but where considered to be epidemiologically unrelated (i.e., similar exposures were not identified).

## Data Preparation

To successfully run the *clValid* R package, the data must be formatted as a numeric matrix, data frame or an *ExpressionSet* object, and the columns MUST ONLY contain numeric data (i.e., NA’s must be replaced and categorical variables must be removed). Most importantly, the *clValid* function operates under the assumption that the rows in the dataset contain the “observations” that are intended to be clustered [3], and thus, special care must be exercised when formatting the data for analyses.

**Load the Datasets**

*Outbreak 1*

#Load data and set row.name to Isolate and replace ? with NA's  
LmOB2\_data = read.csv(here("LmOutbreakData.csv"), row.names = "Isolate", na.strings = "?")  
  
#Remove categorial variables  
LmOB2\_subdata = subset(LmOB2\_data, select = -c(Group))  
  
#Check data for NA's  
table(is.na(LmOB2\_subdata)) #20896 NA's in the dataset; these actually represent NO allele call for a particular locus

##   
## FALSE TRUE   
## 222120 20896

#Replace NA's with 99 (i.e., use an arbitrary # that is not already represented in the dataset)  
LmOB2\_clean = replace(LmOB2\_subdata, is.na(LmOB2\_subdata), 99L) # replacing with zero proved to cause internal issues with the clustering algorithms so 99 is being used; less problematic  
#L is used to change the objects from dbl integer to interger, which also proved to be an issue for downstream analyses  
  
# Identify which columns (i.e., loci) have zero variance   
which(apply(LmOB2\_clean, 2, var)==0)

## LMO00125 LMO00277 LMO00417 LMO00474 LMO00708 LMO01547 LMO01576 LMO01681   
## 27 94 222 279 489 1299 1328 1433   
## LMO02028 LMO02758 LMO02819 LMO02820 LMO02825 LMO02827 LMO02834 LMO02835   
## 1777 2457 2517 2518 2523 2525 2532 2533   
## LMO02839 LMO03064 LMO03161 LMO03384 LMO03891 LMO03908 LMO04070 LMO04503   
## 2537 2760 2840 2899 3027 3037 3199 3243   
## LMO04531   
## 3256

#Example of Locus LMO00125, which has zero variance  
summary(LmOB2\_clean$LMO00125)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1 1 1 1 1 1

#Remove columns with 0 variance as it is problematic for downstreamn analyses  
LmOB2 = LmOB2\_clean[ ,apply(LmOB2\_clean, 2, var) !=0]  
  
#Check to see that columns have been removed  
dim(LmOB2\_clean) #contains columns with zero variance

## [1] 74 3284

dim(LmOB2) # 25 columns with zero variance have been removed

## [1] 74 3259

**Sporadic**

#Load data and set row.name to Isolate and replace ? with NA's  
LmSP\_data = read.csv("./LmSporadic.csv", row.names = "Isolate", na.strings = "?")  
  
#Remove categorial variables  
LmSP\_subdata = subset(LmSP\_data, select = -c(Group))  
  
#Check data for NA's  
table(is.na(LmSP\_subdata)) #29509 NA's in the dataset; these actually represent NO allele call for a particular locus

##   
## FALSE TRUE   
## 73841 29509

#Replace NA's with 99  
LmSP\_clean = replace(LmSP\_subdata, is.na(LmSP\_subdata), 99L) # replacing with zero proved to cause internal issues with the clustering algorithms so 99 is being used; less problematic  
#L is used to change the objects from dbl integer to interger, which also proved to be an issue for downstream analyses  
  
# Identify which columns (i.e., loci) have zero variance   
which(apply(LmSP\_clean, 2, var)==0)

## LMO00132 LMO00474 LMO01547 LMO01576 LMO01681 LMO02028 LMO02820 LMO02825   
## 63 368 1408 1437 1542 1889 2668 2673   
## LMO02835 LMO02839   
## 2683 2687

#Example of Locus LMO00048, which has zero variance  
summary(LmSP\_clean$LMO00132)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1 1 1 1 1 1

#Remove columns with 0 variance as it is problematic for eclust in the factoextra R package  
LmSP = LmSP\_clean[ ,apply(LmSP\_clean, 2, var) !=0]  
  
#Check to see that columns have been removed  
dim(LmSP\_clean) #contains columns with zero variance

## [1] 26 3975

dim(LmSP) # 10 columns with zero variance have been removed

## [1] 26 3965

The authors of *clValid* recommend to run all validation measures simultaneously for each of the cluster algorithms over a range of cluster numbers as it is reported to be more computationally efficient [3], however for the current study, the validation measures will be run separately. Following a preliminary analysis of the data, it was discovered that some metrics are quite computationally demanding (i.e., stability), and therefore, require additional resources beyond the capabilities of RStudio.

## Internal Validity

The purpose of internal validation measures is to assess the appropriateness of the clustering algorithm based on the intrinsic properties of the dataset, and therefore, these metrics do not take into account external sources of information [1, 3, 10]. More specifically, these measures are used to evaluate whether the selected clustering algorithm captures the natural structure of the data.

Three internal validation metrics have been implemented in the *clValid* package, and include “connectedness”, “compactness”, and “separation” [3]. A brief explanation of each metric is included below.

*Connectivity*

Connectivity, or more commonly, connectedness, is used to measure the degree to which observations are placed in the same cluster as their “*k*-nearest” neighbours (for a given algorithm) [3, 10].

For a particular clustering algorithm, , the connectivity can be defined as follows:

* Where,
  + *L* = number of neighbours that contribute to the connectivity
  + *N* = number of observations placed into *k* clusters
  + = is the *j*th nearest neighbour to observation *i*
  + = zero if *i* and are in the same cluster, and 1/*j* if they are not in the same cluster

The connectivity validation measure returns a value between zero and infinity, where lower values (i.e., closer to zero) reflect optimal connectivity [3].

*Compactness and Separation*

Compactness is used to assess how close observations are within the same cluster (i.e., intra-cluster variance), while separation quantifies the distance between clusters. The neighbSize, which indicates the number of observations to consider “neighbours” when calculating the connectivity is set to a default of 10 [3]. This default was used unless otherwise stated.

Given the opposing nature of these metrics (i.e., an increase in the compactness of clusters will result in a subsequent decrease in the separation between clusters), these measures are typically combined into a single score. Two commonly used methods that assess both the compactness and separation of clusters are the Silhouette Width [3, 4, 11, 12] and the Dunn Index [3, 4, 11, 13], which are described in greater detail below.

*Silhouette Width*

The Silhouette Width is used to evaluate how well observations are clustered based on the average distance between clusters (i.e., how close is an observation in one cluster to an observation in a neighbouring cluster?) [3, 4, 12].

The Silhouette Width is defined as follows and is computed according to the following steps:

1. Calculate the average dissimilarity, , for each observation, *i*, and all other observations in the same cluster.
2. Calculate the average dissimilarity, of to all observations in other clusters . The smallest value of is defined as , where is the average distance between and the observations in the nearest cluster to which it does not belong.

The values are reported in the range of -1 to 1 where observations are considered to be well clustered if the value is large (i.e., close to 1). In contrast, an value close to zero indicates the observation likely lies somewhere between two clusters, while a negative value suggests the observation is most likely in the wrong cluster [11].

*Dunn Index*

Provides a ratio of the smallest distance between two observations that are not in the same cluster to the largest distance between observations in the same cluster. The values reported for the Dunn Index are between zero and infinity with a large Dunn Index indicating the clusters are compact and well-separated [3, 11, 13].

The Dunn Index is defined as follows and is computed according to the following steps:

1. Compute the distance between each observation in a cluster and the observations in other clusters to determine the minimum pairwise inter-cluster distance, which represents the .
2. Compute the distance between observations in the same cluster to determine the maximal intra-cluster distance, which represents the .

**Internal Validity Assessment**

*Outbreak 1*

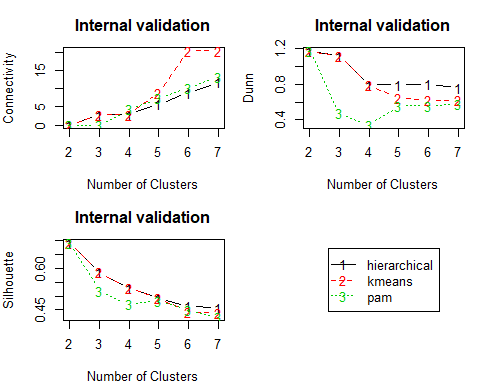
#Internal Validity Check for 2 to 7 clusters for 3 clustering methods  
intcheck\_LmOB2 = clValid(LmOB2, nClust = 2:7, clMethods = c("hierarchical", "kmeans", "pam"), validation = "internal")  
  
#View summary results and optimal scores  
summary(intcheck\_LmOB2)

##   
## Clustering Methods:  
## hierarchical kmeans pam   
##   
## Cluster sizes:  
## 2 3 4 5 6 7   
##   
## Validation Measures:  
## 2 3 4 5 6 7  
##   
## hierarchical Connectivity 0.0000 2.9290 2.9290 5.8579 8.7869 11.7159  
## Dunn 1.1787 1.1159 0.7952 0.7941 0.7995 0.7618  
## Silhouette 0.6934 0.5859 0.5285 0.4934 0.4630 0.4570  
## kmeans Connectivity 0.0000 2.9290 2.9290 8.7869 20.3361 20.3361  
## Dunn 1.1787 1.1159 0.7952 0.6475 0.6207 0.6207  
## Silhouette 0.6934 0.5859 0.5285 0.4931 0.4431 0.4403  
## pam Connectivity 0.0000 0.0000 4.3825 7.3115 10.2405 13.1694  
## Dunn 1.1787 0.4739 0.3400 0.5603 0.5603 0.5835  
## Silhouette 0.6934 0.5198 0.4699 0.4838 0.4487 0.4234  
##   
## Optimal Scores:  
##   
## Score Method Clusters  
## Connectivity 0.0000 hierarchical 2   
## Dunn 1.1787 hierarchical 2   
## Silhouette 0.6934 hierarchical 2

optimalScores(intcheck\_LmOB2)

## Score Method Clusters  
## Connectivity 0.0000000 hierarchical 2  
## Dunn 1.1786937 hierarchical 2  
## Silhouette 0.6933805 hierarchical 2

#Plot the Results  
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))  
plot(intcheck\_LmOB2, legend = FALSE)  
plot(nClusters(intcheck\_LmOB2), measures(intcheck\_LmOB2, "Dunn")[, , 1], type = "n",   
 axes = F, xlab = "", ylab = "")  
legend("center", clusterMethods(intcheck\_LmOB2), col = 1:3, lty = 1:3, pch = paste(1:3))



Based on the optimal scores for the internal validation results for the Outbreak 1 dataset, it appears that hierarchical clustering with k=2 clusters is a more suitable algorithm than *k*-means or PAM. The connectivity is minimized, whereas the Dunn Index and Silhouette Width are maximized. This suggests that the hierarchical clustering method with k=2 clusters generates clusters that are compact, well-separated and demonstrate optimal connectivity.

*Sporadic*

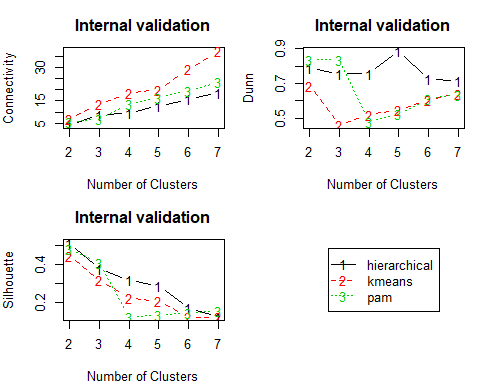
#Internal Validity Check for 2 to 7 clusters for 3 clustering methods  
intcheck\_LmSP = clValid(LmSP, nClust = 2:7, clMethods = c("hierarchical", "kmeans", "pam"), validation = "internal")  
  
#View summary results and optimal scores  
summary(intcheck\_LmSP)

##   
## Clustering Methods:  
## hierarchical kmeans pam   
##   
## Cluster sizes:  
## 2 3 4 5 6 7   
##   
## Validation Measures:  
## 2 3 4 5 6 7  
##   
## hierarchical Connectivity 4.1258 8.7159 9.7159 12.6448 15.7849 18.7139  
## Dunn 0.7905 0.7559 0.7559 0.8889 0.7272 0.7158  
## Silhouette 0.5159 0.3796 0.3191 0.2905 0.1703 0.1233  
## kmeans Connectivity 7.1147 13.5238 18.4290 19.8901 29.1012 37.4016  
## Dunn 0.6882 0.4590 0.5176 0.5488 0.6060 0.6423  
## Silhouette 0.4475 0.3190 0.2257 0.2087 0.1232 0.1236  
## pam Connectivity 5.6857 6.7869 13.5369 16.4659 19.3948 23.4627  
## Dunn 0.8376 0.8376 0.4806 0.5250 0.6115 0.6344  
## Silhouette 0.4850 0.4086 0.1242 0.1353 0.1463 0.1573  
##   
## Optimal Scores:  
##   
## Score Method Clusters  
## Connectivity 4.1258 hierarchical 2   
## Dunn 0.8889 hierarchical 5   
## Silhouette 0.5159 hierarchical 2

optimalScores(intcheck\_LmSP)

## Score Method Clusters  
## Connectivity 4.1257937 hierarchical 2  
## Dunn 0.8888771 hierarchical 5  
## Silhouette 0.5159272 hierarchical 2

#Plot the Results  
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))  
plot(intcheck\_LmSP, legend = FALSE)  
plot(nClusters(intcheck\_LmSP), measures(intcheck\_LmSP, "Dunn")[, , 1], type = "n",   
 axes = F, xlab = "", ylab = "")  
legend("center", clusterMethods(intcheck\_LmSP), col = 1:3, lty = 1:3, pch = paste(1:3))



According to the results for the Sporadic Dataset, hierarchical also appears to be the most suitable clustering method, however, there is not an obvious consensus on the optimal number of clusters. This result is to be expected as the dataset does not contain any clusters. Although the connectivity measure suggests an optimal number of two clusters for the hierarchical clustering algorithm, the reported value is high (i.e., >4). Since the Dunn Index might be the most optimal score reported (0.8888), a cluster size of k=5 may be the most suitable. To determine what would happen if the number of clusters was modified, k was increased to k=25 for the Sporadic dataset. This resulted in an increase in the Dunn Index, while the other metrics remained relatively constant. In contrast, as the number of clusters (k) was increased for Outbreak 1, all values remained relatively constant as the number of optimal clusters was likely already detected.

## Compare clValid results to other R packages

Although not the primary objective of the study, the results reported by *clValid* for each of the validation measures investigated were compared to two commonly used R packages for cluster validation, fpc [6] and factoextra [7] to determine if similar results would be generated. More specifically, this step was conducted to address the inherent issue associated with cluster validation in that a clustering algorithm that performs well for one validation index, does not necessarily imply that it will perform equally as well for another validation metric [14].

Of note, the eclust function implements a default bootstrapping value of 100 (i.e., nboot = 100), which was maintained throughout the analyses below. To ensure replication of the results, a set has been set for each of the R code chunks below. As well, it is important to note the cluster.stats function returns two Dunn Indices, however, the value of interest for the purpose of this study corresponds to “dunn”, which is analogous to the Dunn Index reported by *clValid*.

#View the manual page for eclust for additional details  
?eclust

## starting httpd help server ... done

#View the manual page for the cluster.stats function for more details  
?cluster.stats

*Outbreak 1*

set.seed(1234)  
  
#Assess the K-Means Clustering Algorithm using the eclust function  
LmOB2\_km = eclust(LmOB2, FUNcluster = "kmeans", k = NULL, k.max = 7, hc\_metric = "euclidean", graph = FALSE)  
#k = NULL -> optimal number of clusters is determined by the gap statistic  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
km\_LmOB2stats = cluster.stats(dist(LmOB2), LmOB2\_km$cluster)  
  
#Results Summary  
#Number of Clusters = 2  
#Cluster Size = 43, 31  
#Silhouette Width = 0.2499  
#Dunn Index = 0.3038  
  
#View the number of observations in each cluster  
LmOB2\_km$cluster

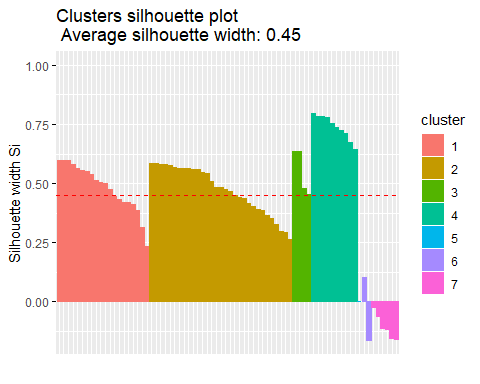
## 08-4802 08-4805 08-5243 08-5319 08-5371 08-5372   
## 2 2 1 1 2 2   
## 08-5373 08-5376 08-5377 08-5378 08-5489 08-5577   
## 2 2 7 7 2 5   
## 08-5578 08-5580 08-5581 08-5583 08-5703 08-5756   
## 1 2 2 2 2 7   
## 08-5758 08-5759 08-5761 08-5826 08-5827 08-5828   
## 2 1 1 7 7 2   
## 08-5871 08-5924 08-6030 08-6031 08-6032 08-6033   
## 1 1 1 2 1 1   
## 08-6056 08-6069 08-6071 08-6132 08-6133 08-6134   
## 1 1 2 2 1 1   
## 08-6135 08-6136 08-6137 08-6211 08-6306 08-6308   
## 1 2 2 2 2 2   
## 08-6309 08-6311 08-6335 08-6336 08-6337 08-6406   
## 2 2 7 2 2 2   
## 08-6421 08-6537 08-6569 08-6706 08-6997 08-7006   
## 1 2 1 1 2 2   
## 08-7007 08-7140 08-7275 08-7373 08-7374 08-7547   
## 2 1 2 3 3 3   
## 10-0814 10-0815 15-8753 15-9019 15C1046165 15S0018124   
## 3 1 4 4 6 6   
## 16-0117 16-0532 OLF16006-1 OLF16007-1 OLF16008-1 OLF16008-2   
## 4 4 4 4 4 4   
## OLF16009-3 OLF16009-5   
## 4 4

#Check the Silhouette Values  
LmOB2\_km$silinfo

## $widths  
## cluster neighbor sil\_width  
## 08-5243 1 3 0.59824454  
## 08-5319 1 3 0.59824454  
## 08-6706 1 3 0.59824454  
## 08-6421 1 3 0.58149488  
## 08-5924 1 3 0.56417635  
## 08-7140 1 3 0.55718736  
## 08-5761 1 3 0.55030854  
## 08-6135 1 3 0.53938173  
## 08-6134 1 3 0.51484802  
## 08-5759 1 3 0.50304395  
## 08-6069 1 7 0.50164417  
## 08-6030 1 3 0.47627139  
## 10-0815 1 3 0.44517293  
## 08-6056 1 3 0.43405853  
## 08-6569 1 3 0.42106975  
## 08-6133 1 7 0.41995523  
## 08-5578 1 3 0.41323855  
## 08-6033 1 3 0.38694275  
## 08-6032 1 3 0.31404145  
## 08-5871 1 3 0.23519570  
## 08-4802 2 1 0.58452141  
## 08-5371 2 1 0.58442195  
## 08-7275 2 1 0.58218166  
## 08-7006 2 1 0.58193891  
## 08-5581 2 1 0.57536558  
## 08-7007 2 1 0.56923764  
## 08-4805 2 1 0.56595863  
## 08-5373 2 1 0.56448955  
## 08-5828 2 1 0.56435866  
## 08-6308 2 1 0.55914838  
## 08-6136 2 1 0.55819419  
## 08-6309 2 1 0.54897384  
## 08-6406 2 1 0.54483177  
## 08-6137 2 1 0.51023669  
## 08-5703 2 1 0.48538803  
## 08-5372 2 1 0.48277018  
## 08-6997 2 1 0.47709283  
## 08-5376 2 1 0.46664484  
## 08-5758 2 1 0.44951524  
## 08-6306 2 1 0.43999619  
## 08-6132 2 1 0.43910562  
## 08-6311 2 1 0.41514073  
## 08-6031 2 1 0.40544562  
## 08-6537 2 1 0.39029287  
## 08-5580 2 1 0.38512716  
## 08-6211 2 1 0.36427102  
## 08-5489 2 1 0.35302373  
## 08-6337 2 1 0.32749018  
## 08-6071 2 1 0.29975186  
## 08-5583 2 1 0.29146375  
## 08-6336 2 7 0.26414153  
## 08-7373 3 1 0.63518935  
## 08-7547 3 1 0.63518935  
## 10-0814 3 1 0.47954983  
## 08-7374 3 1 0.45553547  
## 15-9019 4 6 0.79742038  
## OLF16009-3 4 6 0.78541651  
## 16-0117 4 6 0.78313055  
## OLF16009-5 4 6 0.78016644  
## OLF16008-2 4 6 0.75278381  
## 16-0532 4 6 0.73652376  
## OLF16007-1 4 6 0.72486878  
## 15-8753 4 6 0.71248002  
## OLF16006-1 4 6 0.67366008  
## OLF16008-1 4 6 0.64454043  
## 08-5577 5 7 0.00000000  
## 15S0018124 6 4 0.10248220  
## 15C1046165 6 4 -0.16354570  
## 08-5756 7 1 -0.02534493  
## 08-5377 7 1 -0.06315825  
## 08-5826 7 1 -0.11087528  
## 08-6335 7 1 -0.11490104  
## 08-5378 7 1 -0.15265387  
## 08-5827 7 1 -0.15782720  
##   
## $clus.avg.widths  
## [1] 0.48263825 0.47195227 0.55136600 0.73909908 0.00000000 -0.03053175  
## [7] -0.10412676  
##   
## $avg.width  
## [1] 0.4485664

#Plot Sil Width  
fviz\_silhouette(LmOB2\_km)

## cluster size ave.sil.width  
## 1 1 20 0.48  
## 2 2 31 0.47  
## 3 3 4 0.55  
## 4 4 10 0.74  
## 5 5 1 0.00  
## 6 6 2 -0.03  
## 7 7 6 -0.10

 The plot in the knitted version is incorrect; please refer to

Also, the results output generated by the code chunk above is incorrect in the knitted version. When reviewing the results, please refer to the results generated within the current R markdown file (although a brief summary has been provided in the code above).

Using the eclust and cluster.stats functions, a kmeans clustering with k=2 clusters was considered to be optimal. However, cluster 1 contains a mixture of observations from both outbreaks while cluster 2 only contains observations from the outbreak associated with RTE. Additionally, the average Silhouette Width is lower (0.2499) than the Silhouette Width reported by clValid for the same clustering parameters (0.6934). This difference is likely attributed to a group of isolates with negative Silhouette values identified using the eclust function, which correspond to the observations associated with the RTE outbreak that have been placed in the cluster with observations associated with a leafy green outbreak. These negative values suggest the observations are most likely placed in the wrong cluster, which is consistent with the ground truth. It is possible that the Silhouette Width in the clValid package did not account for these misclassifications.

set.seed(1234)  
  
#Assess the PAM Clustering Algorithm  
LmOB2\_pam = eclust(LmOB2, FUNcluster = "pam", k = NULL, k.max = 7, hc\_metric = "euclidean", graph = FALSE)  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
pam\_LmOB2stats = cluster.stats(dist(LmOB2), LmOB2\_pam$clustering)  
  
  
#Results Summary  
#Number of Clusters = 6  
#Cluster Size = 31, 26, 1, 4, 11, 1  
#Silhouette Width = 0.4487  
#Dunn Index = 0.5603  
  
#View the cluster number for each observation  
LmOB2\_pam$clustering

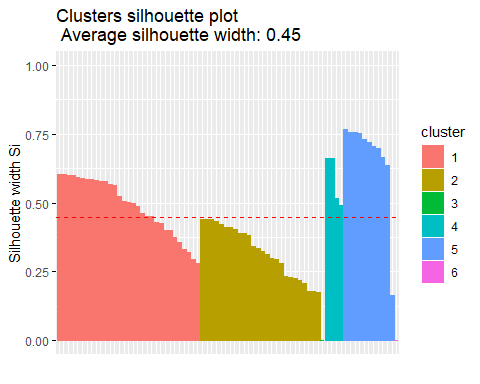
## 08-4802 08-4805 08-5243 08-5319 08-5371 08-5372   
## 1 1 2 2 1 1   
## 08-5373 08-5376 08-5377 08-5378 08-5489 08-5577   
## 1 1 2 2 1 3   
## 08-5578 08-5580 08-5581 08-5583 08-5703 08-5756   
## 2 1 1 1 1 2   
## 08-5758 08-5759 08-5761 08-5826 08-5827 08-5828   
## 1 2 2 2 2 1   
## 08-5871 08-5924 08-6030 08-6031 08-6032 08-6033   
## 2 2 2 1 2 2   
## 08-6056 08-6069 08-6071 08-6132 08-6133 08-6134   
## 2 2 1 1 2 2   
## 08-6135 08-6136 08-6137 08-6211 08-6306 08-6308   
## 2 1 1 1 1 1   
## 08-6309 08-6311 08-6335 08-6336 08-6337 08-6406   
## 1 1 2 1 1 1   
## 08-6421 08-6537 08-6569 08-6706 08-6997 08-7006   
## 2 1 2 2 1 1   
## 08-7007 08-7140 08-7275 08-7373 08-7374 08-7547   
## 1 2 1 4 4 4   
## 10-0814 10-0815 15-8753 15-9019 15C1046165 15S0018124   
## 4 2 5 5 5 6   
## 16-0117 16-0532 OLF16006-1 OLF16007-1 OLF16008-1 OLF16008-2   
## 5 5 5 5 5 5   
## OLF16009-3 OLF16009-5   
## 5 5

#Check the Silhouette Values  
LmOB2\_pam$silinfo

## $widths  
## cluster neighbor sil\_width  
## 08-4802 1 2 0.6031405  
## 08-5371 1 2 0.6030436  
## 08-7275 1 2 0.6008894  
## 08-7006 1 2 0.6004124  
## 08-5581 1 2 0.5938471  
## 08-7007 1 2 0.5881203  
## 08-4805 1 2 0.5852064  
## 08-5373 1 2 0.5837669  
## 08-5828 1 2 0.5822797  
## 08-6308 1 2 0.5766811  
## 08-6136 1 2 0.5765207  
## 08-6309 1 2 0.5685360  
## 08-6406 1 2 0.5628332  
## 08-6137 1 2 0.5246589  
## 08-5372 1 2 0.5053102  
## 08-5703 1 2 0.5001716  
## 08-6997 1 2 0.4990524  
## 08-5376 1 2 0.4889182  
## 08-5758 1 2 0.4620518  
## 08-6132 1 2 0.4522967  
## 08-6306 1 2 0.4515614  
## 08-6031 1 2 0.4293419  
## 08-6311 1 2 0.4253421  
## 08-6537 1 2 0.4009417  
## 08-5580 1 2 0.4007564  
## 08-6211 1 2 0.3729870  
## 08-5489 1 2 0.3568552  
## 08-6337 1 2 0.3327128  
## 08-6071 1 2 0.3210994  
## 08-5583 1 2 0.2933373  
## 08-6336 1 2 0.2785299  
## 08-5243 2 4 0.4412379  
## 08-5319 2 4 0.4412379  
## 08-6706 2 4 0.4412379  
## 08-6421 2 4 0.4315701  
## 08-5924 2 4 0.4203206  
## 08-5761 2 4 0.4118230  
## 08-7140 2 4 0.4106938  
## 08-6135 2 4 0.4039258  
## 08-6134 2 4 0.3884104  
## 08-6069 2 4 0.3876626  
## 08-5759 2 4 0.3805165  
## 08-6030 2 4 0.3433862  
## 08-6133 2 4 0.3357694  
## 10-0815 2 4 0.3249557  
## 08-6056 2 4 0.3115926  
## 08-6569 2 4 0.2966303  
## 08-5578 2 4 0.2965545  
## 08-6033 2 4 0.2815017  
## 08-5827 2 4 0.2332463  
## 08-6032 2 4 0.2309817  
## 08-5378 2 4 0.2272115  
## 08-6335 2 4 0.2194280  
## 08-5826 2 4 0.2069352  
## 08-5377 2 4 0.1768829  
## 08-5871 2 4 0.1768428  
## 08-5756 2 4 0.1737023  
## 08-5577 3 2 0.0000000  
## 08-7373 4 2 0.6631255  
## 08-7547 4 2 0.6631255  
## 10-0814 4 2 0.5160281  
## 08-7374 4 2 0.4918642  
## 15-9019 5 6 0.7677048  
## OLF16009-3 5 6 0.7578072  
## 16-0117 5 6 0.7562144  
## OLF16009-5 5 6 0.7519941  
## OLF16008-2 5 6 0.7311142  
## 16-0532 5 6 0.7181148  
## OLF16007-1 5 6 0.7067977  
## 15-8753 5 6 0.6985368  
## OLF16006-1 5 6 0.6661648  
## OLF16008-1 5 6 0.6370689  
## 15C1046165 5 6 0.1635457  
## 15S0018124 6 5 0.0000000  
##   
## $clus.avg.widths  
## [1] 0.4877807 0.3228561 0.0000000 0.5835358 0.6686421 0.0000000  
##   
## $avg.width  
## [1] 0.4487117

#Plot Sil Width  
fviz\_silhouette(LmOB2\_pam)

## cluster size ave.sil.width  
## 1 1 31 0.49  
## 2 2 26 0.32  
## 3 3 1 0.00  
## 4 4 4 0.58  
## 5 5 11 0.67  
## 6 6 1 0.00



Using the eclust and cluster.stats functions, a PAM clustering with k=6 clusters was considered to be optimal. Although more clusters were identified, the observations associated with RTE meat (clusters 1-4) are clustered separately from observations associated with an exposure to leafy greens (clusters 5-6). The average Silhouette Width (0.4487) is also consistent with the average Silhouette Width reported by clValid for the PAM algorithm with 6 clusters (0.4487).

set.seed(1234)  
  
#Assess the Hierarchical Clustering Algorithm  
LmOB2\_hc = eclust(LmOB2, FUNcluster = "hclust", k = NULL, k.max = 7, hc\_metric = "euclidean", hc\_method = "average", graph = FALSE)  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
hc\_LmOB2stats = cluster.stats(dist(LmOB2), LmOB2\_hc$cluster)  
  
  
  
#Results Summary  
#Number of Clusters = 7  
#Cluster Size = 29, 30, 1, 1, 1, 11, 1  
#Silhouette Width = 0.4570  
#Dunn Index = 0.7618  
  
#View the cluster number for each observation  
LmOB2\_hc$cluster

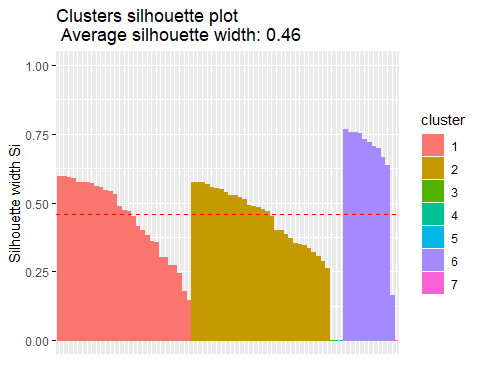
## 08-4802 08-4805 08-5243 08-5319 08-5371 08-5372   
## 1 1 2 2 1 1   
## 08-5373 08-5376 08-5377 08-5378 08-5489 08-5577   
## 1 1 2 2 1 3   
## 08-5578 08-5580 08-5581 08-5583 08-5703 08-5756   
## 2 1 1 4 1 2   
## 08-5758 08-5759 08-5761 08-5826 08-5827 08-5828   
## 1 2 2 2 2 1   
## 08-5871 08-5924 08-6030 08-6031 08-6032 08-6033   
## 2 2 2 1 2 2   
## 08-6056 08-6069 08-6071 08-6132 08-6133 08-6134   
## 2 2 1 1 2 2   
## 08-6135 08-6136 08-6137 08-6211 08-6306 08-6308   
## 2 1 1 1 1 1   
## 08-6309 08-6311 08-6335 08-6336 08-6337 08-6406   
## 1 1 2 5 1 1   
## 08-6421 08-6537 08-6569 08-6706 08-6997 08-7006   
## 2 1 2 2 1 1   
## 08-7007 08-7140 08-7275 08-7373 08-7374 08-7547   
## 1 2 1 2 2 2   
## 10-0814 10-0815 15-8753 15-9019 15C1046165 15S0018124   
## 2 2 6 6 6 7   
## 16-0117 16-0532 OLF16006-1 OLF16007-1 OLF16008-1 OLF16008-2   
## 6 6 6 6 6 6   
## OLF16009-3 OLF16009-5   
## 6 6

#Check the Silhouette Values  
LmOB2\_hc$silinfo

## $widths  
## cluster neighbor sil\_width  
## 08-4802 1 4 0.5957631  
## 08-5371 1 4 0.5957069  
## 08-7275 1 4 0.5931028  
## 08-7006 1 4 0.5899381  
## 08-7007 1 4 0.5755756  
## 08-4805 1 4 0.5752715  
## 08-5373 1 4 0.5732475  
## 08-5581 1 4 0.5724115  
## 08-5828 1 4 0.5598362  
## 08-6136 1 4 0.5557878  
## 08-6309 1 4 0.5467233  
## 08-6308 1 4 0.5406100  
## 08-6406 1 4 0.5311015  
## 08-5372 1 4 0.4874964  
## 08-6997 1 4 0.4744234  
## 08-5376 1 4 0.4679888  
## 08-6137 1 4 0.4525715  
## 08-5703 1 4 0.4141779  
## 08-6031 1 4 0.4011214  
## 08-6132 1 5 0.3811434  
## 08-5758 1 4 0.3607523  
## 08-6306 1 4 0.3567084  
## 08-6311 1 5 0.3036957  
## 08-5580 1 4 0.3013940  
## 08-6537 1 4 0.2721439  
## 08-6071 1 4 0.2715514  
## 08-6211 1 4 0.2424322  
## 08-5489 1 5 0.1798745  
## 08-6337 1 5 0.1463961  
## 08-5243 2 1 0.5732700  
## 08-5319 2 1 0.5732700  
## 08-6706 2 1 0.5732700  
## 08-6421 2 1 0.5663410  
## 08-5924 2 1 0.5573216  
## 08-7140 2 1 0.5537141  
## 08-5761 2 1 0.5500567  
## 08-6135 2 1 0.5385937  
## 08-6134 2 1 0.5282648  
## 08-6069 2 1 0.5268682  
## 08-5759 2 1 0.5193151  
## 08-6030 2 1 0.5114464  
## 08-6133 2 1 0.4916737  
## 10-0815 2 1 0.4862603  
## 08-6569 2 1 0.4829916  
## 08-6056 2 1 0.4760549  
## 08-5578 2 1 0.4682681  
## 08-6033 2 1 0.4503758  
## 08-7373 2 1 0.4013175  
## 08-7547 2 1 0.4013175  
## 08-6032 2 1 0.3855834  
## 10-0814 2 1 0.3702622  
## 08-7374 2 1 0.3524416  
## 08-5827 2 1 0.3475477  
## 08-5378 2 1 0.3464957  
## 08-5826 2 1 0.3350281  
## 08-6335 2 1 0.3184696  
## 08-5871 2 1 0.3071580  
## 08-5377 2 1 0.2866841  
## 08-5756 2 1 0.2619432  
## 08-5577 3 2 0.0000000  
## 08-5583 4 1 0.0000000  
## 08-6336 5 1 0.0000000  
## 15-9019 6 7 0.7677048  
## OLF16009-3 6 7 0.7578072  
## 16-0117 6 7 0.7562144  
## OLF16009-5 6 7 0.7519941  
## OLF16008-2 6 7 0.7311142  
## 16-0532 6 7 0.7181148  
## OLF16007-1 6 7 0.7067977  
## 15-8753 6 7 0.6985368  
## OLF16006-1 6 7 0.6661648  
## OLF16008-1 6 7 0.6370689  
## 15C1046165 6 7 0.1635457  
## 15S0018124 7 6 0.0000000  
##   
## $clus.avg.widths  
## [1] 0.4454809 0.4513868 0.0000000 0.0000000 0.0000000 0.6686421 0.0000000  
##   
## $avg.width  
## [1] 0.4569678

#Plot the Silhouette Values  
fviz\_silhouette(LmOB2\_hc)

## cluster size ave.sil.width  
## 1 1 29 0.45  
## 2 2 30 0.45  
## 3 3 1 0.00  
## 4 4 1 0.00  
## 5 5 1 0.00  
## 6 6 11 0.67  
## 7 7 1 0.00



Using the eclust function, a hierarchical clustering with k=7 clusters was considered to be optimal, which was not in agreement with the clValid results (k=2) for the same dataset. Although the observations were split into many more clusters (i.e., 7) with the eclust function in comparison to clValid, the observations associated with the RTE Meat Outbreak are clustered separately (clusters 1-5) from the observations associated with the Leafy Green Outbreak (clusters 6-7). Additionally, the average Silhouette Width (0.4570) is consistent with the average Silhouette Width reported by *clValid* for the hierarchical algorithm with 7 clusters (0.4570).

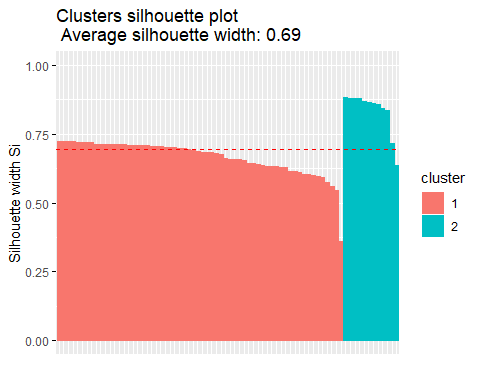
To determine whether the eclust and cluster.stats functions return comparable values to those reported by *clValid* for the optimal clustering algorithm and number of clusters (i.e., hierarchical, k=2), the Outbreak 1 dataset was re-evaluated with the number of clusters (k) adjusted to k=2.

set.seed(1234)  
  
###Re-evaluate Outbreak 1 using the optimal number of cluster identified by clValid for the hierarchical clustering algorithm  
  
#Assess the Hierarchical Clustering Algorithm with k=2  
LmOB2\_hc2 = eclust(LmOB2, FUNcluster = "hclust", k = 2, k.max = 7, hc\_metric = "euclidean", hc\_method = "average", graph = FALSE)  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
hc\_LmOB2stats2 = cluster.stats(dist(LmOB2), LmOB2\_hc2$cluster)  
  
  
#Results Summary  
#Number of Clusters = 2  
#Cluster Size = 62, 12  
#Silhouette Width = 0.6934  
#Dunn Index = 1.1787  
  
  
#View the cluster number for each observation  
LmOB2\_hc2$cluster

## 08-4802 08-4805 08-5243 08-5319 08-5371 08-5372   
## 1 1 1 1 1 1   
## 08-5373 08-5376 08-5377 08-5378 08-5489 08-5577   
## 1 1 1 1 1 1   
## 08-5578 08-5580 08-5581 08-5583 08-5703 08-5756   
## 1 1 1 1 1 1   
## 08-5758 08-5759 08-5761 08-5826 08-5827 08-5828   
## 1 1 1 1 1 1   
## 08-5871 08-5924 08-6030 08-6031 08-6032 08-6033   
## 1 1 1 1 1 1   
## 08-6056 08-6069 08-6071 08-6132 08-6133 08-6134   
## 1 1 1 1 1 1   
## 08-6135 08-6136 08-6137 08-6211 08-6306 08-6308   
## 1 1 1 1 1 1   
## 08-6309 08-6311 08-6335 08-6336 08-6337 08-6406   
## 1 1 1 1 1 1   
## 08-6421 08-6537 08-6569 08-6706 08-6997 08-7006   
## 1 1 1 1 1 1   
## 08-7007 08-7140 08-7275 08-7373 08-7374 08-7547   
## 1 1 1 1 1 1   
## 10-0814 10-0815 15-8753 15-9019 15C1046165 15S0018124   
## 1 1 2 2 2 2   
## 16-0117 16-0532 OLF16006-1 OLF16007-1 OLF16008-1 OLF16008-2   
## 2 2 2 2 2 2   
## OLF16009-3 OLF16009-5   
## 2 2

#Plot Sil Width  
fviz\_silhouette(LmOB2\_hc2)

## cluster size ave.sil.width  
## 1 1 62 0.67  
## 2 2 12 0.83



Following an adjustment of the number of clusters to k=2 for the hierarchical clustering algorithm, the summary statistics returned by the eclust function and the cluster.stats function were nearly identical to those obtained by the *clValid* package. Thus, the results suggest that k=2 leads to a more appropriate clustering of the observations according to the outbreak they are associated with.

*Sporadic*

set.seed(1234)  
  
#Assess the K-Means Clustering Algorithm using the eclust function  
LmSP\_km = eclust(LmSP, FUNcluster = c("kmeans"), k = NULL, k.max = 7, hc\_metric = "euclidean", graph = FALSE)  
#k = NULL -> optimal number of clusters is determined by the gap statistic  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
km\_LmSPstats = cluster.stats(dist(LmSP), LmSP\_km$cluster)  
  
  
#Number of Clusters = 4  
#Cluster Size = 9, 2, 2, 13  
#Silhouette Width = 0.1362  
#Dunn Index = 0.4928  
  
  
#View the number of observations in each cluster  
LmSP\_km$cluster

## 18-6485 18-6564 18-6748 18-7068 18-7785 18-7786   
## 2 3 4 1 3 4   
## 18-7846 18-8104 18-9504 18-9557 18PF1261 18PF1300   
## 2 4 4 4 4 4   
## 18PF1301 18PF1329 18PF1431 18PF1432 18PF1510 18PF1511   
## 1 2 4 2 1 4   
## 18PF1547 OLF18117-2 OLF18132-3 OLF18135 OLF18136-1 OLF18147-2   
## 4 4 4 4 1 4   
## OLF18173 OLF18174   
## 4 4

#Check the Silhouette Values  
LmSP\_km$silinfo

## $widths  
## cluster neighbor sil\_width  
## 18-7068 1 3 0.080407440  
## 18PF1510 1 4 0.060917622  
## 18PF1301 1 4 -0.056854765  
## OLF18136-1 1 4 -0.167378070  
## 18-7846 2 4 0.343795888  
## 18PF1329 2 4 0.334876791  
## 18PF1432 2 4 0.312588556  
## 18-6485 2 4 0.308820890  
## 18-7785 3 1 0.649596685  
## 18-6564 3 1 0.642411530  
## 18-7786 4 2 0.151785250  
## OLF18117-2 4 2 0.136909383  
## 18PF1261 4 2 0.134132057  
## 18-9557 4 2 0.130619127  
## 18-6748 4 2 0.126811553  
## 18PF1511 4 2 0.111383393  
## OLF18132-3 4 2 0.103232850  
## OLF18173 4 2 0.100433505  
## OLF18147-2 4 2 0.082785807  
## 18-8104 4 2 0.082017897  
## 18-9504 4 2 0.066685398  
## 18PF1547 4 2 0.050520248  
## 18PF1300 4 2 0.049221960  
## OLF18135 4 2 0.041038688  
## OLF18174 4 2 0.003113616  
## 18PF1431 4 2 -0.023441680  
##   
## $clus.avg.widths  
## [1] -0.02072694 0.32502053 0.64600411 0.08420307  
##   
## $avg.width  
## [1] 0.1483243

Using the eclust function, a kmeans clustering with k=4 clusters was considered to be optimal, however, multiple unrelated observations are included in the same cluster. Not surprisingly, the average Silhouette Width is slightly lower (0.1362) than the Silhouette Width reported by clValid (0.2257) for the same clustering parameters. This difference is likely attributed to a group of isolates with negative Silhouette values; this suggests these isolates are most likely placed in the wrong cluster, which is consistent with the ground truth. Given the discrepancy, it is possible that the Silhouette Width in the clValid package did not “catch”/ account for these misclassifications.

set.seed(1234)  
  
#Assess the PAM Clustering Algorithm  
LmSP\_pam = eclust(LmSP, FUNcluster = c("pam"), k = NULL, k.max = 7, hc\_metric = "euclidean", graph = FALSE)  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
pam\_LmSPstats = cluster.stats(dist(LmSP), LmSP\_pam$clustering)  
  
  
  
#Number of Clusters = 2  
#Cluster Size = 23, 3  
#Silhouette Width = 0.4850  
#Dunn Index = 0.8376  
  
  
#View the cluster number for each observation  
LmSP\_pam$clustering

## 18-6485 18-6564 18-6748 18-7068 18-7785 18-7786   
## 1 2 1 2 2 1   
## 18-7846 18-8104 18-9504 18-9557 18PF1261 18PF1300   
## 1 1 1 1 1 1   
## 18PF1301 18PF1329 18PF1431 18PF1432 18PF1510 18PF1511   
## 1 1 1 1 1 1   
## 18PF1547 OLF18117-2 OLF18132-3 OLF18135 OLF18136-1 OLF18147-2   
## 1 1 1 1 1 1   
## OLF18173 OLF18174   
## 1 1

#Check the Silhouette Values  
LmSP\_pam$silinfo

## $widths  
## cluster neighbor sil\_width  
## OLF18135 1 2 0.60382959  
## 18PF1300 1 2 0.60154138  
## OLF18132-3 1 2 0.59410902  
## 18-9504 1 2 0.58758344  
## 18PF1432 1 2 0.58156679  
## 18-7846 1 2 0.57848672  
## OLF18174 1 2 0.57489476  
## 18PF1329 1 2 0.55957126  
## 18-9557 1 2 0.55712611  
## 18-8104 1 2 0.55273031  
## OLF18173 1 2 0.55181017  
## OLF18117-2 1 2 0.53297873  
## 18-6485 1 2 0.52809722  
## 18PF1547 1 2 0.52179898  
## 18PF1431 1 2 0.51666797  
## OLF18147-2 1 2 0.51636490  
## 18PF1511 1 2 0.51324067  
## 18PF1261 1 2 0.51080244  
## 18-7786 1 2 0.47040200  
## 18-6748 1 2 0.40797986  
## OLF18136-1 1 2 0.29022877  
## 18PF1301 1 2 0.26990534  
## 18PF1510 1 2 0.10662187  
## 18-7785 2 1 0.50591461  
## 18-6564 2 1 0.50177651  
## 18-7068 2 1 0.07476194  
##   
## $clus.avg.widths  
## [1] 0.5012321 0.3608177  
##   
## $avg.width  
## [1] 0.4850304

Using the eclust and cluster.stats functions, a PAM clustering with k=2 clusters was considered to be optimal, however, unrelated observations are grouped together, with a majority of observations contained in a single cluster (i.e., cluster 1). The average Silhouette Width (0.4487) is consistent with the average Silhouette Width reported by clValid for the same clustering parameters (0.4487).

set.seed(1234)  
  
#Assess the Hierarchical Clustering Algorithm  
LmSP\_hc = eclust(LmSP, FUNcluster = c("hclust"), k = NULL, k.max = 7, hc\_metric = "euclidean", hc\_method = "average", graph = FALSE)  
  
#View the cluster validation statistics using the cluster.stats function in the fpc R package  
hc\_LmSPstats = cluster.stats(dist(LmSP), LmSP\_hc$cluster)  
  
  
#Results Summary  
#Number of Clusters = 5  
#Cluster Size = 21, 2, 1, 1, 1  
#Silhouette Width = 0.2905  
#Dunn Index = 0.8889  
  
  
#View the cluster number for each observation  
LmSP\_hc$cluster

## 18-6485 18-6564 18-6748 18-7068 18-7785 18-7786   
## 1 2 1 3 2 1   
## 18-7846 18-8104 18-9504 18-9557 18PF1261 18PF1300   
## 1 1 1 1 1 1   
## 18PF1301 18PF1329 18PF1431 18PF1432 18PF1510 18PF1511   
## 4 1 1 1 5 1   
## 18PF1547 OLF18117-2 OLF18132-3 OLF18135 OLF18136-1 OLF18147-2   
## 1 1 1 1 1 1   
## OLF18173 OLF18174   
## 1 1

#Check the Silhouette Values  
LmSP\_hc$silinfo #Average Silhouette Width (0.2905) is consistent with the average Silhouette Width reported by clValid (0.2905).

## $widths  
## cluster neighbor sil\_width  
## 18PF1300 1 4 0.39578057  
## OLF18135 1 4 0.39319892  
## 18PF1432 1 4 0.39063307  
## 18-7846 1 4 0.38780145  
## OLF18132-3 1 4 0.37140388  
## 18-9504 1 4 0.37031763  
## OLF18174 1 4 0.36635819  
## 18PF1329 1 4 0.36219619  
## 18-6485 1 4 0.33162241  
## 18-8104 1 4 0.32638476  
## 18-9557 1 4 0.32151818  
## 18PF1431 1 4 0.30347772  
## OLF18117-2 1 4 0.27936539  
## 18PF1547 1 4 0.27437287  
## OLF18173 1 4 0.27279143  
## OLF18147-2 1 4 0.26245335  
## 18PF1261 1 4 0.25055527  
## 18PF1511 1 4 0.24293552  
## 18-7786 1 4 0.18483176  
## 18-6748 1 4 0.14930726  
## OLF18136-1 1 5 0.07815878  
## 18-7785 2 3 0.62191073  
## 18-6564 2 3 0.61636476  
## 18-7068 3 5 0.00000000  
## 18PF1301 4 1 0.00000000  
## 18PF1510 5 4 0.00000000  
##   
## $clus.avg.widths  
## [1] 0.3007364 0.6191377 0.0000000 0.0000000 0.0000000  
##   
## $avg.width  
## [1] 0.2905285

Using the eclust function, a hierarchical clustering with k=5 clusters was considered to be optimal, which was not in agreement with the clValid results for the same dataset. However, the average Silhouette Width (0.2905) was consistent with the average Silhouette Width reported by clValid for the same clustering parameters (i.e., k=5) (0.2905).

Since this dataset does not actually contain any clusters, the results were not re-evaluated with an adjusted k.

As well, it is important to note that only a range of 2 to 7 clusters was investigated in order to streamline computational requirements. However, if more clusters were permitted, it is possible that each isolate would eventually end up in its own cluster.

## Stability

Stability measures are used to compare the clustering results of the complete dataset to the clustering results after the removal of single column, which is consecutively repeated for all columns in the dataset [3]. If the clustering algorithm is suitable for the given dataset, the clusters should remain the same as stable clusters should be robust to random changes in the data [3].

The *clValid* package offers four stability measures including the average proportion of non-overlap (APN), the average distance (AD), the average distance between means (ADM), and the figure of merit (FOM), which are reported to be unique to the tool by the authors [3]. A brief description of each metric is included below.

**APN**

Measures the average proportion of observations that are not placed in the same cluster based on the clustering of the complete dataset and the clustering after the removal of a single column [3].

The APN is defined as follows:

* Where,
  + = total number of observations in the dataset
  + = total number of columns
  + = cluster containing observation based on the clustering of the complete dataset
  + = cluster containing observation based on the clustering after the removal of a single column

The APN results are reported in a range of 0 to 1, with values closer to 0 indicating a high degree of consistency in the clustering.

**AD**

Measures the average distance of observations placed in the same cluster based on the clustering of the complete dataset and the clustering after the removal of a single column [3].

The AD results are reported in a range of 0 to infinity, where smaller values indicate a better degree of clustering.

**ADM**

Measures the average distance between cluster centers based on the clustering of the complete dataset and the clustering after the removal of a single column. Of note, the ADM currently only implements the Euclidean distance [3].

The ADM is defined as follows:

* Where,
  + = mean of observations in the cluster containing observation for the clustering of the complete dataset
  + = mean of observations in the cluster containing observation for the clustering after the removal of a single column.

The ADM results are reported in a range of 0 and infinity, where smaller values indicate a better degree of clustering.

**FOM**

Measures the average intra-cluster variance of observations in the deleted column

**Stability Check**

Given the high computational demands of this particular validity measure (which the authors of *clValid* do not address in the publication), the R code was run in a conda environment on a high computing cluster hosted at the National Microbiology Laboratory (i.e., Waffles).

The R output files (scheckLmOB2.Rout and scheckLmSP.Rout) have been included in the <https://github.com/rumorej/ModernStatsforModernBiol.git> repository in the folder “Final Project”.

An example of the R code that could be run in RStudio (although time consuming) for each of the datasets is included below (but has been commented out).

*Outbreak 1*

#Stability Check for 2 to 7 clusters for 3 clustering methods  
stacheck\_LmOB2 = clValid(LmOB2, nClust = 2:7, clMethods = c("hierarchical", "kmeans", "pam"), validation = "stability")  
  
#View Optimal Results  
optimalScores(stacheck\_LmOB2)  
  
#Plot the Results  
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))  
plot(stabcheck\_LmOB2, legend = FALSE)  
plot(nClusters(stabcheck\_LmOB2), measures(stabcheck\_LmOB2, "Dunn")[, , 1], type = "n",   
 axes = F, xlab = "", ylab = "")  
legend("center", clusterMethods(stabcheck\_LmOB2), col = 1:3, lty = 1:3, pch = paste(1:3))

*Sporadic*

#Stability Check for 2 to 7 clusters for 3 clustering methods  
stacheck\_LmSP = clValid(LmSP, nClust = 2:7, clMethods = c("hierarchical", "kmeans", "pam"), validation = "stability")  
  
#View Optimal Results  
optimalScores(stacheck\_LmSP)  
  
  
#Plot the Results  
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))  
plot(stabcheck\_LmSP, legend = FALSE)  
plot(nClusters(stabcheck\_LmSP), measures(stabcheck\_LmSP, "Dunn")[, , 1], type = "n",   
 axes = F, xlab = "", ylab = "")  
legend("center", clusterMethods(stabcheck\_LmSP), col = 1:3, lty = 1:3, pch = paste(1:3))

Based on the results of the stability validation analysis obtained from file scheck\_LmOB1.Rout, both hierarchical with k=2 and pam with k=7 were reported to generate stable clusters. Although pam with k=7 was reported to be stable according to the AD (421.3878) and the FOM (2.2482), the results from the internal validation reported a low Silhouette Width and Dunn Index for this particular clustering solution. Therefore, it would not be considered appropriate for the dataset. Hierarchical with k=2 was also reported as the most stable clustering solution for the Sporadic dataset. A number of different combinations of clustering algorithms and cluster numbers were reported, however, k=2 was reported most consistently between the internal and stability validation measures. Again, had more clusters been evaluated for the Sporadic dataset, the results returned by *clValid* would also have likely improved.

## Bootstrapping

The stability results obtained from the *clValid* package for each of the three datasets was compared to the results obtained from the *clusterboot* function in the *fpc* R package to determine whether similar results are generated.

Briefly, bootstrapping, which encompasses repetitive re sampling of the data (with replacement), is an alternative way to evaluate the stability of a clustering algorithm.

More specifically, the clusterboot function uses the Jaccard coefficient to measure the similarity between the results obtained from the various instances of the bootstrapping method applied to the data) to determine which clusters are similar, and which are distinct. The mean value of the Jaccard coefficient over all the bootstrap iterations of the data is used to determine the cluster stability of each cluster in the original clustering [15].

Since this stability index is not the main focus of the current study, only a simplified example of the inner workings of the clusterboot function is provided below [15].

1. Cluster the data using the algorithm of interest
2. Draw a new dataset (of the same size as the original) by resampling the original dataset with replacement (i.e., some of the data points may show up more than once, and others not at all).
3. Cluster the new dataset using the same algorithm as above.
4. Find the most similar clusters between the original clustering and the new clustering (i.e., cluster with a maximum Jaccard coefficient).
5. If the maximum Jaccard coefficient is less than 0.5, the original cluster is considered to be dissolved (i.e., it didn’t show up in the new clustering). A dissolved cluster does not likely represent a “real” cluster.
6. Repeat steps 2-5 based on the number of times the bootstrapping method is to be applied (i.e., B=50).

The Jaccard Coefficient is reported on a scale of 0-1 (or 0%-100%), with higher values indicating higher stability. A more precise interpretation is provided below, which was obtained from the ?clusterboot help page in R Studio.

* Interpretation:
  + <0.60 = cluster is unstable and should not be trusted
  + 0.6-0.75 = cluster may be real, but however, the certainty of what observations belong to the cluster is not high
  + 0.75-0.84 = cluster is stable and likely represents a true cluster
  + greater than or equal to 0.85 = cluster is highly stable

To note, the datasets shown below have been assessed using a bootstrap value of 50 (i.e., the datasets are resampled 50 times).

*Outbreak 1*

#Run bootstrapping for kmeans  
LmOB2\_kmboot = clusterboot(LmOB2, B=50, distances=FALSE, clustermethod=kmeansCBI, seed=1234, krange=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmOB2\_kmboot

## \* Cluster stability assessment \*  
## Cluster method: kmeans   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 3   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 0.8643092 0.8834434 0.9950000  
## dissolved:  
## [1] 9 4 0  
## recovered:  
## [1] 38 38 49

#Run bootstrapping for PAM  
LmOB2\_pamboot = clusterboot(LmOB2, B=50, distances=FALSE, clustermethod=pamkCBI, seed=1234, krange=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmOB2\_pamboot

## \* Cluster stability assessment \*  
## Cluster method: pam/estimated k   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 2   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 1 1  
## dissolved:  
## [1] 0 0  
## recovered:  
## [1] 50 50

#Run bootstrapping for hierarchical  
LmOB2\_hcboot = clusterboot(LmOB2, B=50, distances=FALSE, clustermethod=hclustCBI, method="average", seed=1234, k=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmOB2\_hcboot

## \* Cluster stability assessment \*  
## Cluster method: hclust/cutree   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 7   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 1 1 0 0 0 0 0  
## dissolved:  
## [1] 0 0 50 50 50 50 50  
## recovered:  
## [1] 50 50 0 0 0 0 0

Using the clusterboot function to assess the stability of the three clustering algorithms for the Outbreak 1 dataset, a total of k=3 clusters were identified for *k*-means, with relatively high Jaccard coefficients for each of the clusters (i.e., 0.8268, 1, 0.8251), which suggests these clusters are quite stable. However, the observations associated with the RTE outbreak were observed to be split across two clusters. The pam clustering algorithm generated even more stable clusterings k=2, with perfect Jaccard Coefficients (1, 1). The observations were also clustered according to outbreak. A total of k=7 clusters were observed for the hierarchical clustering algorithm, however, only two clusterings were considered stable as indicated by the Jaccard Coefficient (1,1,0,0,0,0,0).

*Sporadic*

#Run bootstrapping for kmeans  
LmSP\_kmboot = clusterboot(LmSP, B=50, distances=FALSE, clustermethod=kmeansCBI, seed=1234, krange=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmSP\_kmboot

## \* Cluster stability assessment \*  
## Cluster method: kmeans   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 2   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 0.9595160 0.8843333  
## dissolved:  
## [1] 1 5  
## recovered:  
## [1] 49 38

#Run bootstrapping for PAM  
LmSP\_pamboot = clusterboot(LmSP, B=50, distances=FALSE, clustermethod=pamkCBI, seed=1234, krange=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmSP\_pamboot

## \* Cluster stability assessment \*  
## Cluster method: pam/estimated k   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 2   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 0.9934167 0.9400000  
## dissolved:  
## [1] 0 3  
## recovered:  
## [1] 50 47

#Run bootstrapping for hierarchical  
LmSP\_hcboot = clusterboot(LmSP, B=50, distances=FALSE, clustermethod=hclustCBI, method="average", seed=1234, k=2:7)

## boot 1   
## boot 2   
## boot 3   
## boot 4   
## boot 5   
## boot 6   
## boot 7   
## boot 8   
## boot 9   
## boot 10   
## boot 11   
## boot 12   
## boot 13   
## boot 14   
## boot 15   
## boot 16   
## boot 17   
## boot 18   
## boot 19   
## boot 20   
## boot 21   
## boot 22   
## boot 23   
## boot 24   
## boot 25   
## boot 26   
## boot 27   
## boot 28   
## boot 29   
## boot 30   
## boot 31   
## boot 32   
## boot 33   
## boot 34   
## boot 35   
## boot 36   
## boot 37   
## boot 38   
## boot 39   
## boot 40   
## boot 41   
## boot 42   
## boot 43   
## boot 44   
## boot 45   
## boot 46   
## boot 47   
## boot 48   
## boot 49   
## boot 50

#View the results  
LmSP\_hcboot

## \* Cluster stability assessment \*  
## Cluster method: hclust/cutree   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 50   
##   
## Number of clusters found in data: 7   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 0.9960833 0.9400000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
## dissolved:  
## [1] 0 3 50 50 50 50 50  
## recovered:  
## [1] 50 47 0 0 0 0 0

With respect to the Sporadic data, k=2 was found to be the most stable for both *k*-means and pam, with Jaccard Coefficients of 0.8930, 0.9621 and 0.9901, 0.9200, respectively. These high values suggest the *k*-means and pam clustering algorithms resulted in stable clusters when k=2. In contrast, k=7 was identified for hierarchical clustering, however, only two of the clusters appear to be stable based on the Jaccard Coefficient (0.9950, 0.9200, 0, 0, 0, 0, 0). In both cases, observations associated with different outbreaks have been placed in the same cluster, which we know is not optimal.

## Biological

As the metric would imply, biological validation measures assess whether the clustering algorithm is able to produce biologically meaningful clusters, or more simply put, do the cluster assignments agree with externally available information? According to the authors, the inclusion of biological validation metrics is also novel to the *clValid* package [3]. This particular validation measure is most appropriate for micro array or RNA seq data where observations corresponds to genes.

Two biological validation measures, the Biological Homogeneity Index (BHI) and Biological Stability Index (BSI) are available in *clValid* [3, 16]. A brief description of each metric is included below.

**BHI**

Measures how biologically homogeneous the clusters are with respect to the functional class (i.e., gene expression datasets). BHI values are in the range of 0 to 1 where larger values indicate more biologically homogeneous clusters [3, 14].

**BSI**

Measure the stability of clusters in the biological context. The stability of clusters containing isolates with genes demonstrating similar functional classes for the complete dataset is compared to the clustering of genes with similar functional classes where a single column has been removed [3, 16]. BSI values are in the range of 0 to 1 where larger values indicating more stable clusters [3, 16].

What is most interesting about the biological validation metrics in *clValid* is that they can utilize of a number of annotation packages available in Bioconductor [17-19] to map the genes in a given dataset to their corresponding gene ontology (GO)[20]. Therefore, the functional class of the genes does not have to be pre-determined.

Although the datasets investigated in this study can be analyzed using the biological validation metrics described with the “functional class”, representing the “group” to which the observations belong to (i.e., Outbreak), these metrics are considerably computationally demanding, and therefore, would require a significant amount of time to run with very little gain. An example of the R code that could be used to run the biological validation for the Outbreak 1 dataset is shown below (but has been commented out).

*Outbreak 1*

#Append Exposure information to rownames  
LmOB2\_exp = tapply(rownames(LmOB2), LmOB2\_data$Group, c)  
  
#Evaluate the biological validity  
biocheck2\_LmOB2 = clValid(LmOB2, nClust = 2:3, clMethods = c("hierarchical", "kmeans", "pam"), validation = "biological", annotation = LmOB2\_exp)

Given the nature of the datasets explored in the current study (i.e., allele data), conducting an external validation to assess the results of the three clustering algorithms investigated in this study in comparison to the ground truth would be most appropriate.

## External Validation

External validation is used to compare the results of a cluster analysis to an externally known result, which is most often represented in the form of a “class label” [4, 11].

Therefore, to assess the external validity of a cluster analysis, the ground truth with respect to the “clustering” of the data must be known (i.e., exposure), which unfortunately, isn’t always available. However, for the current study, the ground truth for the selected datasets is known, and thus, can be compared to the results of the cluster validation analyses that were previously generated.

To conduct the external validation, a cross-tabulation between the clustering algorithm (i.e., kmeans, PAM, or hierarchical) and the ground truth (i.e., Groups) for each of the datasets is first generated, followed by the calculation of the corrected Rand Index [4, 11]. To note, “Group” represents the correct class label, which corresponds to the exposure for each of the observations included in the dataset.

*Corrected Rand Index*

Briefly, the corrected, or more commonly, the Adjusted Rand Index (ARI), which is an extension of the Rand Index, is a statistical measure frequently used to quantify the similarity between two clustering, taking into account the potential for grouping of the data by “chance” [21]. The Adjusted Rand Index reports values between -1 (no agreement) and 1 (perfect agreement), where a large index is desired for externally valid clusters [11, 21, 22].

ARI =

* Where
  + a = pairs of observations that are similarily placed (i.e., in agreement) in the same cluster between clusterings
  + d = pairs of observations that are similarily placed (i.e., in agreement) in different clusters between clusterings
  + b = pairs of observations that are not similarily placed (i.e., disagreement) in the same cluster between clusterings
  + c = pairs of observations that are not similarily placed (i.e., disagreement) in different clusters between clusterings

*Outbreak 1*

For the Group Assignment for Outbreak 1: RTE = Ready-to-Eat

#Cross Tabulation for kmeans  
table(LmOB2\_data$Group, LmOB2\_km$cluster)

##   
## 1 2 3 4 5 6 7  
## LG 0 0 0 10 0 2 0  
## RTE 20 31 4 0 1 0 6

#Convert exposures to a numeric value  
LmOB2\_expnum = as.numeric(LmOB2\_data$Group)  
  
#Compute the Cluster Stats  
LmOB2\_kmstats = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_km$cluster)  
LmOB2\_kmstats

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 0.2436097  
##   
## $vi  
## [1] 1.062501

From the cross-tabulation table it is apparent that the observations associated with the RTE Meat Outbreak are placed in the wrong cluster. These results are not in agreement with the ground truth, which is supported by an extremely low corrected Rand Index (0.0072).

#Re-evaluate with k=3, which was determined to be most stable by the clusterboot function  
table(LmOB2\_data$Group, LmOB2\_kmboot$partition)

##   
## 1 2 3  
## LG 0 0 12  
## RTE 31 31 0

#Compute the Cluster Stats  
LmOB2\_kbootstats = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_kmboot$partition)  
LmOB2\_kbootstats

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 0.3634511  
##   
## $vi  
## [1] 0.5807449

The ARI (0.3635) slightly increased for the *k*-means cluster algorithm when k was increased to 3, which is likely due to the fact that the observations associated with each outbreak were clustered separately. However, the observations associated with the RTE outbreak were split across two clusters.

#Cross Tabulation for PAM  
table(LmOB2\_data$Group, LmOB2\_pam$clustering)

##   
## 1 2 3 4 5 6  
## LG 0 0 0 0 11 1  
## RTE 31 26 1 4 0 0

#Compute the Cluster Stats  
LmOB2\_pamstats = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_pam$clustering)  
LmOB2\_pamstats

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 0.2976984  
##   
## $vi  
## [1] 0.8461496

From the cross-tabulation table it is apparent that the observations are well separated according to exposure, however, the observations associated with the RTE outbreak are spread out across 4 clusters (i.e., clusters 1-4) and the observations associated with the leafy green outbreak are spread out over two clusters (i.e., clusters 4-6). These results are not in agreement with the ground truth, which is further supported by a low Adjusted Rand Index (0.2977).

#Re-evaluate with k=2, which was determined to be most stable by the clusterboot function  
#Cross Tabulation for PAM  
table(LmOB2\_data$Group, LmOB2\_pamboot$partition)

##   
## 1 2  
## LG 0 12  
## RTE 62 0

#Compute the Cluster Stats  
LmOB2\_pambootstats = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_pamboot$partition)  
LmOB2\_pambootstats

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 1  
##   
## $vi  
## [1] 0

When k=2 was run for the PAM clustering algorithm, perfect agreement between the clustering and the ground truth was obtained (ARI=1).

#Cross Tabulation for Hierarchical  
table(LmOB2\_data$Group, LmOB2\_hc$cluster)

##   
## 1 2 3 4 5 6 7  
## LG 0 0 0 0 0 11 1  
## RTE 29 30 1 1 1 0 0

#Compute the Cluster Stats  
LmOB2\_hcstats = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_hc$cluster)  
LmOB2\_hcstats

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 0.3175149  
##   
## $vi  
## [1] 0.8059036

#Use the same exposure variable generated above

From the cross-tabulation table it is apparent that the observations are well separated according to exposure, however, the observations associated with the RTE outbreak are spread out across 5 clusters (i.e., clusters 1-5) and the observations associated with the leafy green outbreak (LG) are spread out over two clusters (i.e., clusters 6-7). These results suggest that hierarchical clustering, although better than *K*-means and PAM clustering, is still not optimal (ARI = 0.3175).

#Re-evaluate with k=2  
  
#Cross Tabulation for Hierarchical  
table(LmOB2\_data$Group, LmOB2\_hc2$cluster)

##   
## 1 2  
## LG 0 12  
## RTE 62 0

#Compute the Cluster Stats  
LmOB2\_hcstats2 = cluster.stats(d = dist(LmOB2), LmOB2\_expnum, LmOB2\_hc2$cluster)  
LmOB2\_hcstats2

## $n  
## [1] 74  
##   
## $cluster.number  
## [1] 2  
##   
## $cluster.size  
## [1] 12 62  
##   
## $min.cluster.size  
## [1] 12  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 965.3818 1910.4298  
##   
## $average.distance  
## [1] 409.9878 818.0303  
##   
## $median.distance  
## [1] 261.9084 873.7557  
##   
## $separation  
## [1] 2251.811 2251.811  
##   
## $average.toother  
## [1] 2448.251 2448.251  
##   
## $separation.matrix  
## [,1] [,2]  
## [1,] 0.000 2251.811  
## [2,] 2251.811 0.000  
##   
## $ave.between.matrix  
## [,1] [,2]  
## [1,] 0.000 2448.251  
## [2,] 2448.251 0.000  
##   
## $average.between  
## [1] 2448.251  
##   
## $average.within  
## [1] 751.8612  
##   
## $n.between  
## [1] 744  
##   
## $n.within  
## [1] 1957  
##   
## $max.diameter  
## [1] 1910.43  
##   
## $min.separation  
## [1] 2251.811  
##   
## $within.cluster.ss  
## [1] 25380678  
##   
## $clus.avg.silwidths  
## 1 2   
## 0.8340084 0.6661622   
##   
## $avg.silwidth  
## [1] 0.6933805  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.9255062  
##   
## $dunn  
## [1] 1.178694  
##   
## $dunn2  
## [1] 2.992861  
##   
## $entropy  
## [1] 0.4432379  
##   
## $wb.ratio  
## [1] 0.3071014  
##   
## $ch  
## [1] 156.8357  
##   
## $cwidegap  
## [1] 846.1117 1569.4945  
##   
## $widestgap  
## [1] 1569.495  
##   
## $sindex  
## [1] 2255.8  
##   
## $corrected.rand  
## [1] 1  
##   
## $vi  
## [1] 0

#Use the same exposure variable generated above

Following re-evaluation of the previous analysis with k=2, the cross-tabulation and Adjusted Rand Index (ARI=1) suggest a perfect agreement between the two clustering.

*Sporadic*

To note: the R code below WILL NOT run for datasets that DO NOT contain clusters. This is due to the fact that the Adjusted Rand Index (ARI) is calculated based on the number of pairs of observations in the same cluster or in different clusters among partitions, and therefore, if no pairs are present, the index cannot be calculated.

#Cross Tabulation for kmeans  
  
table(LmSP\_data$Group, LmSP\_km$cluster)  
  
#Convert exposures to a numeric value  
LmSP\_expnum = as.numeric(LmSP\_data$Group)  
LmSP\_expnum  
  
#Compute the Cluster Stats  
LmSP\_kmstats = cluster.stats(d = dist(LmSP), LmSP\_expnum, LmSP\_km$cluster)  
LmSP\_kmstats

#Cross Tabulation for PAM  
table(LmSP\_data$Group, LmSP\_pam$clustering)  
  
#Convert exposures to a numeric value  
LmSP\_expnum = as.numeric(LmSP\_data$Group)  
LmSP\_expnum  
  
#Compute the Cluster Stats  
LmSP\_pamstats = cluster.stats(d = dist(LmSP), LmSP\_expnum, LmSP\_pam$clustering)  
LmSP\_pamstats

#Cross Tabulation for hierarchical  
table(LmSP\_data$Group, LmSP\_hc$cluster)  
  
#Convert exposures to a numeric value  
LmSP\_expnum = as.numeric(LmSP\_data$Group)  
LmSP\_expnum  
  
#Compute the Cluster Stats  
LmSP\_pamstats = cluster.stats(d = dist(LmSP), LmSP\_expnum, LmSP\_hc$cluster)  
LmSP\_pamstats

## Discussion

Overall, the optimal clustering solution for Outbreak 1 was determined to be hierarchical with k=2 using the *clValid* R package. From the analyses, it is apparent that *clValid* can be extremely useful, particularly if the dataset contains clusters, as the tool was able to identify the optimal number of clusters and clustering algorithm with a single line of code for the Outbreak 1 dataset. With respect to the Sporadic dataset, it appeared that k=2 was also considered to be “optimal”. However, as the number of clusters increased, the validation measures were also observed to improve. Therefore, had a larger range of clusters been investigated, *clValid* may have reported a more appropriate clustering solution for the Sporadic dataset.

Unlike *clValid*, the other R packages investigated in this study, including *fpc* and *factoextra*, required the user to test each combination of clustering algorithm and cluster number separately, and then manually assess all results to determine the “optimal” clustering algorithm and corresponding number of clusters. Such an approach is both time-consuming, and challenging.

Although a promising tool for cluster validation, *clValid* as it currently stands, is quite resource intensive. As stated previously, based on the computational demands of the stability validation measure (which the authors of *clValid* failed to mention), which can take several hours to run, the R code had to be executed as a batch command in a conda environment using a high computing cluster hosted at the National Microbiology Laboratory (i.e., Waffles). On average, the analyses took ~ 13 to 15 hours with 32 GB of RAM to assess three clustering algorithms over a range of 2 to 7 clusters.

Similarly, as reported in a recent study, the average analyses time to assess all currently available clustering algorithms in the *clValid* package (i.e., 10 algorithms in total) with a cluster size of k=2 was reported to take up to 92 hours using 32 GB of RAM [14]. Since the common laptop has a maximum of 16GB of RAM, running such an analyses would be extremely time-consuming, and therefore, would not be suitable for time-sensitive analyses unless access to a high computing cluster (HPC) is available. To encourage widespread use of the *clValid* package, it would be beneficial for the authors to incorporate some type of functionality that would enable the code to run more efficiently in order to increase the speed of the analysis and return of results.

Evaluating a combination of validation measures proved to be useful in the current study as conflicting results were often reported by either *clValid*, or the combination of *fpc* and *factoextra*. Therefore, using a variety of measures will help prevent incorrect assumptions about the “optimal” clustering solution, ultimately leading to more informed decisions about the most appropriate method for a given dataset. It is important to note that the validation measures investigated in the current study do not represent an exhaustive list, and therefore, other validation metrics may be equally useful.

Despite the overall utility of *clValid*, it is important to note that the optimal scores reported by the tool are not always in agreement with respect to the optimal clustering algorithm and number of clusters across all validation measures (as was seen in the current study). This finding highlights the complexity of selecting the “best” algorithm based on the optimal scores reported by *clValid*. Since these “optimal” scores can differ between validation metrics, it can become quite confusing as to which combination of clustering method and cluster number is the most suitable for a given dataset.

Although not evaluated as part of the current study, a recently developed R package, *optCluster* [14] is claimed to be able to use the output generated by *clValid* to return a SINGLE “optimal” clustering algorithm and cluster number based on the validation metrics that were investigated. This functionality would essentially overcome the tedious task of visually inspecting the results to determine the “optimal” clustering solution when a number of different combinations of clustering algorithms and cluster numbers are reported by *clValid*.

To conclude, cluster validation is an extremely important step in the cluster analysis workflow. Since clustering is a predictive method, there technically is no “correct” answer with respect to the best clustering solution (i.e., clustering algorithm and number of clusters) for a given dataset, however, cluster validation can provide valuable insight as to which algorithm would be the most appropriate for each unique situation. Although cluster validation can be more time-consuming than the actual act of clustering the data, it is a necessary process that must be adopted in order to obtain the most meaningful results from the data. Therefore, strong efforts should be made by researchers to implement a routine validation step in their cluster analysis workflow.

## References

[1] Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. Bioinformatics, 21(15), 3201–3212. <https://doi.org/10.1093/bioinformatics/bti517>

[2] Holmes, S., & Huber, W. (n.d.). Home | Modern Statistics for Modern Biology. Retrieved from <http://web.stanford.edu/class/bios221/book/>

[3] Brock, G., Pihur, V., Datta, S., & Datta, S. (2008). clValid: An R Package for Cluster Validation. Journal of Statistical Software, 25(1), 1–22. <https://doi.org/10.18637/jss.v025.i04>

[4] Hassani, M., & Seidl, T. (2017). Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. Vietnam Journal of Computer Science, 4(3), 171–183. <https://doi.org/10.1007/s40595-016-0086-9>

[5] R Core Team (2012). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org>

[6] Hennig, C. (2019). fpc: Flexible Procedures for Clustering (Version 2.2-3). Retrieved from <https://CRAN.R-project.org/package=fpc>

[7] Kassambara, A., & Mundt, F. (2017). factoextra: Extract and Visualize the Results of Multivariate Data Analyses (Version 1.0.5). Retrieved from <https://CRAN.R-project.org/package=factoextra>

[8] Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software, 61, 1–36. <https://doi.org/10.18637/jss.v061.i06>

[9] Nieweglowski, L. (2013). clv: Cluster Validation Techniques (Version 0.3-2.1). Retrieved from <https://CRAN.R-project.org/package=clv>

[10] Nadon, C., Van Walle, I., Gerner-Smidt, P., Campos, J., Chinen, I., Concepcion-Acevedo, J., Gilpin, B., Smith, A.M., Man Kam, K., Perez, E., et al. (2017). PulseNet International: Vision for the implementation of whole genome sequencing (WGS) for global food-borne disease surveillance. Euro Surveill. Bull. Eur. Sur Mal. Transm. Eur. Commun. Dis. Bull. 22.

[11] Cluster Validation Statistics: Must Know Methods. (n.d.). Retrieved November 26, 2019, from Datanovia website: <https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/>

[12] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. <https://doi.org/10.1016/0377-0427(87)90125-7>

[13] Dunn, J. C. (2008). Well-Separated Clusters and Optimal Fuzzy Partitions. Cybernetics and Systems, 4, 95–104. <https://doi.org/10.1080/01969727408546059>

[14] Sekula, M., Datta, S., & Datta, S. (2017). optCluster: An R Package for Determining the Optimal Clustering Algorithm. Bioinformation, 13(3), 101–103. <https://doi.org/10.6026/97320630013101>

[15] TagTeam: Bootstrap Evaluation of Clusters—R-bloggers—Statistics and Visualization. (n.d.). Retrieved November 28, 2019, from <http://tagteam.harvard.edu/hub_feeds/1981/feed_items/2126110>

[16] Datta, S., & Datta, S. (2006). Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC Bioinformatics, 7(1), 397. <https://doi.org/10.1186/1471-2105-7-397>

[17] Bioconductor—Home. (n.d.). Retrieved November 27, 2019, from <https://www.bioconductor.org/>

[18] Gentleman, R. (2019). annotate: Annotation for microarrays (Version 1.64.0). <https://doi.org/10.18129/B9.bioc.annotate>

[19] Gentleman, R., Carey, V., Morgan, M., & Falcon, S. (2019). Biobase: Biobase: Base functions for Bioconductor (Version 2.46.0). <https://doi.org/10.18129/B9.bioc.Biobase>

[20] Carlson M (2019). GO.db: A set of annotation maps describing the entire Gene Ontology. R package version 3.8.2. <http://bioconductor.org/packages/GO.db/>

[21] Hubert,A. (1985) Comparing partitions. J. Classif., 2, 193–198.

[22] Rand index. (2019). In Wikipedia. Retrieved from <https://en.wikipedia.org/w/index.php?title=Rand_index&oldid=909008884>