Performance Evaluation of Consensus Clustering Algorithms

STAT 598 Progress Report

Derek Chiu

August 25, 2015

Contents

1 Preface

This progress report fulfills the UBC Science Co-op requirement to submit a work term report at the end of every four month period. BC Cancer Agency (BCCA) is a not-for-profit organization that aims to provide care for cancer patients and conduct innovative cancer research. Our department, OvCaRe, is the Ovarian Cancer Research team tasked with studying ovarian cancers of many types. The objective of the project I am working on is to discover a viable classifer for ovarian high-grade serous carcinoma (HGSC). My role is to help devise a clustering algorithm that can statistically partition subtypes of HGSC without knowledge of the pathological properties of each sample. The progress report will evaluate the method we are using, consensus clustering, on a publicly available data set as well as simulated data sets. The final technical report will contain results of our method applied on HGSC data from our own cohort.

2 Introduction

Unsupervised learning is the process of inferring something about a data structure without knowing its true class labels. Cluster analysis is an unsupervised learning method of assigning entities into different groups based on one or more of their attributes. It is unsupervised because we do not know the true partitions of the entities. The objective is to place similar objects together in the same cluster and separate dissimilar objects into different clusters. For example, in genomics studies, we frequently try and cluster patient samples measured on a large number of molecular features. When we get a clustering assignment from an algorithm, we often want to evaluate its performance. Ideally, a good clustering algorithm is able to differentiate entities without knowledge of the true class labels. In addition, we want the algorithm to arrive at a stable and optimal number of clusters. The choice of the number of clusters is not trivial in some cases.

3 Methods

3.1 Clustering Algorithms

There are many clustering algorithms, each approaching the clustering problem in a different way. It is most important to note the advantages and limitations of each algorithm. These are some definitions of clustering performance:

- **Compactness**: how close together clustered objects are to each other
- **Separation**: how far apart objects in different clusters are
- **Connectivity**: how connected the objects are in the feature space

3.1.1 Hierarchical Clustering

This clustering algorithm is very popular because of its intuitive representation using dendrograms (trees). Based on a distance matrix, the objects/features are clustered based on a linkage type. More similar objects are joined near the bottom of a dendrogram whereas less similar objects are joined at a higher tree height. A

linkage criterion determines the distances amongst a set of objects/features using the pairwise distances. For example, an average linkage would use the average pairwise distances. In this way, a dendrogram with all objects/features can be made by recursively linking increasingly larger sets of observations together.

Single linkage works very well for data sets exhibiting connectivity but not compactness. An example of this would be tree rings. The clusters are circles, and objects that are far away can be in the same cluster compared to objects that are close. On the other hand, average linkage works well for data sets exhibiting compactness. This works where the clusters are look like non-overlapping blobs.

3.1.2 K-Means and PAM

First, k means are randomly initialized in the multi-dimensional object/feature space that we wish to cluster. Assigning each object/feature to its closest mean forms the clusters. The k means are re-calculated based on the centroids (center points) within each cluster. This process is iterated until the centroids converge.

There are two caveats to note when using k-means. First, the cluster assignments are unstable because they depend on the random initialization of the centroids. We preferably want to repeat the algorithm many times to see whether the clusters are sensitive to the choice of initial centroid. Secondly, choosing k is not an arbitrary. Cross-validation using an appropriate loss function is a popular method for choosing k. Other methods use evaluation indices, some of which we will describe later in the report.

Partitioning Around Medoids (PAM) is very similar to k-means except that we initialize k random data points as the medoids. This means after each iteration the new medoids will always be a real data points, whereas k-means allows the centroids to be anywhere in the feature space.

3.1.3 Nonnegative Matrix Factorization (NMF)

Given a non-negative data matrix A, we can factor it into two matrices W and H, which are also non negative. W and H have important properties. Suppose A has genes as rows and samples as columns. If we are interested in clustering samples, then H has a reduced gene space of metagenes that fully explain the samples. Samples are clustered based on the metagene they are most associated with. If we are interested in clustering genes, then W has a reduced sample space of metasamples that fully explain the genes. Genes are clustered based on the metasample they are most associated with.

In gene expression data, it is common to standardize the genes. Doing so would likely disrupt the nonnegativity of A required for NMF. A simple remedy can solve this problem: append the matrix $-A$ to the bottom of A (preserving the same number of columns), and set all negative entries to 0. The computational complexity has been doubled as a result. NMF takes a long time to run, but clustering assignments are highly stable.

3.2 Consensus Clustering

Consensus clustering is as much an algorithm itself as it is a way of combining realizations of other clustering algorithms. For this, we explain it in its own section. Algorithms like k-means and PAM are unstable, as the clustering assignments depend on the initialization of centroids and medoids, respectively. Consensus clustering combines results from repeated runs of a clustering algorithm. Typically, each run uses different random subsamples of the data to model sampling variability. A final clustering assignment is determined based on agreement across replicates.

The consensus matrix is a significant aspect of consensus clustering. Consider a consensus matrix C. Entry C_{ij} is the proportion of times that object i and object j were clustered together. The matrix is symmetric, and all entries range from 0 to 1. A perfect clustering would consist of a consensus matrix with only 0s (never clustered together) and 1s (always clustered together). The final step to obtaining the consensus clustering assignment is to perform hierarchical clustering on the consensus matrix.

Based on the hierarchical clustering, we can plot a heatmap of the consensus matrix. Repeating this for different values of k (number of clusters) and looking at the corresponding heatmaps, we can see which value of k provides the most stable clustering assignment. The goal is to have a well-defined diagonal block structure, one block per cluster. Using a performance measure such as PAC (which we will describe) would be a more formal method of assessment that doesn't rely on potentially subjective visualizations.

Consensus clustering attempts to stabilize results from randomness introduced by subsampling and the clustering algorithm. A natural extension is sometimes called meta consensus clustering, where we aggregate results across different algorithms. For instance, we may combine the consensus clustering assignments from 10 different algorithms to come up with a final clustering. The choice of which algorithms to use is not trivial.

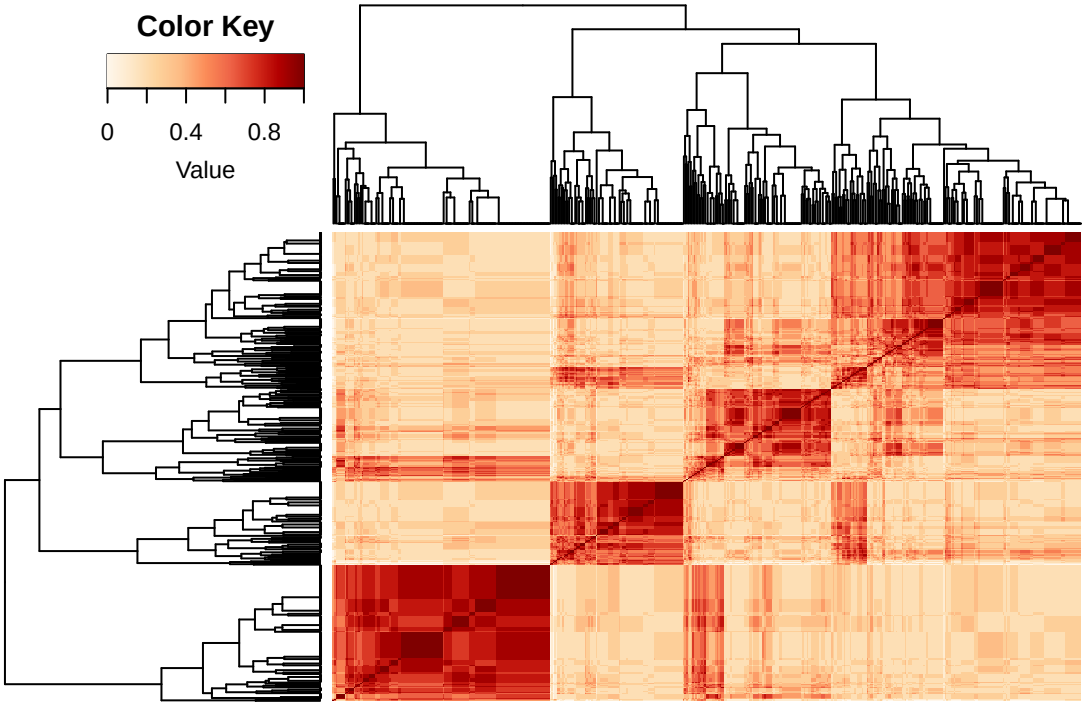
4 Performance Evaluation: TCGA Dataset

A published dataset from TCGA uses 321 genes to cluster 489 HGSC samples into the four subtypes: C1, C2, C4, and C5. They used consensus clustering with NMF. In this report, we consider the following clustering algorithms to use in consensus clustering:

- Hierarchical Clustering with Euclidean distance
  - Average Linkage
  - Single Linkage
  - DIvisive ANAlysis (DIANA)
- K-Means
  - Euclidean distance
  - Spearman distance
  - Mutual Information distance
- PAM
  - Euclidean distance
  - Spearman distance
  - Mutual Information distance
- NMF
  - KL divergence
  - Euclidean distance

The number of repetitions used for consensus clustering is 1000. Each replicate uses a subsample of the data with 80% of the total number of features.

The figure below shows the meta consensus matrix across the 11 algorithms, each of which used consensus clustering.



From the heatmap, we do not see a very high concordance across algorithms. For example, the proportion of cases with at least 0.6 agreement is only 0.2293322. There is some evidence of a four-class data structure.

The confusion matrix is shown below for the meta consensus classes compared to TCGA's classification. Several metrics are shown for each class.

	C1	C2	C4	C5
C1	106	23	23	8
C2	2	59	11	4
C4	0	25	91	22
C5	1	0	10	104

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Prevalence	Detection Rate	Detection Prevalence	Balanced Accuracy
Class: C1	0.9725	0.8579	0.6625	0.9909	0.2229	0.2168	0.3272	0.9152
Class: C2	0.5514	0.9555	0.7763	0.8838	0.2188	0.1207	0.1554	0.7534
Class: C4	0.6741	0.8672	0.6594	0.8746	0.2761	0.1861	0.2822	0.7707
Class: C5	0.7536	0.9687	0.9043	0.9091	0.2822	0.2127	0.2352	0.8611

4.1 External Evaluation

External evaluation refers to the case when we compare our clustering assignments to true class labels, or have some gold standard to compare to. In applications, this might be the published clustering result. The downside of using external evaluation is that the reference classes may not be correctly clustered themselves,

and we are treating these as the norm. None the less, we can explore a few metrics.

We expect that our own NMF-based algorithms to perform well on these evaluation indices because the reference classes were clustered using NMF too.

4.1.1 Purity and Entropy

Purity is defined as the sum of the entities of maximal class in each cluster divided by the total number of entities. The equation is:

$$Purity = \frac{1}{n} \sum_{r=1}^{k} \max_{i}(n_r^i)$$

where n is the total number of entities, k is the number of clusters, i is a particular class, and n_r^i is the number of objects classified into class i in cluster r. The larger the purity, the better the clustering accuracy.

Algorithms	Purity
NMF (Div)	0.7669
NMF (Euc)	0.7444
PAM (Spear)	0.7362
KM (Spear)	0.7157
PAM (Euc)	0.6687
HC (Diana)	0.6646
KM (Euc)	0.5971
KM (MI)	0.4376
PAM (MI)	0.4131
HC (Avg Euc)	0.2883
HC (Sing Euc)	0.2843

Entropy measures the amount of uncertainty in each cluster. The equation is:

$$Entropy = -\frac{1}{n \log q} \sum_{r=1}^{k} \sum_{i=1}^{q} n_r^i \log \frac{n_r^i}{n_r}$$

The smaller the entropy, the less uncertain we are of the cluster membership, and the better the clustering performance.

Algorithms	Entropy
NMF (Div)	0.5101
NMF (Euc)	0.5322
PAM (Spear)	0.5471
KM (Spear)	0.5703
PAM (Euc)	0.6012
HC (Diana)	0.6035
KM (Euc)	0.6634
PAM (MI)	0.8759
KM (MI)	0.9107
HC (Avg Euc)	0.9814
HC (Sing Euc)	0.9891

4.1.2 Kappa Statistics

Cohen's kappa statistic κ measures the level of agreement between two raters. In our case, the raters are different clustering algorithms.

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

The weighted κ is as follows:

$$\kappa_w = 1 - \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} x_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} m_{ij}}$$

Fleiss gave the following interpretations for κ, albeit arbitrary and not numerically determined.

Rating	Kappa
Poor	Less than 0.40
Fair	Between 0.40 and 0.75
Excellent	Greater than 0.75

The results for the TCGA dataset are shown below:

Algorithm	kappa	weighted.kappa	Rating
NMF (Euc)	0.658	0.7912	Excellent
NMF (Div)	0.6887	0.7894	Excellent
Meta	0.6477	0.7826	Excellent
KM (Spear)	0.6215	0.7606	Excellent
KM (Euc)	0.4625	0.741	Fair
PAM (Spear)	0.6501	0.7389	Fair
PAM (Euc)	0.5596	0.7282	Fair
HC (Diana)	0.5539	0.7236	Fair
KM (MI)	0.2521	0.2978	Poor
PAM (MI)	0.2247	0.2188	Poor
HC (Avg Euc)	0.0195	0.02409	Poor
HC (Sing Euc)	0.004911	0.01442	Poor

4.1.3 Adjusted Rand Index

The equation for the adjusted Rand index is shown below:

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

The larger the better.

Algorithms	ARI
NMF (Div)	0.4799
NMF (Euc)	0.4435

Algorithms	ARI
PAM (Spear)	0.427
KM (Spear)	0.4049
PAM (Euc)	0.3434
HC (Diana)	0.3415
KM (Euc)	0.2559
PAM (MI)	0.07465
KM (MI)	0.07369
HC (Sing Euc)	-0.0002933
HC (Avg Euc)	-0.0008894

4.2 Internal Evaluation

4.2.1 Proportion of Ambiguously Clustered Pairs (PAC)

The lower the better.

Algorithms	PAC
HC (Sing Euc)	0.05077
NMF (Euc)	0.1406
NMF (Div)	0.3267
KM (Spear)	0.425
KM (Euc)	0.4398
PAM (Spear)	0.4748
HC (Avg Euc)	0.5232
HC (Diana)	0.5558
PAM (Euc)	0.5984
KM (MI)	0.7167
PAM (MI)	0.9638

The PAC for the meta consensus matrix is 0.9810168.

4.2.2 Davies-Bouldin Index

For DBI, the lower the better.

Algorithms	DBI
HC (Sing Euc)	0.7637
HC (Avg Euc)	1.309
NMF (Euc)	1.702
NMF (Div)	1.702
HC (Diana)	1.719
PAM (Spear)	1.72
PAM (Euc)	1.725
KM (Spear)	1.725
KM (Euc)	1.741
PAM (MI)	1.935
KM (MI)	1.972

4.2.3 Dunn Index

For DI, the larger the better.

Algorithms	DI
HC (Sing Euc)	0.4364
HC (Avg Euc)	0.3696
KM (Spear)	0.3181
PAM (Spear)	0.3144
NMF (Euc)	0.2888
PAM (Euc)	0.286
HC (Diana)	0.2816
NMF (Div)	0.2766
KM (Euc)	0.2408
KM (MI)	0.2135
PAM (MI)	0.2135

4.2.4 Rousseeuw's Silhouette

For Rousseuw's Silhouette internal cluster quality index (RS), the larger the better.

Algorithms	RS
HC (Sing Euc)	0.1781
HC (Avg Euc)	0.1575
PAM (Euc)	0.1171
HC (Diana)	0.1124
NMF (Div)	0.1111
NMF (Euc)	0.1084
PAM (Spear)	0.1069
KM (Spear)	0.1059
KM (Euc)	0.08707
PAM (MI)	-0.0061
KM (MI)	-0.007504

4.2.5 C-Index

For CI, the lower the better.

Algorithms	CI
KM (Euc)	0.2816
NMF (Div)	0.3023
PAM (Euc)	0.31
HC (Diana)	0.321
NMF (Euc)	0.3369
PAM (MI)	0.3376
PAM (Spear)	0.3489
KM (MI)	0.3529
KM (Spear)	0.356
HC (Sing Euc)	0.4535
HC (Avg Euc)	0.4795

4.2.6 Gamma Index

For GI, the larger the better.

Algorithms	GI
PAM (Euc)	1.754
PAM (Spear)	1.715
KM (Euc)	1.676
NMF (Euc)	1.666
HC (Diana)	1.639
KM (Spear)	1.627
NMF (Div)	1.617
HC (Sing Euc)	0.7621
HC (Avg Euc)	0.7232
KM (MI)	-3.142
PAM (MI)	-3.358

4.2.7 CH Index

For CHI, the larger the better.

Algorithms	CHI
HC (Diana)	80.42
NMF (Div)	80.36
NMF (Euc)	79.26
KM (Spear)	78.71
PAM (Spear)	77.19
PAM (Euc)	75.71
KM (Euc)	75.38
PAM (MI)	13.23
KM (MI)	7.322
HC (Avg Euc)	5.452
HC (Sing Euc)	2.342

4.2.8 Connectivity

For connectivity, the smaller the better.

Algorithms	Conn
HC (Sing Euc)	8.93
HC (Avg Euc)	21.77
HC (Diana)	252.6
NMF (Euc)	260.8
NMF (Div)	263.8
KM (Spear)	270.7
PAM (Spear)	287.1
PAM (Euc)	297.3
KM (Euc)	329.1
KM (MI)	707.3
PAM (MI)	722.1

4.2.9 Summary

Here is a summary of all the indices for each algorithm, in unsorted order.

Algorithms	PAC	DBI	DI	RS	CI	GI	CHI	Conn
HC (Avg Euc)	0.5232	1.309	0.3696	0.1575	0.4795	0.7232	5.452	21.77
HC (Diana)	0.5558	1.719	0.2816	0.1124	0.321	1.639	80.42	252.6
HC (Sing Euc)	0.05077	0.7637	0.4364	0.1781	0.4535	0.7621	2.342	8.93
KM (Euc)	0.4398	1.741	0.2408	0.08707	0.2816	1.676	75.38	329.1
KM (MI)	0.7167	1.972	0.2135	-0.007504	0.3529	-3.142	7.322	707.3
KM (Spear)	0.425	1.725	0.3181	0.1059	0.356	1.627	78.71	270.7
NMF (Div)	0.3267	1.702	0.2766	0.1111	0.3023	1.617	80.36	263.8
NMF (Euc)	0.1406	1.702	0.2888	0.1084	0.3369	1.666	79.26	260.8
PAM (Euc)	0.5984	1.725	0.286	0.1171	0.31	1.754	75.71	297.3
PAM (MI)	0.9638	1.935	0.2135	-0.0061	0.3376	-3.358	13.23	722.1
PAM (Spear)	0.4748	1.72	0.3144	0.1069	0.3489	1.715	77.19	287.1

4.3 Ranked Indices

The table below shows the ranking of algorithms for performance on a clustering index, for each index. There is an additional column that shows the propoportion of indices where an algorithm was ranked **first or second**.

Algorithms	PAC	DBI	DI	RS	CI	GI	CHI	Conn	Top
HC (Sing Euc)	1	1	1	1	10	8	11	1	62.5%
HC (Avg Euc)	7	2	2	2	11	9	10	2	50%
NMF (Div)	3	4	8	5	2	7	2	5	25%
HC (Diana)	8	5	7	4	4	5	1	3	12.5%
KM (Euc)	5	9	9	9	1	3	7	9	12.5%
NMF (Euc)	2	3	5	6	5	4	3	4	12.5%
PAM (Euc)	9	7	6	3	3	1	6	8	12.5%
PAM (Spear)	6	6	4	7	7	2	5	7	12.5%
KM (MI)	10	11	10.5	11	8	10	9	10	0%
KM (Spear)	4	8	3	8	9	6	4	6	0%
PAM (MI)	11	10	10.5	10	6	11	8	11	0%

If we were to conduct a meta consensus clustering, a weight for each algorithm needs to be assigned. One such way of doing so is using the inverse rank sums. We can sum the ranks for each algorithm, and assign higher weight for consistently high ranked methods (1st or 2nd) and vice versa. This is shown below:

Algorithms	Top	Sum	Weight
NMF (Euc)	12.5%	32	12.54%
HC (Sing Euc)	62.5%	34	11.8%
NMF (Div)	25%	36	11.15%
HC (Diana)	12.5%	37	10.84%
PAM (Euc)	12.5%	43	9.33%
PAM (Spear)	12.5%	44	9.12%
HC (Avg Euc)	50%	45	8.92%
KM (Spear)	0%	48	8.36%

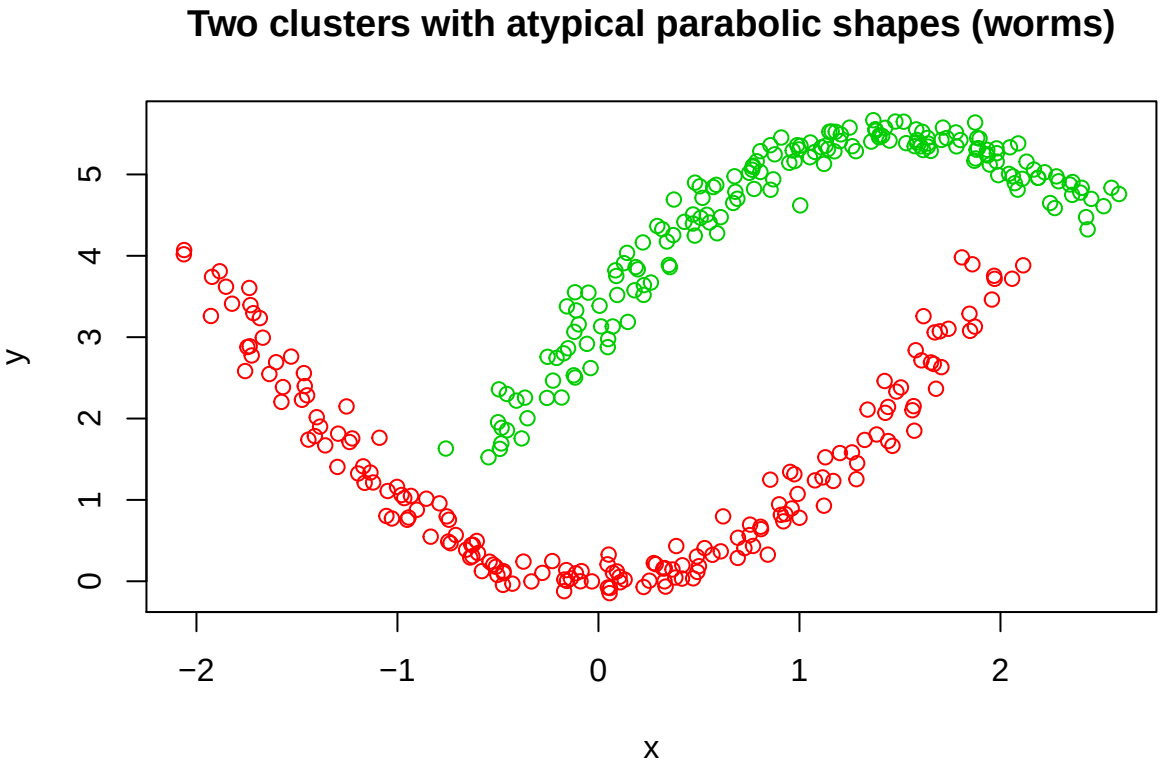
Algorithms	Top	Sum	Weight
KM (Euc)	12.5%	52	7.72%
PAM (MI)	0%	77.5	5.18%
KM (MI)	0%	79.5	5.05%

5 Simulations

To confirm the clustering results from using the TCGA dataset, we need to try out the algorithms on a few simulated datasets, designed to test the robustness of each method. The `clusterSim` package provides very good built-in examples to use.

5.1 Worms Dataset

The following two-cluster dataset is our first simulation:



The ranked indices and weights tables are shown below:

Algorithms	PAC	DBI	DI	RS	CI	GI	CHI	Conn	Top
KM (Euc)	8	1	4	1	1	1	1	5	62.5%
HC (Diana)	7	4	2	3	2	2	3	4	37.5%
NMF (Div)	6	2	10	2	3	3	2	11	37.5%
HC (Sing Euc)	3	8	1	7	7	7	7	1	25%
KM (Spear)	1	10	8	8	8	8	8	7	12.5%
PAM (Spear)	1	10	8	8	8	8	8	7	12.5%
PAM (MI)	4	7	6	10	11	10	11	2	12.5%

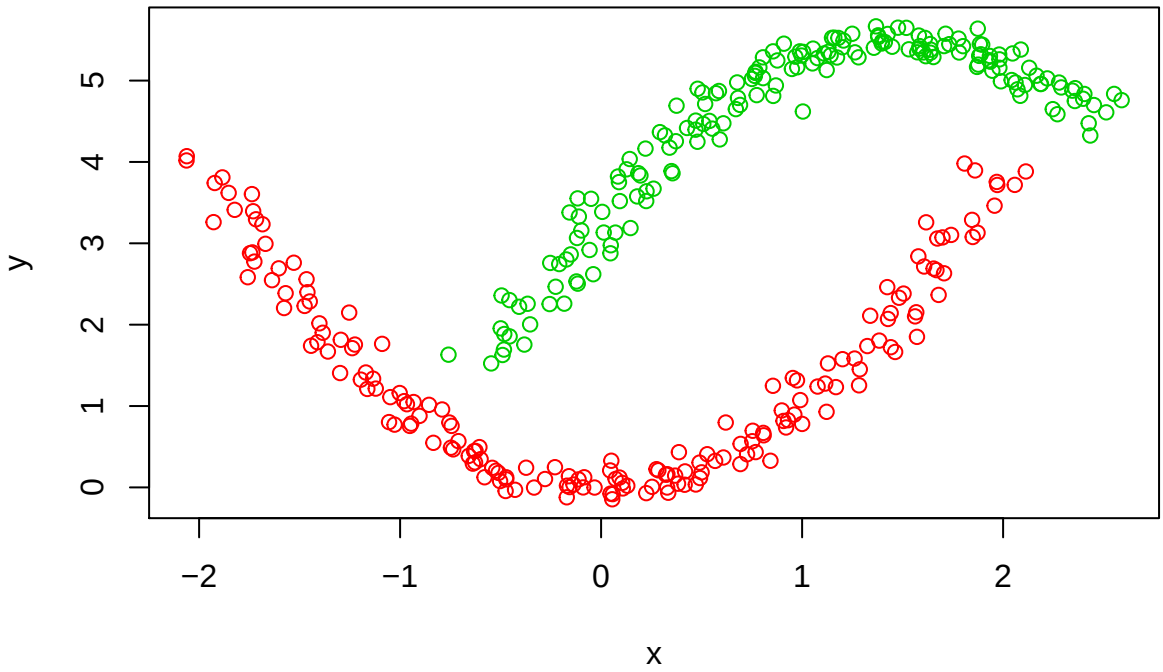
Algorithms	PAC	DBI	DI	RS	CI	GI	CHI	Conn	Top
HC (Avg Euc)	11	6	3	6	6	6	6	3	0%
KM (MI)	9	9	7	11	10	11	10	6	0%
PAM (Euc)	10	3	5	4	5	5	4	9	0%
NMF (Euc)	5	5	10	5	4	4	5	10	0%

Algorithms	Top	Sum	Weight
KM (Euc)	62.5%	22	17.39%
HC (Diana)	37.5%	27	14.17%
NMF (Div)	37.5%	39	9.81%
HC (Sing Euc)	25%	41	9.33%
PAM (Euc)	0%	45	8.5%
HC (Avg Euc)	0%	47	8.14%
NMF (Euc)	0%	48	7.97%
KM (Spear)	12.5%	58	6.6%
PAM (Spear)	12.5%	58	6.6%
PAM (MI)	12.5%	61	6.27%
KM (MI)	0%	73	5.24%

Using hierarchical clustering with Wald's method on the meta-consensus matrix across the ten algorithms, we can obtain meta-consensus classes. The following table shows how the different methods compare, based on the number of matches to the true class labels.

	Matches
HC (Sing Euc)	360
KM (Euc)	294
HC (Diana)	291
NMF (Div)	291
Meta	289
NMF (Euc)	287
PAM (Euc)	285
HC (Avg Euc)	277
KM (MI)	182
PAM (MI)	181
KM (Spear)	91
PAM (Spear)	91

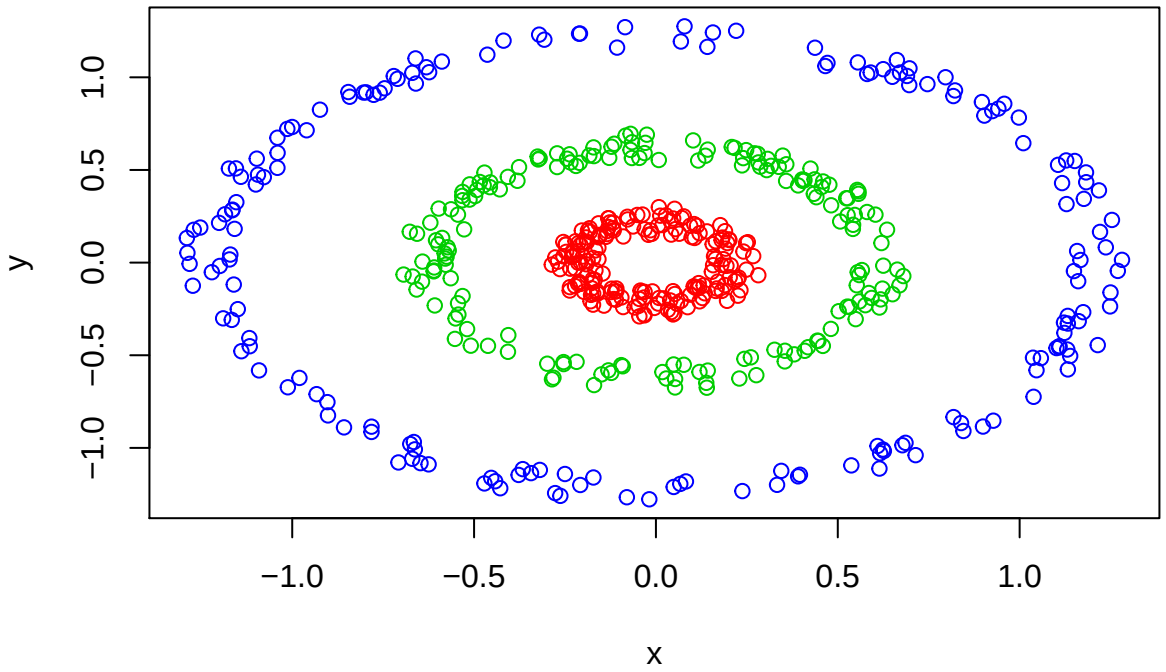
Let's visualize how the best algorithm, hierarchical clustering using single linkage and Euclidean distance, separates the clusters:



It appears that the best algorithm can separate the clusters, yet the indices do not favour such a partition and it appears that the algorithm is performing poorly.

5.2 Rings Dataset

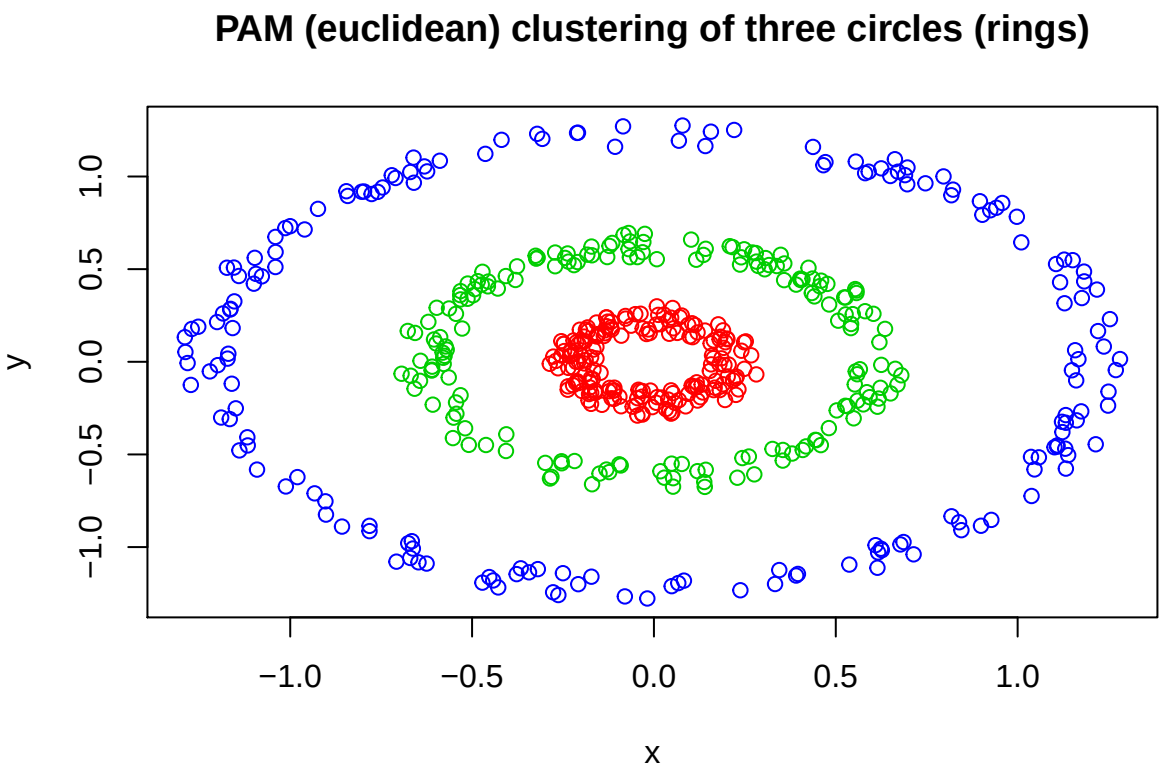
The following three-cluster dataset is our second simulation:



Algorithms	PAC	DBI	DI	RS	CI	GI	CHI	Conn	Top
KM (Euc)	7	7	2	1	1	1	6	2	62.5%
HC (Avg Euc)	6	1	3	2	8	9	5	3	25%
HC (Sing Euc)	3	8	1	9	9	10	9	1	25%
PAM (MI)	1	2	9	7	5	5	7	9	25%
NMF (Div)	2	4	8	6	7	8	2	6	25%
HC (Diana)	8	5	4	3	2	3	4	5	12.5%
KM (MI)	10	10	7	10	10	2	10	4	12.5%
NMF (Euc)	5	3	5	5	4	7	1	8	12.5%
KM (Spear)	4	9	9	8	6	6	8	10	0%
PAM (Euc)	9	6	6	4	3	4	3	7	0%

Algorithms	Top	Sum	Weight
KM (Euc)	62.5%	27	15.33%
HC (Diana)	12.5%	34	12.17%
HC (Avg Euc)	25%	37	11.19%
NMF (Euc)	12.5%	38	10.89%
PAM (Euc)	0%	42	9.85%
NMF (Div)	25%	43	9.62%
PAM (MI)	25%	45	9.2%
HC (Sing Euc)	25%	50	8.28%
KM (Spear)	0%	60	6.9%
KM (MI)	12.5%	63	6.57%

	Matches
HC (Sing Euc)	540
HC (Diana)	323
PAM (Euc)	288
KM (Euc)	270
Meta	240
HC (Avg Euc)	211
KM (Spear)	180
NMF (Euc)	180
KM (MI)	176
PAM (MI)	170
NMF (Div)	161



Again, we are only able to make linear separations.

6 References

Brunet, J. P., Tamayo, P., Golub, T. R., & Mesirov, J. P. (2004). "Metagenes and molecular pattern discovery using matrix factorization." *Proceedings of the national academy of sciences*, 101 (12), 4164-4169.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data." *Machine learning*, 52 (1-2), 91-118.

Cancer Genome Atlas Research Network. (2011). "Integrated genomic analyses of ovarian carcinoma." *Nature*, 474 (7353), 609-615.

Conrad, J. G., Al-Kofahi, K., Zhao, Y., & Karypis, G. (2005, June). "Effective document clustering for large heterogeneous law firm collections." In *Proceedings of the 10th international conference on Artificial intelligence and law* (pp. 177-187). ACM.

Fleiss, J.L. (1981). *Statistical methods for rates and proportions* (2nd ed.). New York: John Wiley.

Cohen, J. (1960). "A coefficient of agreement for nominal scales." *Educational and psychological measurement*, 20 (1), 37-46.

Cohen, J. (1968). "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit." *Psychological bulletin*, 70 (4), 213.

Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical Association*, 66 (336), 846-850.

Hubert, L., & Arabie, P. (1985). "Comparing partitions." *Journal of Classification*, 2 (1), 193-218.

Vinh, N. X., Epps, J., & Bailey, J. (2009, June). "Information theoretic measures for clusterings comparison: is a correction for chance necessary?." *In Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 1073-1080). ACM.

Senbabaoglu, Y., Michailidis, G., & Li, J. Z. (2014). "Critical limitations of consensus clustering in class discovery." *Scientific reports*, 4.

Davies, David L.; Bouldin, Donald W. (1979). "A Cluster Separation Measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-1 (2): 224–227.

Dunn, J. C. (1973). "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters." *Journal of Cybernetics* 3 (3): 32–57.

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.

Hubert, L. J., & Levin, J. R. (1976). "A general statistical framework for assessing categorical clustering in free recall." *Psychological bulletin*, 83 (6), 1072-1080.

Baker, F. B., & Hubert, L. J. (1975). "Measuring the power of hierarchical cluster analysis." *Journal of the American Statistical Association*, 70 (349), 31-38.

Calinski, T., & Harabasz, J. (1974). "A dendrite method for cluster analysis." *Communications in Statistics-theory and Methods*, 3 (1), 1-27.