Exploring the Efficacy of Popular Clustering Techniques on Gene Expression Data

S. TKS. Batista (730221)
Supervisors: Byron Jacobs & Chrissie Rey
Contributor: Warren Freeborough

November 18, 2020

School of Computer Science and Applied Mathematics

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Abstract

High throughput data has presented a wealth of genomic information, but as of yet a golden standard has not been presented and tested as means for the analysis of this data. Posing the question of whether biological function can be inferred solely from gene expression data of a host at different states. In-light of the lack of information that exits on the procedure to be employed in a true gene expression data exploratory process, a robust methodology was implemented. This included the use of a wide array of clustering algorithms along with numerous validation indices to attempt to discover the natural biological classes that existed within significantly unannotated data. While not being the most novel of the machine-learning techniques proposed for such data analysis, the k-means algorithm outperformed other methods when validated using known model validation techniques. The testing of the functional biological validity of these results were found to present a sufficiently accurate image of the underlying biological functions. These results while promising would require further validation via experimental methods to ensure the accuracy of the biological inferences.

Declaration

I, Savannah Trystan Kira Schlebusch Batista, (730221) am a student registered for the degree of MSc(Dissertation) in the academic year 2020

I hereby declare the following:

- I am aware that plagiarism (the use of someone else?s work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for the assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a evidence that this is not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I have included, as an appendix to this assignment, a report from ?Turnitin? software indicating the level of plagiarism in my research document, where stated as a requirement for the assignment submission.

Contents

1 Introduction

Cassava is a root vegetable, native to Brazil, that is consumed extensively in developing countries due to its high carbohydrate content and tolerance to drought. When other crops in Africa fail, namely maize, cassava is essential to more than 200 million African people's survival [1]. When considering these facts along with the global shortage of food[1], it becomes clear that making this plant virus-resistant would indeed be advantageous. This is potentially achievable if the molecular basis of host defence or susceptibility can be discovered [3].

The data used in this project was extracted by evaluating the genetic effect of geminivirus (South African cassava mosaic virus-SACMV) on the cassava plant. Although this same disease affects tomato and sweet potato crops in South Africa, currently only its affect on the cassava plant was addressed.
For the purpose of this project the characteristic of interest is the differential expression of a gene. In the context of gene expression analysis, the initial assumption that is made is that genes which are co-functional or co-expressed will display a similar gene expression profile [4]. If this assumption is correct then under cluster analysis these *similar* genes will be in the same cluster while genes that have different expression profiles and therefore genes that are not co-expressed or functionally related, will be placed in a different cluster. Successful clustering of gene expression data could provide vital information on genes whose functions are as of yet unknown. The core aim of the entire project being to discover if a mechanism for host immunity can be found by looking at differential expression in a susceptible and tolerant host. The clustering process is used to aid in this investigation.
However these clusters would need to be validated via experimental methods to ensure related biological function of the genes. This would verify that the clustering method is grouping by biological function or pathway and provide an experimental starting point for the investigation behind the mechanism of host immunity. A further insight that could potentially be derived is the functional annotation of as yet unannotated genes in the Cassava genome.
In this project differential gene expression data was clustered using Hierarchical clustering, the k-means method, Gaussian Mixture Model, Fuzzy Clustering, Spectral Clustering, DBSCAN Clustering, Affinity Propagation and Mean Shift Clustering. A consensus was sought between the results obtained from these methods as this presented a solution which was most representative of the methods. These results were used to generate *protein-protein-interaction* networks by using the *STRING* database which was further augmented with The Arabidopsis Information Resource(*TAIR*). The final solution presented is a possible pattern for the mechanism that confers immunity in the Cassava plant. This solution will need to be experimentally validated if it is to be considered a viable mechanism for cassava host immunity.

2 Background

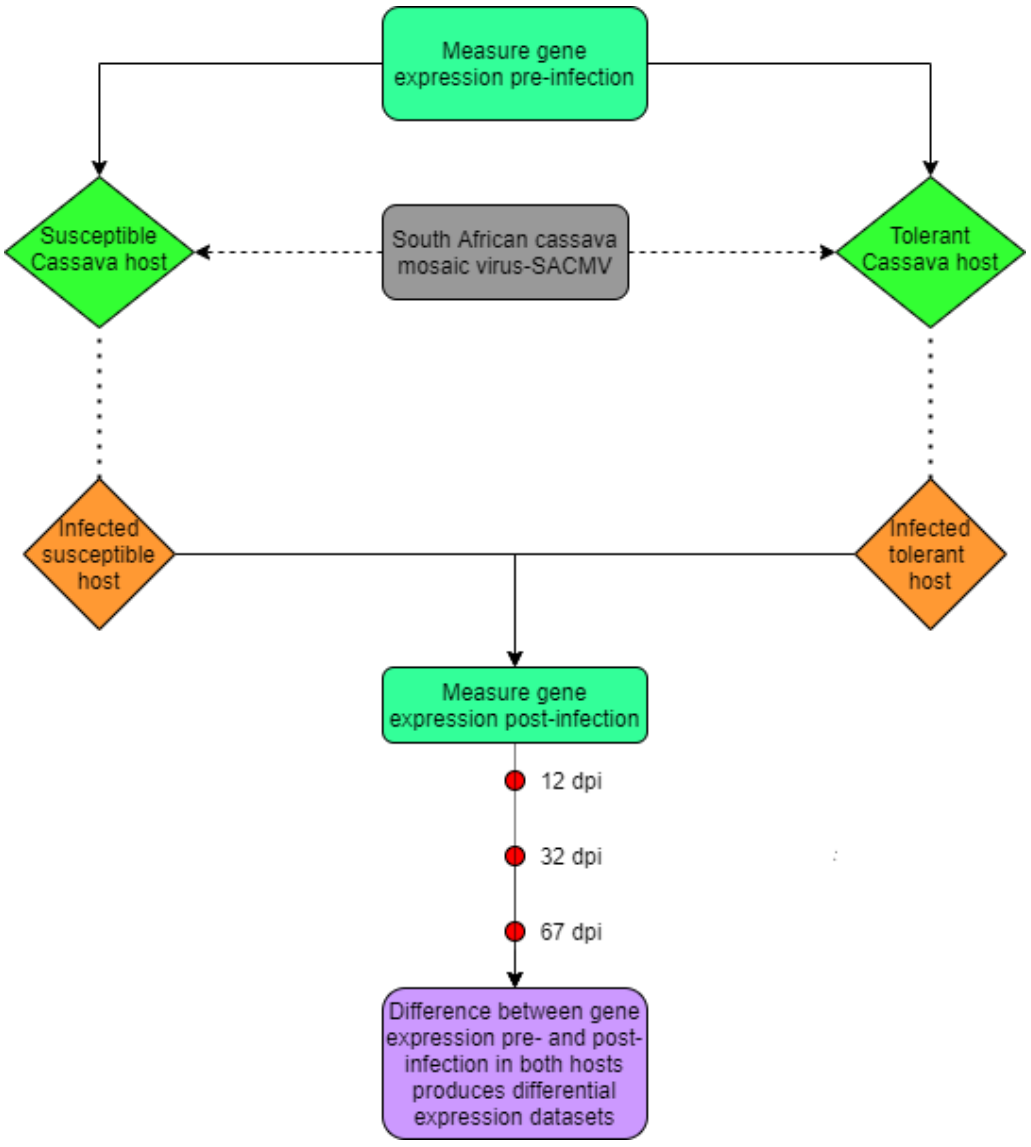
2.1 Biological Context

Gene expression profiling is done to measure activity of a cell's genome to produce DNA micro-array data. This is done to glean information regarding the function of the genes in the cell as well as the biological pathways/processes that these genes are involved in [5]. This falls in the sphere of Genomics - a relatively new field in science.

A DNA micro-array is a collection of DNA nucleotide sequences[2] on a solid surface, which are used to study gene expression of a large number of genes in a genome. Gene expression is measured by looking at the amount of expressed mRNA in a cell. This is the result of mRNA being directly responsible for gene expression. Prior to DNA micro-array technology, molecular biologists were concerned with understanding single genes. This single gene analysis is insufficient when looking at host immunity responses, which involve a multitude of genes [5]. Prior to the recent studies being done on the molecular basis for host immunity - as discussed by Lazzaro and Schneider in [6] - it was thought that host immunity was primarily determined by the immune system of the host. It is becoming increasingly more clear that the genetics of immunity involves multiple genes and various environmental factors, each contributing to produce host immunity to pathogens[6]. The consequence of this being that an evaluation of host defence or susceptibility to a pathogen would require insight into the genome-wide effects produced.

[1] *The Food and Agriculture Organisation of the United Nations reported that 820 million people had insufficient food in 2019[2].*
[2] Complimentary to the potential mRNA from the genes of interest.

Genetic profiling was done on two plants- the susceptible Cassava landrace (T200 dataset) and the tolerant Cassava landrace (TME3 dataset) at a pre and post-infection state, at three different time points. These time points were 12, 32 and 67 days-post-infection(dpi). Pre-infection profiling of the host plant provided a benchmark with which post-infection genetic profiling results could be compared against. The comparison between the gene expression levels of the pre and post-infection host provides the change in gene expression levels produced by the induced state change - referred to as differential expression[7]. The experimental procedure employed microarrays to determine the amount of bound mRNA to its DNA target as an evaluation of the differential expression of those genes. An increase or decrease in bound mRNA is indicative of respective up-regulation or down-regulation of a gene. A high-level overview of the process required to produce differential gene expression datasets is presented in Figure 1a.



(a)

Figure 1: A summarised view of the process required to produce differential gene expression data.

The six datasets produced from this profiling are considerable in size as they contain gene expression levels for

most if not all the genes in the plants genome, making interpretation of its results and information a sizeable task.

2.2 Cluster Analysis in Differential Gene Expression Analysis

Cluster analysis is an analytical tool which groups together objects that have similar characteristics. One of the first proposals of clustering as a means for quantitative gene expression analysis on a large-scale was made by G. Michaels *et al.* in 1998 [8]. These pioneers who implemented cluster analysis as an analytical tool for gene expression analysis on human genome data found that it was a viable tool in functional genomics as the clustering results produced correlated with shared biological pathways. The underlying assumption employed when using cluster analysis in gene expression analysis, is that genes with similar differential expressions may be co-regulated or may be involved in a common process which would explain the similarity in expression.
G. Michaels *et al.* utilized Euclidean distance and Mutual Information as a means to cluster/group their data. Since then clustering techniques have been used extensively within the broad context of functional genomics. In 2002 A. Gasch and M. Eisen implemented the fuzzy k-means algorithm on yeast gene expression data and found it to be a useful tool in the extraction of biological insights[9]. K. Yeung *et al.* sought to improve on the use of heuristic clustering techniques by employing the use of model-based clustering on synthetically derived and real gene expression data. They found that while model-based clustering produced superior results when applied to synthesized data, in the application of these methods on real data the results were not markedly different to those produced by heuristic algorithms[10]. G. Getz *et al.* proposed and further concluded in 2002 that Coupled two-way clustering was a suitable analysis method for gene expression data produced from sampling done on colon cancer and leukemia patients[11]. D. Huang and W. Pan in 2006 combined known gene functions with the k-medoids algorithm and hierarchical clustering on simulated data to find that this combination produced superior results when compared to results produced from solely the k-medoids or hierarchical clustering algorithms[12]. S. Wu *et al.* introduced the one-prototype-take-one-cluster (OPTOC) algorithm for the analysing of simulated and yeast cell cycle gene expression data and found that this particular method could too be considered a suitable method for analysis in functional genomics[13]. The results presented by K. Yeung *et al.* provide a relevant example of one of the hurdles that arise when seeking out an optimal clustering algorithm to analyse the distinct gene expression data at hand [9]. The quality of the *partitions* produced by each clustering algorithm are bound to the nature of the underlying data. This is to say that an individual clustering algorithm may be superior when applied to one dataset that has a particular underlying distribution, but this same algorithm will fall short when the underlying distribution is not distinctly matched to the particular way in which the algorithm seeks to partition the given dataset. This wealth of literature based information, while providing a substantial basis for the use of clustering algorithms for gene expression data analysis, does not point to a single method as being the *golden standard*. It is therefore proposed that a multitude of clustering algorithms be tested in order to produce a clustering consensus, within which the *actual* partitions of the dataset may be discovered.

There are no less than a dozen clustering methods that are available, each with its own mathematical method of dividing the data that has been presented. The first category of clustering methods that needs be discussed is non-parametric clustering techniques. These methods are used in a wide array of fields some of which include statistical data analysis, machine learning, image analysis, pattern recognition, bioinformatics and computer graphics. This group of methods is called non-parametric as the the number of clusters that governs the data is not known beforehand and the methods do not require this information as an input to analyse the given data.

On the other end of the spectrum are parametric clustering techniques. Here the algorithm is told just how many clusters there are in the data before the process has begun.

While parametric clustering implies that more information on the given dataset is at hand, both of these techniques are used for exploratory data analysis. They are but the first step when it comes to data analysis. They can only supply certain insights into the nature of the underlying data. The results obtained from cluster analysis need to be evaluated within the context of the supplied data to gain relevant insight into the data.

Cluster analysis can be considered quite subjective in nature. While there has been a fair amount of work done to improve on the accuracy of the methods that will be presented, there is not one method that is superior for each data set that is provided. The method that is selected for the data depends on the underlying distribution.

2.2.1 Hierarchical Clustering

The first group of methods that will be used are hierarchical clustering techniques. Within this category, there are divisive and agglomerative methods. In divisive clustering, the process starts with one exhaustive cluster. This one large cluster is then iteratively divided up into smaller clusters using a distance measure. These distances are computed at each step of the process and are placed into a similarity matrix. The algorithm stops when each point is in its own cluster. This is clearly not the desired result as it would simply produce a partition for each gene [3]. A dendrogram is therefore used to determine which is the optimal number of clusters.

Agglomerative Hierarchical clustering starts with each data point being in its own cluster. Proceeding iteratively, clusters are then merged according to the distance and linkage criterion decided upon. With this method, a dendrogram is also used to determine the optimal number of clusters.

Hierarchical clustering is used quite extensively in the biological sphere because this method works best with data that has a natural hierarchy. It is also an unsupervised method and therefore does not require the number of clusters to be known beforehand. This method produces a good visual representation of the data which includes the entire hierarchy of the dataset.

While hierarchical clustering can be a powerful method when used in conjunction with data that has a natural hierarchy, it has a couple of significant drawbacks. These being:

1. Only one pass is made over the data when producing the final clusters. This means that an incorrect assignment made in the beginning of the process, cannot be undone or optimised.
2. There is not an objective function that is directly minimised or maximised by the method.
3. The methods can be quite sensitive to outliers and noise.
4. Clusters of different shapes and sizes are not always dealt with correctly. Large clusters may be split up unnecessarily.
5. There is no automatic means of discovering the optimal number of clusters for each dataset.

It is for these aforementioned reasons that other methods shall be proposed alongside hierarchical clustering. This method may behave in a robust manner in biological study, but it does lack a certain flexibility that is required with gene expression analysis.

2.2.2 k-means Algorithm

The next method that is proposed for this data is the k-means method. This method is categorically referred to as a partitioning method as it attempts to partition the data into k (the number of clusters that governs the data) parts. This value of k needs to be provided beforehand and therefore is not considered an non-parametric method. There are numerous statistical tests that are available to help determine this optimal value of k.

In k-means clustering k points are randomly selected as the centre of k clusters. Data points that are closest to this point are then assigned to that cluster. The mean of each cluster is calculated and is assigned as the new centre of the cluster. The process continues to iterate until the cluster centres stop moving.

This method has a simple yet powerful methodology. It seeks out points that are closest and assigns these points to the same cluster. Points that are not near each other are then assigned to a different cluster. The result of this clustering is then a sufficiently partitioned dataset. The k- means method offers a sufficient jumping-off point in the data exploration process but it does not necessarily offer the most optimal clustering assignments. This is due to the inherent restriction that is placed on the size and shape of the resulting clusters. k- means assumes that each cluster in a dataset is circular, of the same size(in terms of cluster diameter), and of the same density. While this assumption may be viable for test data with an even distribution, it is not necessarily applicable to *real-world* data.

This result is easy to conceptualise when considering the implications of having a dataset with evenly sized and shaped clusters. The implication being that the classes or categories that exist in the data are of the same size and

[3] Implying that only one gene relates to one function.

are distributed in the same way. By making this assumption, one substantially limits the integrity of the results that are obtained as natural groups that may exist in the data are forced into clusters that may not be suitable to the actual spread of the underlying data.

A second important feature of k - means clustering which needs be addressed is the fact that it makes hard assignments of the data points[4]. There is no probability that is offered along with k - means clustering results which could lead to a more comprehensive and contextual interpretation. It is due to these drawbacks that the Gaussian Mixture Model was proposed by K. Yeung *et al.* in [10].

2.2.3 The Gaussian Mixture Model

The Gaussian Mixture Model is a probabilistic clustering technique that aims to address the rigidity that is introduced by traditional k - means clustering. In a probabilistic model a data point does not have to solely belong to just one cluster[5]. Instead, there is a probability associated with each point that indicates *how much* a point belongs to a cluster or multiple clusters. Furthermore, Gaussian Mixture Models allow for clusters of varying shapes and densities. This is done by estimating not just the mean parameter but also the variance parameter. The underlying logic being that the data being analysed is produced by k Gaussian's. The algorithm then attempts to learn the parameters of these Gaussian's, namely the mean and variance[6]. The tangible strength of the Gaussian Mixture Model is that it is able to identify clusters that are perhaps not spherical in nature. This feature becomes especially useful if there is no a priori knowledge of the distribution and features of the data at hand.

While the Gaussian Mixture Model presents a clustering option that can not necessarily be provided by the k - means method alone, it is not the only algorithm that seeks to make soft or *fuzzy* assignments of the data points. The Fuzzy C-means algorithm is one such alternative soft assignment method.

2.2.4 Fuzzy Clustering

This algorithm was first introduced in 1965 by J. Zadeh and then it was later modified in 1981 by J.C. Bezdek [14]. This algorithm uses the notion of membership as a means for allowing data points to belong to more than one cluster. Any point can belong to more than one cluster with a membership value between 0 and 1, with the constraint that all these membership values need to sum to one [15]. In fuzzy c-means clustering the initialisation begins with k centres of the clusters being defined, each point is then assigned to a cluster using Euclidean distance as a measure. Membership values are calculated for each point which defines the fuzziness of the clustering. The algorithm iteratively repeats until the centres cease moving which is then identified as the optimal solution [14]. Like k-means clustering and the Gaussian Mixture Model, fuzzy clustering is parametric as a value for the number of clusters needs to be supplied at the onset of the calculations. The last of the parametric clustering techniques to be implemented is the Spectral Clustering algorithm.

2.2.5 Spectral Clustering

While components of the spectral clustering algorithm can be found in literature as far back as 1969 [16], it wasn't considered among the popular machine learning methods until the work done by S. Jianbo & M. Jitendra in 2000 and A. Ng & M. Jordan in 2002 [17, 18]. This class of techniques utilises the eigen-structure underpinning a similarity matrix of the data to create partitions in the dataset. These partitions create separate clusters in which high similarity points are grouped together while points with low similarity are placed in different clusters [19]. Unlike the k-means algorithm, spectral clustering does not make hard assumptions on the structure of the underlying cluster, implying that it is able to detect clusters of an irregular shape [20]. This may prove to be especially useful if the natural classes of the data are not spherical or ellipsoidal. This method may have been seen to outperform other clustering algorithms - like the k-means or fuzzy clustering algorithms - when tested on suitable data, but it still requires knowledge on the number of clusters present in the data beforehand. Therefore non-parametric techniques will be tested alongside the parametric methods mentioned. One of the non-parametric methods that will be tested is the Density-Based Spatial Clustering of Applications with Noise as developed by M. Ester *et al.*

[4] Hard clustering entails assigning each data point to only one cluster.
[5] This feature of a point being able to belong to more than one cluster is known as a soft assignment.
[6] Here the variance refers to the width of the bell shape curve that is produced by a Gaussian distribution.

in 1996 [21]. This method could provide insight into the natural number of clusters or classes that exist within the data, as a value for the number of clusters is not supplied rather it is sought out by the algorithm.

2.2.6 DBSCAN Clustering

In DBSCAN clustering the density that is assigned to each point is calculated by looking at the count of points that lies within a pre-defined region of this point. If this density is above a certain value then this collection of points are defined to be a cluster [22]. This clustering technique presents as an attractive analytical tool as not only does it not require a priori knowledge of the number of clusters that govern the data, but it is able to form clusters that are non-spherical. The latter of these qualities result from the methodology that is employed in defining and forming the clusters. One drawback that may become evident with this method is that if there are clusters that have similar densities then the algorithm may not be capable of distinguishing between these classes [22]. Therefore two other non-parametric clustering techniques will be utilised. The first of which being the Affinity Propagation clustering algorithm.

2.2.7 Affinity Propagation Clustering

Affinity Propagation was first presented as a means for pattern recognition of a dataset by B. Frey and D. Dueck in 2007 [23]. The method developed by these authors was tested on a range of data-based problems which included microarray data, cities whose access points include travel via airlines, images of faces and the lines of a manuscript. By using the similarity of pairs of data points as an input, which then was exchanged among these points as a message to seek out exemplars (points that are best suited to represent their cluster) and clusters. They found that they were able to produce clusters that contained less errors than other clustering methods that were tested and these results were produced in a fraction of the time [23]. The last non-parametric technique that will be evaluated is the Mean Shift Algorithm.

2.2.8 Mean Shift Clustering

This technique sought to cluster points by relying on the concept of kernel density estimation and was introduced by K. Fukunaga and L. Hostetter in 1975 [24]. Using an iterative approach, the authors aim was to seek out the modes of the probability distribution of the data which was achieved by maximising the kernel density estimation[7]. This method is non-parametric and therefore is able to identify clusters of irregular shapes [25]. It does however come with a high computational cost as the number of points being clustered increases.

2.2.9 Parameter Selection

The DBSCAN, Affinity Propagation and Mean Shift clustering algorithms offer the advantage of not requiring the number of clusters that governs the data to be supplied as the input. The number of clusters that are found by each algorithm for the data could be used as the input for the parametric clustering techniques. The downfall however of this methodology is that it would require the clustering result of the non-parametric methods to be the optimal partitioning for the dataset which would then produce this value for k. Moreover, there is no guarantee that each of the three methods would agree on a value for the optimal number of clusters. If they were to disagree on this number then it would be unclear on which cluster number to use. It is for this reason that formalised model/parameter[8] selection tests will be used.

In the statistical tests that will be used, each clustering result that is paired with a value for k (the number of clusters that have been used to perform the clustering) is considered to be a possible model. These tests then look at a specific feature that marks a cluster as well-fitted which is then used as a comparison metric to determine

[7] The kernel is a weighting function, the most popular of which is the Gaussian kernel [25].

[8] The number of clusters to be used for a clustering algorithm is technically a parameter which would lend itself to the process being considered parameter selection. The process of discovering this value for k involves comparing clustering results produced by a partitioning process which therefore would allow the interchangeability of the terms parameter and model selection.

which model(clustering solution) is more optimal/favourable. The first of these tests is *sum of squared estimate of errors (SSE)*.

The SSE measures the squared difference between a point and the mean of the cluster that it belongs to. This test therefore can be used to measure the amount of variation that exists within a cluster(intra-cluster variation) [26]. A preferable model will be one that minimises the total SSE as it would imply a solution that has minimal intra-cluster variation. This intra-cluster variation can also be referred to as cohesion and is the feature that is examined in Silhouette Analysis.

P. Rousseeuw presented Silhouette analysis as the process which examines the distance between a point and the clusters that surround it [27]. Collectively these values make up the silhouette coefficient of the model, with values closer to one indicating more cohesion while a value closer to negative one would indicate low cohesion within the model. The third metric that uses intra-cluster variation as a basis for model selection is Hartigan's Rule.

In 1975 J. Hartigan presented an equation that uses the intra-cluster variation for a model and the number of clusters that have been implemented in the model [28]. With these inputs the equation computes whether another cluster should be added to the partitioning to reduce the current intra-cluster variation. The Calinski-Harabasz Index too uses this definition of intra-cluster variation in its testing of model validity, but with the added information that is gathered from calculating the inter-cluster variation.

While intra-cluster variation examines the variation that exists within the points in a cluster, inter-cluster variation measures the dissimilarity of clusters. A model producing well-separated clusters would have high inter-cluster variation, while one that produces well-formed and dense clusters would be a model that has low intra-cluster variation. The Calinski-Harabasz index examines both of these factors in its model selection [29]. The Davies-Bouldin index is a similar index as it seeks to use the distinctness of clusters to measure the fit of a model.

As explained by L. Yanchi *et al.*, the Davies-Bouldin performs model selection by testing whether the proposed model produces clusters that are not only dissimilar but also well-separated [30]. A concept which was standardised by R. Tibshirani in 2001 with the Gap Statistic [31].

The calculation of the Gap Statistic utilises a normalised equation for the intra-cluster sum of squares which therefore provides a measure of cluster compactness. Furthermore a null distribution (a distribution in which there is no obvious natural partitioning) is used for comparison which aids in the evaluation of the model at hand. This Gap Statistic employs normalisation and standardisation in such a way that it is insensitive to distance measures along with clustering methods. Intra- and inter-cluster variation both provide valuable insight into the quality of a model but they are not the only means available for model selection. The last two tests that will be used for model selection are the Akaike Information Criterion(AIC) and Bayesian Information Criterion(BIC) [26].

Both AIC and BIC indices utilise the maximum likelihood of the model to quantify its quality. This is built on the notions presented in information theory. In the context of clustering, information theory dictates that *information* is lost when a model is fitted to *real-world* data. It follows that the maximum likelihood of a model is the *chance* of producing the actual data that was used, if the only information that is present is the model itself [32]. The AIC and BIC indices therefore rank models by their maximum likelihood, with models having a high maximum likelihood being one's which describe the underlying data best.

While there are similarities between the above mentioned tests in regards to the way in which they test the quality of a model, the results of each test will not be implemented in isolation. Instead a consensus will be sought out to determine the optimal number of cluster to be used in each dataset. This methodology will account for the lack of a priori knowledge of the number of classes that govern the data, but it cannot provide the *true* labellings of the data.

2.2.10 Comparing Clustering Results

When testing the performance of a clustering algorithm, it is standard practice to compare the *predicted* assignments as produced by the method with the *true* assignments that are available for the data. In the context of this project, there is not substantial information regarding the functions of the genes that are being tested, the implication being that there is no *true* assignment information for the provided data. Therefore the performance of the algorithms cannot be objectively tested for the supplied data, which would indicate the method that is the best fit for this data. Instead a consensus in data assignments - as was proposed for model/parameter selection - will be sought out between the clustering methods [33]. If there is a significant amount of agreement between the assignments made

by the extensive methods to be tested, then it supplies a degree of confidence to the functional assignments that will be presented in this project.

This consensus/agreement between the assignments made by each method will be quantified by the use of indices which are conventionally used to measure the level of agreement between *true* labelling assignments and *predicted* labelling assignments(partitions produced by a clustering algorithm). To utilise these indices the results from each method will iteratively be determined as the *true* assignment for the data. The remaining methods will then be measured against this temporary *true* assignment which will provide the level of agreement between the clustering results.

The indices/scores that will be used to measure the consensus between clustering results include the Adjusted Rand Index, The Mutual Information Score, Homogeneity and Completeness Scores, the V-measure and the Fowlkes-Mallows Score. Each index/score - like the tests used in model/parameter selection - has a distinct way in which it tests the agreement of two different assignments.

The Adjusted-Rand Index, Mutual Information Score and the Fowlkes-Mallows Score - while employing different equations in their similarity measure - essentially quantify cluster agreement by testing if the *true* assignments are suitably similar to the *predicted* assignments [33]. This is simply a process of examining whether the same points exist in the same cluster in both the *true* and *predicted* assignments.

The Homogeneity Score requires that *predicted* labelling displays an adequate level of homogeneity to be considered a suitable model [34]. This concept of homogeneity being that clusters(the partitions produced by the clustering algorithm) only contain points that belong to the same class(groupings that are defined by the *true* assignments). The Completeness Score is measured alongside the Homogeneity Score, but instead looks to the completeness of the predicted clusters as a means for measuring similarity [34].

A cluster is considered to be complete if all the points of a class(groupings provided by the *true* assignments) are placed into the same cluster by the algorithm being tested. The V-Measure is a score that combines these notions of completeness and homogeneity.

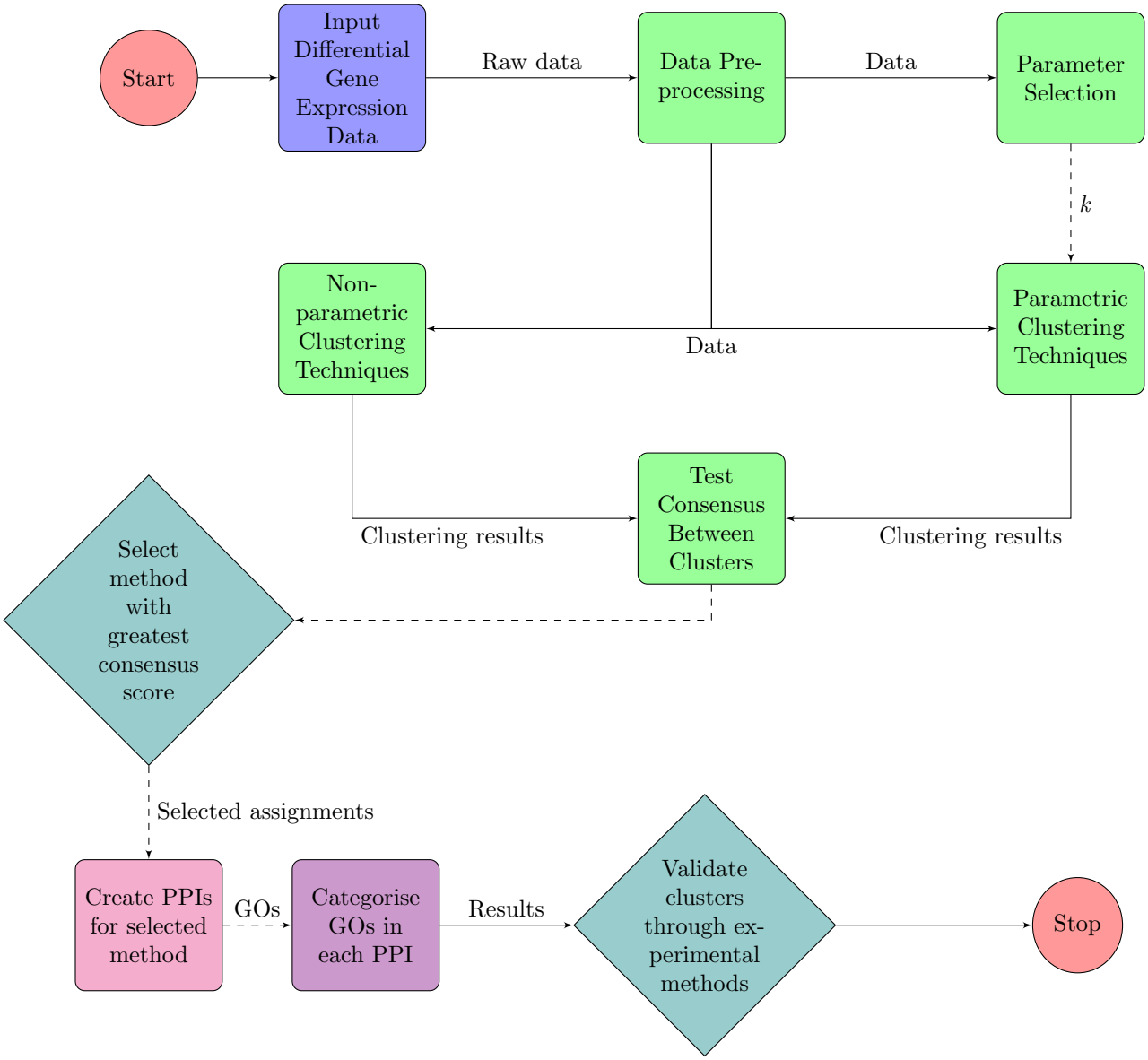
Instead of defining similarity based on *predicted* assignments being solely homogenous or complete, the V-Measure combines these qualities to define similarity as a metric based on *predicted* assignments being **equally** homogenous and complete. The implementation of these similarity metrics will provide valuable insight into the consensus that exists between the proposed clustering algorithms. While a thorough evaluation of the biological implications of each clustering result may prove insightful in the context of method selection for the gene expression data that has been provided, this analysis would require substantial time and computational expenditure. To optimise the time and resources available, the method that displays the greatest level of consensus with the remaining clustering methods will be selected for biological validity testing. The first step of this being to generate a Protein-Protein Interaction map (PPI) for each of the clusters.

2.2.11 Interpretation of Clustering Results

The PPI maps that will be generated using the STRING database for the relevant clusters will provide information of the related protein interactions that may exist within the cluster. If the PPI networks show interactions between the proteins that are encoded by the genes in the cluster, then it could indicate that the clustered genes are connected in terms of biological function. While this would initially validate the use of the tested clustering algorithm, the addition of categorising the proteins in the relevant PPI into their gene ontologies would further provide additional validation. This would provide insight into the dominant biological pathway or function that may be attributed to the clustered genes used to produce the PPI. If it is found that there is a gene ontology that dominates the PPI as produced by each cluster - when compared to the PPI for the whole genome of the plant - this would provide further evidence for the use of the algorithm in clustering genes that are connected by biological function.

3 Methodology

Figure 2: The process to be used to cluster the supplied data is depicted below. Green blocks describe methodology that will be implemented in *Python 3.74* with the aid of the *scikit-learn* package[35]. Pink blocks represent the portions of the process that will implemented in *STRING*[36], while the purple block is the process that to be implemented in *TAIR*[37].



The flow-chart in Figure 2 above gives a broad overview of the process that the data will undergo. The green blocks represent the clustering process itself together with parameter selection. The flow-chart past this point encompasses the biological validation steps which are employed to answer the question of host-immunity in the Cassava plant by looking at differential expression in a susceptible and tolerant host.
The methods derived here utilised data which was given in the following format:

Table 1: A Sample of the raw data supplied for the Cassava genome

Genes	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
MANES_12G133700	37.80779	3.941747205	1.4947	2.63723	0.0084	1
MANES_04G076500	465.0293	-3.06933414	1.1723	-2.6182	0.0088	1
MANES_12G058200	23.29623	4.572961266	1.7521	2.61005	0.0091	1
MANES_07G061600	31.15177	-4.037686016	1.557	-2.5933	0.0095	1
MANES_12G100800	431.8872	3.031648978	1.1702	2.5906	0.0096	1

The values that are of interest are in the *log2FoldChange* column. These provide the *change* in gene expression for the matching genes in the fist column. To clarify, if the gene - MANES_12G133700 is considered. A *log2FoldChange* value of 3.94174720 means that this particular gene was up-regulated in the observed state[9]. Meaning that there was an increase in the expression of this gene in the state under which it was tested.

These *changes* in gene expression were given for the genes in both a susceptible[10] *Cassava* landrace host (T200) as well as a tolerant[11] *Cassava* landrace host (TME3) at 3 different time-steps namely- 12dpi, 32dpi and 67 dpi[12]. It is this data which will be clustered to examine whether biological function and host immunity can be inferred from the produced clusters, but only once the data has undergone cleaning and preprocessing.

3.1 Data Cleaning and Preprocessing

The data that was provided for both the susceptible and tolerant *Cassava* landrace plant first needed to be cleaned and processed before it could be used as inputs for the clustering algorithms. Firstly only differentially expressed genes were included in this analysis. A gene is differentially expressed if the read count of the gene differs significantly between two different experimental conditions [7]. For the data in this project these are before and after infection. A gene whose expression was not sufficiently altered by the introduction of an infection is not considered differentially expressed and is not considered when examining host-immunity as its state was non-phased by the infection and therefore it did not contribute to host defence or immunity.
Furthermore genes with differential expression values falling within the following range were excluded[13] as they are not considered biologically significant[14]:

$$-1.5 < log2FoldChange < 1.5. \tag{3.1.1}$$

This value of 1.5 *log2FoldChange* in differential gene expression provides a sufficient level of confidence that the genes under analysis are in fact differentially expressed. This is a level of confidence that is used extensively in Biological fields and is done to add a level of confidence when one refers to a gene as differentially expressed. For example a gene who had a differential expression of say 1.2 could more likely be attributed to environmental factors or inconsistencies in data capturing and calculating, than an actual existence of differential expression.
Remaining points were then plotted with the susceptible hosts' change in gene expression level (x-axis) against the tolerant hosts (y-axis). Relating the data in this way involved matching up the genes that were differentially expressed in both the susceptible and tolerant host by using the gene IDs (the values in column 1 of Table 1). By clustering the data that has been plotted this way, insight may be gleaned into how a group of genes is expressed in the tolerant and susceptible host. The difference or similarity of which could provide information on the basis for host immunity or susceptibility.

If a gene was differentially expressed in the tolerant host but not the susceptible host (or *vise-versa*) then it will be excluded from the analysis (a natural consequence of matching genes from a susceptible and tolerant host

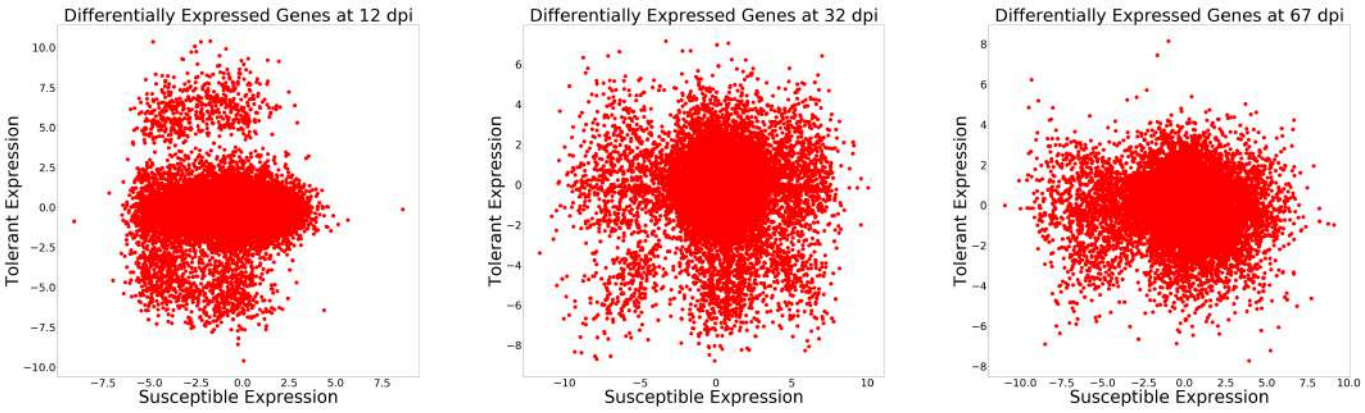
[9] The term state here is used to refer to host plant for which these values originate. The two host plants in this case being the tolerant and susceptible *Cassava* landrace.
[10] Susceptible to the South African cassava mosaic virus-SACMV.
[11] Tolerant to the South African cassava mosaic virus-SACMV.
[12] Days post infection(dpi).
[13] The exclusion of these values produce the "empty" region in Figures 3a, 3b and 3c.
[14] A gene whose differential expression is within the range in Equation 3.1.1, is not deemed to be biologically significant as it does not display a large enough change in count between the two different tested states and therefore is not looked at [38].

and excluding the range given in Equation 3.1.1). The exclusion of these points will not produce an incomplete dataset, as these genes who lack differential expression in either the tolerant or susceptible host are genes that did not display an adverse affect to the South African Cassava mosaic virus[15]. Recall that gene expression levels are gathered pre and post infection, the difference of which provides a differential expression value. If this value is not high enough then it implies that the gene was not adversely influenced by the infection. This is an important distinction because with both host immunity or host susceptibility the plants' response to the virus can be examined by looking at which genes were up or down regulated in response. The core idea being to see the difference between the susceptible and tolerant host. If a gene was differentially expressed in the tolerant host but not the susceptible host. This means that the gene in the susceptible host displayed no reaction to the insertion of the virus. The relating pathway is therefore not a target for the virus. The differential expression of the gene in the tolerant host is therefore more likely attributed to a knock-on effect from another gene or pathway as opposed to a direct relation to the virus. This relation is also relevant to the case where differential expression is seen in the susceptible host and not the tolerant host. Furthermore, this project will not provide a definite mechanism for immunity. Rather it aims to look at groups of genes that have been adversely affecting by the introduction of an infection and see how these reactions differ in a tolerant and susceptible host.

The datasets produced by matching the differentially expressed genes (DEGs) in the tolerant and susceptible host was then split and clustered by Cartesian quadrants. This could be considered a soft clustering step as division of genes by Cartesian quadrants provides a high-level grouping of the underlying data:

- **Quadrant 1** - genes are up-regulated in both the susceptible and tolerant host.
- **Quadrant 2** - genes are up-regulated in the tolerant host and down-regulated in the susceptible host.
- **Quadrant 3** - genes are down-regulated in both the tolerant and susceptible host.
- **Quadrant 4** - genes in susceptible host are up-regulated while genes intolerant host are down-regulated.

Figures 3a - 3c show the plots for the data described above.



(a) 12dpi (b) 32dpi (c) 67dpi

Figure 3: Plots depicting the differentially expressed genes of the *Cassava* landrace plant at 12, 32 and 67 Days Post Infection respectively. The host tolerant to the South African cassava mosaic virus - SACMV is represented on the y-axis while the susceptible host is represented on the x-axis in Figure 3a, 3b and 3c.

The results in the above figures present the raw values[16] of the data that has been provided. It is not advisable for these raw values to be used in the clustering methods that follow. This is because each of the methods proposed

[15] According to the experimental protocol that was used to collect this data.
[16] Values that have not undergone any preprocessing(scaling/normalisation).

in the following sections utilise some similarity metric to perform the partitioning. A similarity metric that- if the euclidean distance between two points is used- would produce results skewed by values which do not have a *variance* of one and a *mean* of zero[17]. This is because un-normalised data can have a large range with a data distribution that does not lend itself to statistical testing using euclidean distance. When un-normalised data is used as inputs for the euclidean distance equation it can produce varied results. This is owing to the fact that euclidean distance between two points is calculated by using the x and y values of the respective points. To further clarify this point, the euclidean distance is calculated as follows:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_n-p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}. \quad , \tag{3.1.2}$$

where d is the euclidean distance between points $\mathbf{p}$ and $\mathbf{q}$ in Euclidean n-space, as calculated by taking the difference between these points' respective x and y values[18].

The features in the data at 12dpi, 32dpi and 67dpi (datasets that will be called B12, B32 and B67 respectively) were therefore scaled/normalised by setting the variance to one and by removing the mean. This is done by using the following relation:

$$z = \frac{(x-u)}{s}, \tag{3.1.3}$$

where x is the unscaled value of a sample (data-point), u is the mean and s is the variance of the whole dataset respectively. Scaling each point and therefore the whole dataset in this way produces data that suitably resembles standard normally distributed datasets.

3.2 k-means Algorithm

The k-means algorithm presents a simple yet powerful starting point for any clustering process. This partitioning algorithm's aim is to group together points that are similar and thus revel the pattern that may exist within the data. The k-means algorithm defines a cluster as one which groups together points that have a similar feature[39].

The process, according to I. Davidson and S.Ravi, starts with the user defining the number of clusters expected to appear in the data, which is represented by k [40]. This k value in turn will refer to the number of centroids[19] the algorithm will initialise. Once the centroids have been defined, the algorithm then matches each point to its closest centroid with the intention of reducing the intra-cluster *sum-of-squares*. This goal is achieved by iteratively moving the initialised centroids until an optimal solution is reached. This optimal solution is defined by one which minimises the intra-cluster *sum-of-squares/inertia* of a model.

Given a set of N samples of X, which are clustered into k clusters, each with a mean of μ_j, the intra-cluster *sum-of-squares* is calculated as follows:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(\|x_i-\mu_j\|^2). \tag{3.2.1}$$

To minimise 3.2.1 the k-means algorithm proceeds as follows:

1. Initialise the algorithm by defining k centroids. This is generally done randomly.
2. Calculate the Euclidean distance between each data point in the set and each centroid that has been defined. Given two points with coordinates/values $\mathbf{p}$= (p_1,p_2) and $\mathbf{q}$= (q_1,q_2) the Euclidean Distance is defined as:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2}. \tag{3.2.2}$$

[17] A data set displaying a normal distribution would have a variance of one and a mean of zero.
[18] These x and y values relate to a points Cartesian coordinates.
[19] A centroid is simply the centre of a cluster.

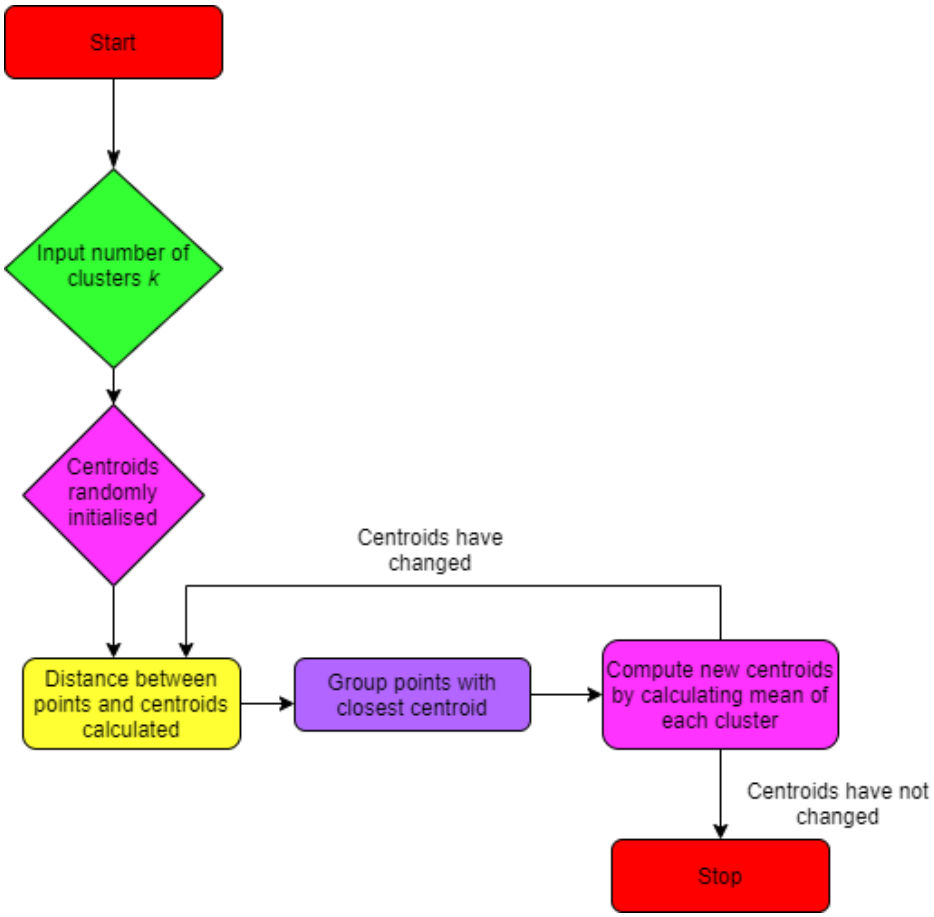
3. Each point is then assigned to the centroid that it is closest to, as calculated by the above equation. If the centroid of each cluster is defined as c_i, and each point as x then the assignment is done by applying the following:

$$\arg\min_{c_i \in C} \; dist(c_i, x)^2. \tag{3.2.3}$$

4. Calculate the average of each cluster, these values will be the new centroid for each cluster[20].
If S_i are the points that have been assigned to the ith cluster, then the new centroid is calculated by taking:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i. \tag{3.2.4}$$

5. Repeat steps 2 till 4 until a predefined limit for iterations is reached or until the centroids stop moving.



(a) k-means Algorithm

Figure 4: A flow diagram providing an outline of the steps involved in the k-means clustering process.

3.3 Hierarchical Clustering

In the initial stages of data analysis, there are two algorithms that are used extensively. These being the k-means algorithm, as derived in Section 3.2, and the Hierarchical Clustering algorithm.

[20] Taking the average or mean of each cluster as the new centroid is incidentally where the method got its name.

Hierarchical clustering is divided into two categories. These being:

- Agglomerative - initially each point is its own cluster, proceeding iteratively, closest clusters are merged until k number of clusters are reached [39].
- Divisive - starting with one exhaustive cluster at each step the clusters are divided until k clusters are reached [39].

The objective of both of these classes of algorithms is to build a binary tree with the data (also known as a dendrogram), which acts as a powerful visualisation tool[41]. In this section, only agglomerative clustering will be employed. The steps that need to be taken for this method, as proposed by Lance and Williams in [42], are outlined below.

Consider two different clusters represented by i and j, with the i^{th} cluster having n_i objects. Then let the distance between these clusters be d_{ij} and let $\mathbf{D}$ be the set that contains all other distances d_{ij}. If N is the total number of objects that need to be clustered then:

1. In the set $\mathbf{D}$, find the smallest value d_{ij}.
2. Let cluster i and j be merged, producing a new cluster k.
3. The new distances d_{km} need to be calculated using;

$$d_{km} = \alpha_i d_{im} + \alpha_j d_{jm} + \beta d_{ij} + \gamma \left| d_{im} - d_{jm} \right|, \tag{3.3.1}$$

   where m is any cluster that is not k. In $\mathbf{D}$, d_{im} and d_{jm} are replaced by the new distances and $n_k = n_i + n_j$. Lastly, α_i, α_j, β and γ are chosen according to the definition of similarity that is selected as outlined in Section 3.3.1.
4. Steps 1 to 3 need to be repeated until the set $\mathbf{D}$ contains only one group that holds N objects.
   The algorithm will therefore terminate at N - 1 iterations.

Clusters are merged by Hierarchical clustering if they are deemed to be suitably *similar*. This *similarity* criterion is dependent on which points in a cluster are selected in the measuring of distance between clusters. The following linkage methods presented in Section 3.3.1, provide the means in which to determine cluster similarity.

3.3.1 Cluster Linkage Methods

3.3.1.1 Single Linkage

This method is also called the *nearest neighbour method*. Here, the two closest members in two groups/clusters are used to calculate the distance between said clusters. This then produces clusters that add objects to a group in a sequential manner[42].
Looking back at equation 3.3.1 the following substitutions need to be made:

$$\alpha_i = \alpha_j = 0.5, \beta = 0, \gamma = -0.5. \tag{3.3.2}$$

3.3.1.2 Complete Linkage

This method is also called the *farthest neighbour method*. Here, the two farthest members in two groups/clusters are used to calculate the distance between said clusters. This then produces clusters relatively compact and well separated[42].
Looking back at equation 3.3.1 the following substitutions need to be made:

$$\alpha_i = \alpha_j = 0.5, \beta = 0, \gamma = 0.5. \tag{3.3.3}$$

3.3.1.3 Group Average Linkage

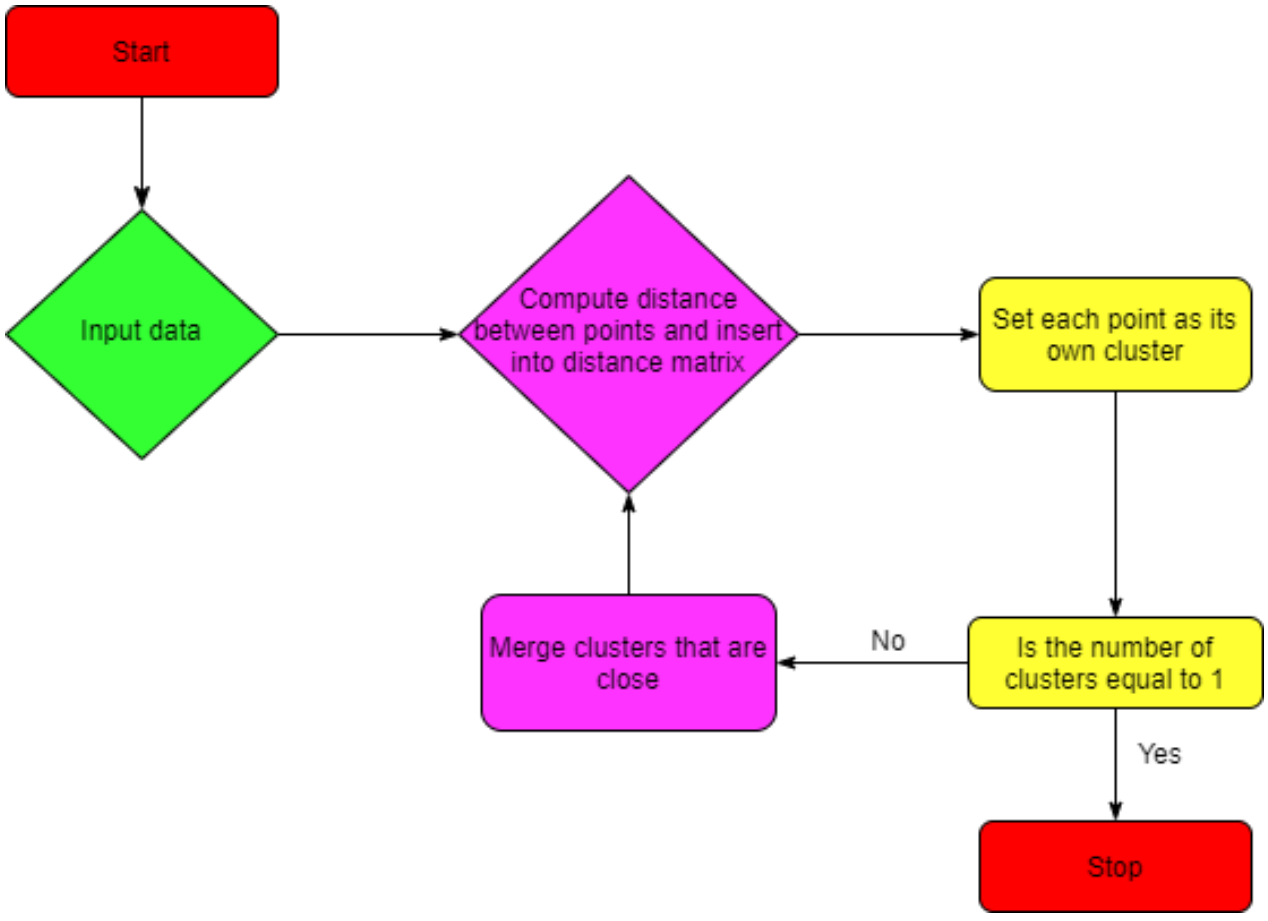
This method is the most commonly used in hierarchical clustering and is also called the *unweighted pair-group method*. Here, all members in two groups/clusters are used and the average distance between each of these members is calculated[42].
Looking back at equation 3.3.1 the following substitutions need to be made:

$$\alpha_i = \frac{n_i}{n_k}, \alpha_j = \frac{n_j}{n_k}, \beta = 0, \gamma = 0. \tag{3.3.4}$$

3.3.1.4 Ward's Minimum Variance

With this method it is not the distances between points that is solely considered. Rather, the aim is to merge clusters only if this results in a small increase in the within-group sum of squares. At the completion of the algorithm an optimal result would be one that minimises the sum of squares in each group[43].
Looking back at equation 3.3.1 the following substitutions need to be made:

$$\alpha_i = \frac{n_i + n_m}{n_k + n_m}, \alpha_j = \frac{n_j + n_m}{n_k + n_m}, \beta = \frac{-n_m}{n_k + n_m}, \gamma = 0. \tag{3.3.5}$$



(a) Hierarchical Clustering Algorithm

Figure 5: A flow diagram providing an outline of the steps involved in the Hierarchical clustering process.

3.4 Fuzzy Clustering

The fuzzy clustering algorithm is built using both fuzzy set theory and fuzzy logic [14]. It can be seen as in extension of the k-means methods that was presented in Section 3.2. The basis of the theory behind fuzzy clustering is that objects can belong to more than one cluster [14]. But an object does not necessarily belong to each cluster equally. There is a weight that is assigned to each object. A weight that describes *how much* an object belongs to a cluster. This style of clustering is particularly useful when the data does not have well-defined borders, as is the case with most *real-world* data. Instead of forcing points into a cluster for the sake of hard-assignments, a point can instead potentially belong to more than one cluster.

To clarify this point, the concept of membership needs to be introduced. In traditional set theory an object has the option of two membership values, these being 0 and 1. A point either belongs to a cluster, in which case its membership for that cluster is 1 while the membership for every other cluster is zero.

In contrast, with fuzzy set theory a point can have a membership value anywhere between 0 and 1. The only condition in this regard, is that the sum of these membership values needs to sum to 1.

Consider now that there is a set of n objects that represents the whole dataset:

$$\mathbf{X} = \{x_1, x_2, ..., x_n\}. \tag{3.4.1}$$

The result of the fuzzy clustering algorithm would produce k clusters, $C_1, C_2, ..., C_k$ with a partition matrix,

$$W = w_{ij} \in [0, 1], \ for\ i = 1, ..., n\ and\ j = 1, ..., k. \tag{3.4.2}$$

Here, w_{ij} is the weight element which informs on the degree of membership of the object x_i to the cluster C_j. While fuzzy clustering allows some flexibility in terms of cluster assignment, it does come with restrictions. These being:

- An object x_i may have multiple weights, but these weights need to sum to 1,

$$\sum_{j=1}^{k} w_{ij} = 1. \tag{3.4.3}$$

- Each cluster C_j needs to hold at least 1 non-zero weighted object., but this same cluster cannot hold all the objects in the set,

$$0 < \sum_{i=1}^{n} w_{ij} < 1 \tag{3.4.4}$$

Equipped with these conditions a *fuzzified* version of the k-means algorithm, according to Lobo in [14], is:

1. Initialise the algorithm by tentatively assigning weight values w_{ij} to each object, which will be called the fuzzy partition.
2. Repeat step 1.
3. Taking this partition into account, calculate the centroid[21] of each cluster:

$$c_j = \frac{\sum_{i=1}^{n} w_{ij}^{p} x_i}{\sum_{i=1}^{n} w_{ij}^{p}}, \tag{3.4.5}$$

   where c_j is the centroid of the cluster C_j, and $p \in [1, \infty)$ is a parameter that aids in the determination of the influence of the weights.

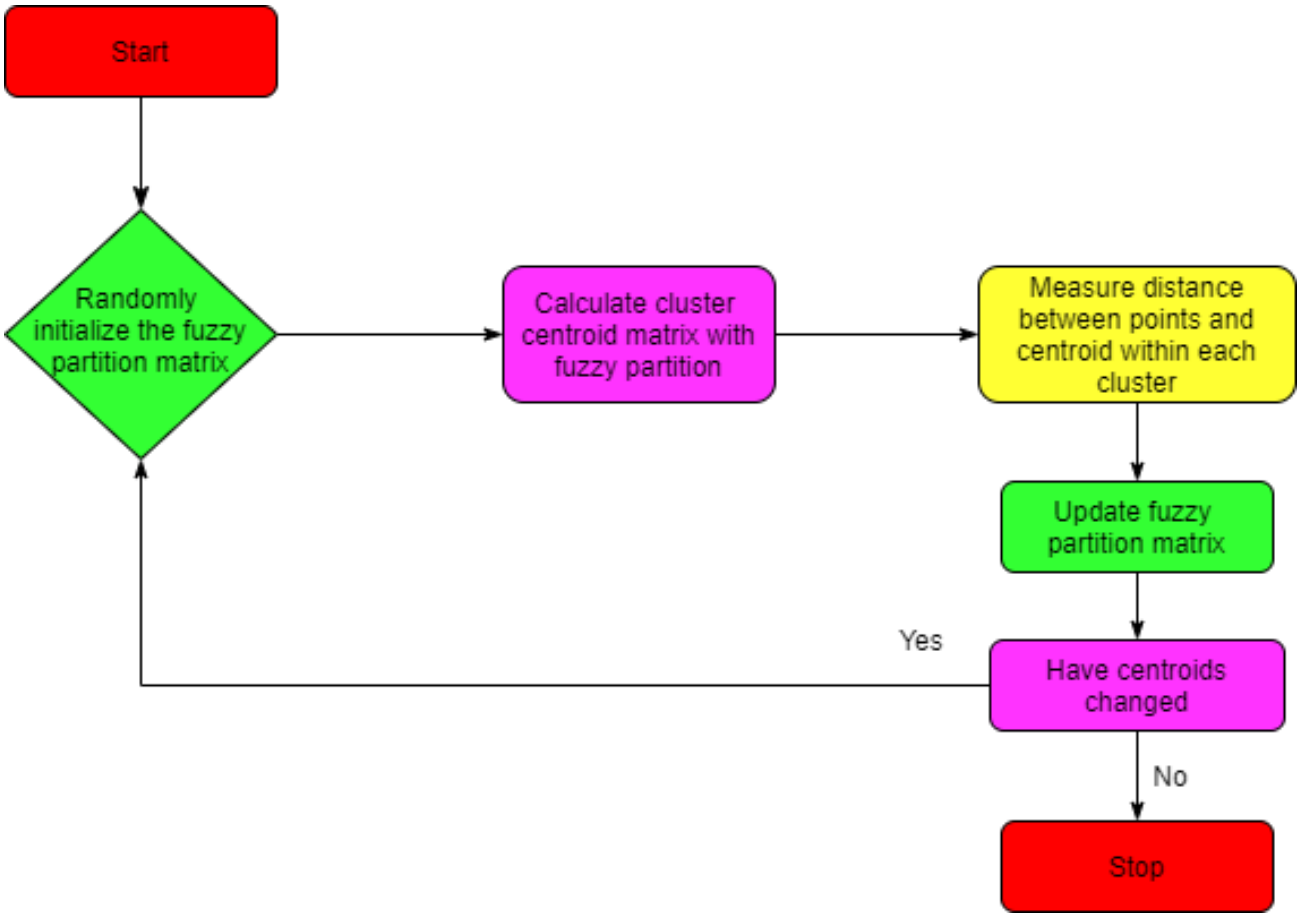
[21] The equation presented to calculate the centroid of a fuzzy cluster is simply an extended version of equation 3.2.4, which was used to calculate k-mean centroids.

4. Now that the new centroids have been calculated, the partition needs to be updated. This can be done by minimising the sum of square error(SSE)[22]. This produces the equation[23]:

$$w_{ij} = \frac{\left(\frac{1}{dist(x_i, c_j)^2}\right)^{\frac{1}{p-1}}}{\sum_{q=1}^{k}\left(\frac{1}{dist(x_i, c_q)^2}\right)^{\frac{1}{p-1}}}. \tag{3.4.6}$$

   If x_i is close to the centroid c_j, then w_{ij} will be high due to $dist(x_i, c_j)$ being low.

5. Repeat steps 2 to 4 until the centroids stop changing[24].



(a) Fuzzy Clustering Algorithm

Figure 6: A flow diagram providing an outline of the steps involved in the Fuzzy clustering process.

[22]The fuzzy c-means algorithm is similar to the k-means in that it aims to minimise the SSE. Small adaptions to the equation 3.2.1 produces the relation for sum of squares error in fuzzy clustering.

[23]The denominator in equation 3.4.6 is included as it normalises the weights for each point.

[24]There are alternate stopping criteria that can be selected. Namely, when the change in SSE is below a certain predefined value or when the change in weights of an object is below a predefined value.

3.5 Density Based Clustering

3.5.1 DBSCAN

The DBSCAN algorithm aims to make a distinction between core points and noise to cluster the given data [22]. To do this regions of high density which are separated by space that contains regions of lower density needs to be found.

In order to achieve this, there needs to be a formalised way of measuring the density of a data space. This is done using the following definitions as described by M. Ester *et al.* [21]:

1. The density at a point **P** can be measured by counting the number of points that fall in the circle that has a radius of ϵ from point **P**.
2. A dense region is then defined as a region where for each point in the cluster, the circle that has a radius of ϵ encompasses at least a certain amount of points. This minimum value of points is named *MinPts*

Therefore when looking at a point **P** that comes from a dataset **D**, the ϵ neighbourhood can be classified as:

$$N(p) = \{q \in \mathbf{D} | dist(p, q) \leq \epsilon\}. \tag{3.5.1}$$

By using the definition of a dense region, a core point is therefore one such point that satisfies the following relation,

$$|N(p)| \geq MinPts. \tag{3.5.2}$$

This will generally be the case in the interior region of a cluster.

A border point on the other hand is a point where the relation is not satisfied, meaning that there are less points than *MinPts* within the ϵ-neighbourhood of that point. But these points do lie within the ϵ - neighbourhood of a core point. This last part is what makes border points distinct from noise.

Points that are classified as noise are neither core points nor border points.

While these definitions provide a robust means to classify points, they are not without their drawbacks. The nature of the definitions of a border point and *MinPts* allow for ϵ - neighbourhoods around border points that could contain significantly less points than the neighbourhoods of core points. Furthermore, *MinPts* is defined at the onset of the algorithm and remains constant. Reducing this parameter is not an option either as this would result in more border points being categorised as core points therefore eliminating the algorithm's ability to detect noise efficiently.

To remedy this issue, there are two concepts that have been defined. These being density-reachable and density-connected points.

A data point a is referred to as directly density reachable from point b if:

- $|N(b)| \geq MinPts$; i.e. b is a core point.
- $a \in N(b)$; i.e. a is in the ϵ - neighbourhood of b.

It is clear that directly density reachable is not symmetric. This follows from the very definitions of a core and border point. While a core point will fall inside the ϵ - neighbourhood of a border point, a border point cannot have enough points in its own neighbourhood to satisfy the first condition above.

A point a is called density-reachable from point b with respect to ϵ and *MinPts* if -

- For a chain of points $b_1, b_2, ..., b_n$, where $b_1 = b$ and $b_n = a$, such that b_{i+1} is directly density reachable form b_i.

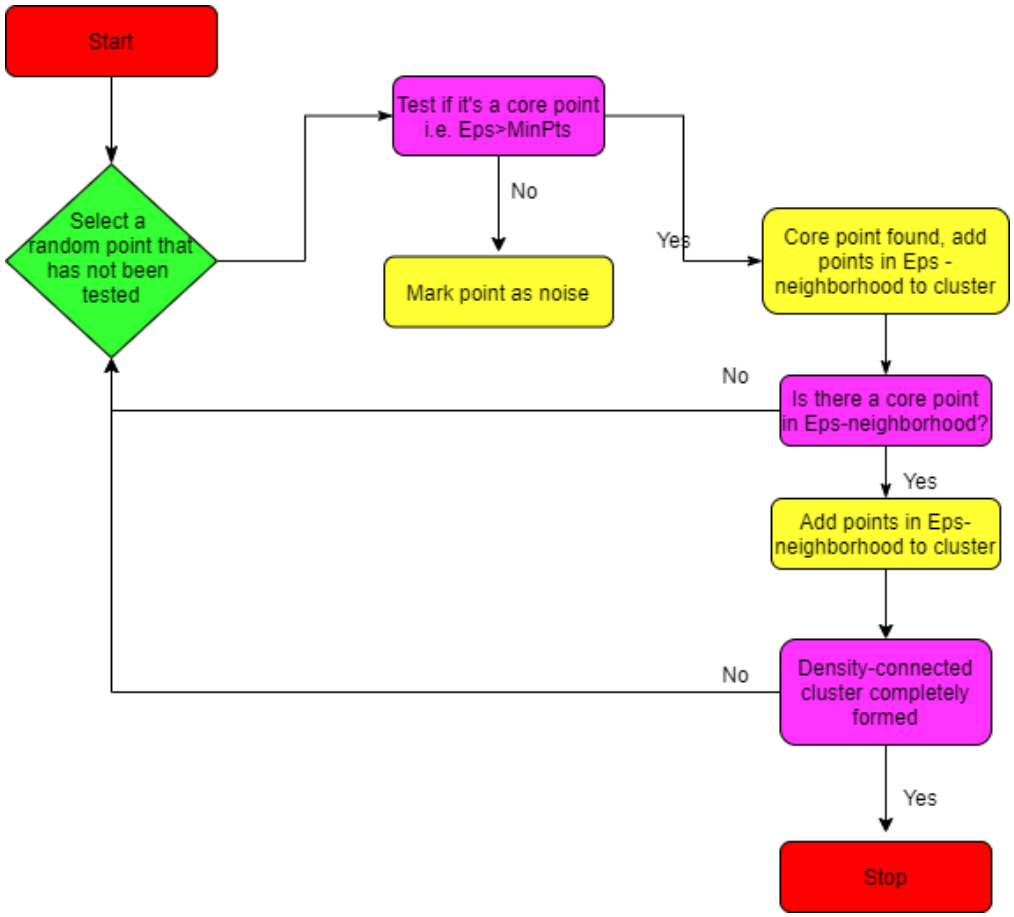
While density reachable may be transitive in nature, like directly density reachable, it is not symmetric.

For the definition of density connected, consider two points that do not share a core point but which are in the same cluster. These points are then considered to be density connected if there is a common core point that exists from which each point is density reachable. Unlike the last two definitions, density connectivity is symmetrical. Formally this relation is defined in the following way:

- A point a is considered to be density connected to a point b with respect to ϵ and *MinPts*, if there is a point c such that both a and b are density reachable from c with respect to ϵ and *MinPts*.

3.5.1.1 Steps of the DBSCAN Algorithm

1. Select a random point that has not yet been tested. Using the ϵ parameter calculate the neighbourhood information on this point.
2. If this point has at least *MinPts* in its ϵ - neighbourhood then cluster formation will begin. If not, then this point is labelled as noise. This is not necessarily a permanent assignment as this same point could land up in another points ϵ - neighbourhood, at which point it will be classified as a border point and it will be assigned to a cluster. It is here where the concepts of density reachable and density connected become important.
3. If a point is found to satisfy the core point conditions then it along with the points in its ϵ - neighbourhood will be added to the cluster. Furthermore, if points in this ϵ - neighbourhood are also found to be core points then their ϵ - neighbourhoods will also be added to the cluster.
4. The steps above will continue until a density - connected cluster is completely found.
5. The process will restart when a new point is found which could be the start of a new cluster or if a point is found to be noise.

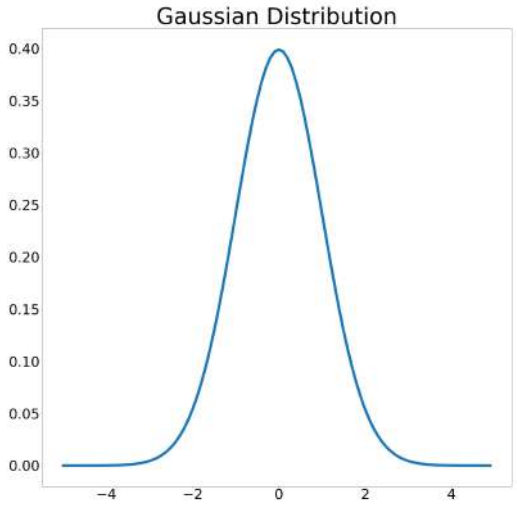


(a) DBSCAN Clustering Algorithm

Figure 7: A flow diagram providing an outline of the steps involved in the DBSCAN clustering process.

3.6 Gaussian Mixture Model

The Gaussian Mixture Model is technically a probability distribution and is a linear superposition of K Gaussians [10]. An example of a Gaussian distribution is provided in Figure In general the GMM assigns a data point to a cluster (assume there are K clusters) and then attempts to learn the parameters of the Gaussian.



(a) A Gaussian Distribution

Figure 8: An example of a standard Gaussian distribution. In the Gaussian Mixture Model each cluster is assumed to have such a distribution.

In the mathematical formulation as presented by K. Yeung *et al.* the *latent* variable $\mathbf{z}$ is used, which takes on values in the discrete set, {0,...,K} [10]. A latent variable is one which is not observed and the value for it is not known ahead of time. This latent variable $\mathbf{z}$, corresponds to the mixture component in the model.

Let $\mathbf{z} = z_1, ..., z_K$ with,

- $z_k \epsilon \{0, 1\}$ and $\sum_k z_k = 1$.

Joint distribution is defined as:

$$p(\mathbf{x}, z) = p(z)p(\mathbf{x}|z), \tag{3.6.1}$$

where,

- $p(z)$ is a multi-modal distribution.
- the parametric form is a Gaussian distribution, $p(\mathbf{x}|z)$.

The probability of each component z_k is given by:

$$p(z_k = 1) = \pi_k, \tag{3.6.2}$$

with,

- $0 \leq \pi_k \leq 1$ and,
- $\sum_k \pi_k = 1$.

As z uses 1-of-K we have the following,

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{3.6.3}$$

For a particular component,the conditional probability is given by:

$$p(x|z_k = 1) = N(x|\mu_k, \Sigma_k). \tag{3.6.4}$$

Therefore $p(x|z)$ can be written as,

$$p(x|z) = \prod_{k=1}^{K} N(x|\mu_k, \Sigma_k)^{z_k}. \tag{3.6.5}$$

By summing over all the possible states of z the marginal distribution of x is obtained:

$$p(x) = \sum p(z)p(x|z) = \sum_{z} \prod_{k=1}^{K} \pi_k^{z_k} N(x|\mu_k, \Sigma_k)^{z_k} = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k). \tag{3.6.6}$$

Equation 3.6.6 is the standard form of a Gaussian mixture with,

- $\mathbf{x}$ is a D-dimensional continuous-valued data vector.
- π_k, $k = 1, ..., K$ are the mixture weights.
- $N(\mathbf{x}|\mu_k, \Sigma_k)$ are component Gaussian densities with the form:

$$N(\mathbf{x}|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^k |\Sigma_k|} \exp\{\frac{-1}{2}(x - \mu_k)' \Sigma_k^{-1} (x - \mu_k)\}, \tag{3.6.7}$$

  - μ_k is the mean vector.
  - Σ_k is the covariance matrix.
- $\lambda = \{\pi_k, \mu_k, \Sigma_k\}$,$k = 1, ..., K$, are the parameters of the complete Gaussian mixture model.

If there are several variables, $x_1, ..., x_N$, since marginal distribution is represented as $p(x) = \sum p(z)p(x|z)$, then it follows that for every observed point x_n there is a corresponding latent variable z_n. This produces a form of Gaussian mixture that involves an explicit latent variable which means that the joint distribution $p(x|z)$ can be used, not the marginal distribution - $p(x)$. This allows for the simplification of the model through the Expectation-Maximisation algorithm.

3.6.1 Maximum Likelihood Parameter Approximation

To estimate the parameters of the Gaussian Mixture model, the log-likelihood of Equation 3.6.6 needs to be considered as presented by R. Nishii [44],

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln p(x_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k). \tag{3.6.8}$$

The following relationship is necessary in the maximising of the likelihood function. By letting the probability $p(z_k = 1|x) = \gamma(z_k)$, Bayes theorem gives:

$$\gamma(z_k) = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j)}, \tag{3.6.9}$$

where $p(z_k = 1) = \pi_k$ is taken to be the prior probability of component k and $\gamma(z_k) = p(z_k = 1|x)$ is the posterior probability once $\mathbf{x}$ has been observed.

The following conditions have to be satisfied at the maximum of the likelihood function:

1. Set the derivative of $\ln p(X|\mu, \Sigma, \pi)$ with respect to μ_k, to zero in (3.6.8) produces:

$$0 = -\sum_{n=1}^{N} \frac{N(x_n|\mu_k, \Sigma_k) \cdot \pi_k}{\sum_{j=1}^{K} N(x_n|\mu_j, \Sigma_j)\pi_j} \Sigma_k(x_n - \mu_k), \tag{3.6.10}$$

multiplying by $\sum_k^{-1}$ (which is non-singular) and rearranging gives:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n. \tag{3.6.11}$$

With,

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{3.6.12}$$

N_k is the effective number of points assigned to cluster k.

2. Set the derivative of $\ln p(X|\mu, \Sigma, \pi)$ in Equation 3.6.8 to zero ,with respect to Σ_k,

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_n)(x_n - \mu_n)^T. \tag{3.6.13}$$

3. Maximise $\ln p(X|\mu, \Sigma, \pi)$ with respect to mixing coefficients, μ_k. This is done by taking the Lagrange Multiplier and maximising the following inequality, (note here that mixing coefficients must sum to 1),

$$0 = \sum_{n=1}^{N} \frac{N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} N(x_n|\mu_j, \Sigma_j)\pi_j} + \lambda. \tag{3.6.14}$$

Multiply both sides by π_k and sum over k gives,

$$\pi_k = \frac{N_k}{N}. \tag{3.6.15}$$

Equations given in 3.6.11, 3.6.13 and 3.6.15 do not give a closed solution for the parameters of the model as the posterior probabilities, $\gamma(z_{nk})$, depend on Equation 3.6.9. The maximisation of the likelihood function in Equation 3.6.8, is done by using the Expectation-Maximisation algorithm.

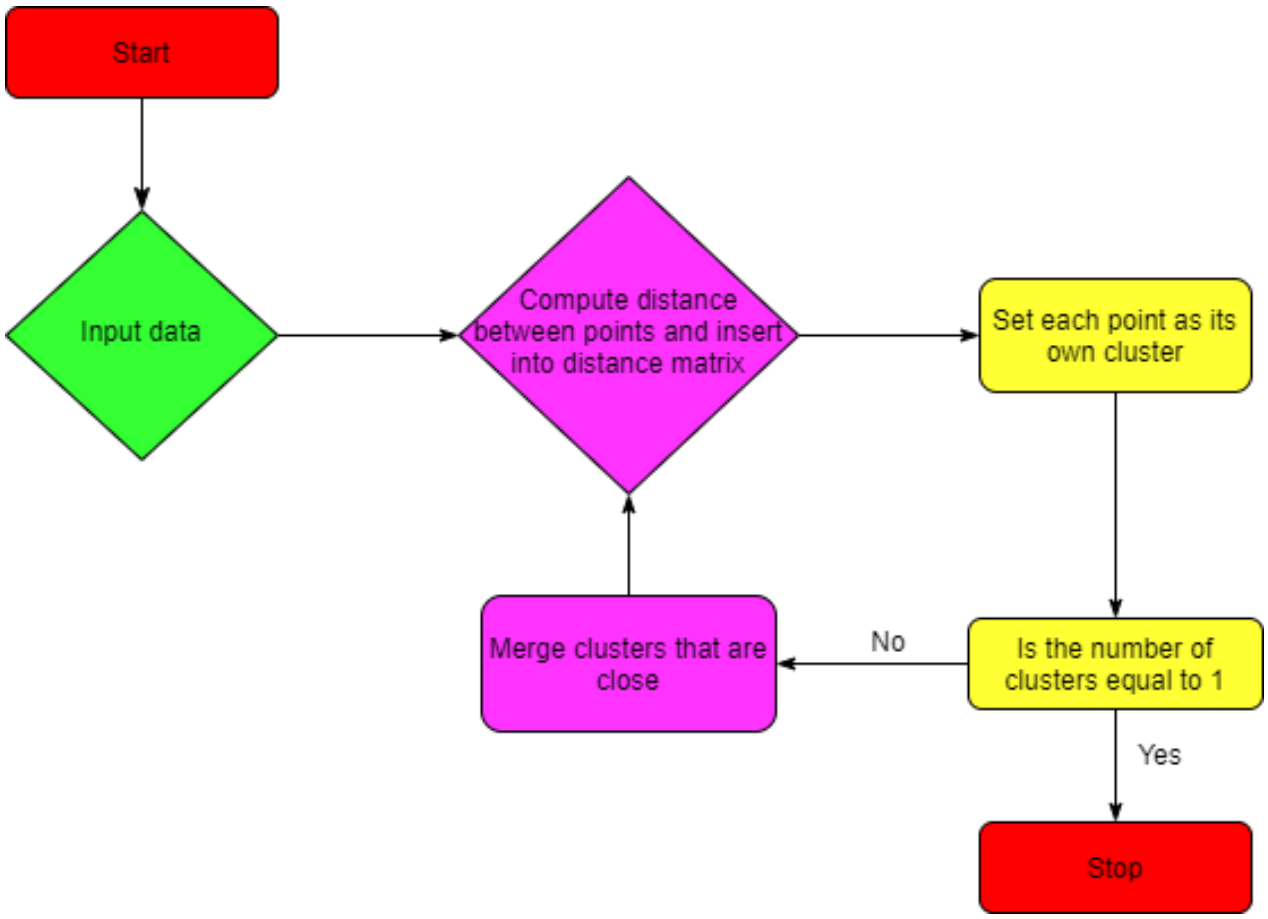
3.6.2 Expectation-Maximisation Algorithm

The EM algorithm is used to estimate the parameters of the mixture model [10]. It is a numerical technique for estimating the maximum likelihood function. The algorithm works iteratively from an initial guess of λ. At each step λ is updated until convergence is reached.

1. **E-Step** - Current values of parameters are used to calculate posterior probabilities. These initial values are obtained by using the k-means algorithm.

2. **M-Step** - Probabilities of the E-step are used to re-evaluate the means,covariances and mixing coefficients using Equations 3.6.11, 3.6.13 and 3.6.15.

The EM algorithm is guaranteed to coverage as it monotonically increases the likelihood function. In practice this convergence is slow and a threshold needs to be calculated.



(a) Gaussian Mixture Model

Figure 9: A flow diagram providing an outline of the steps involved in the Gaussian Mixture Model.

3.7 Affinity Propagation

In affinity propagation as proposed by B. Frey & D. Dueck, the space of data points is considered to be a network [23]. Here each point sends a message to every other point. The subject of these messages is to attempt to ascertain which points are more willing to become the exemplars of the dataset. An exemplar is a point which is the best suited to explain the other data points in the cluster, it is also the most important point in its cluster and there can only be one exemplar in each cluster.

The messages that are sent between data points are stored in two matrices:

1. The Responsibility Matrix (R): In this matrix f(i,k) indicates how well-suited the point k is to be the exemplar for another point i.

2. The Availability Matrix (A): In this matrix a(i,k) would indicate how appropriate it would be for point i to select k as its exemplar.

These two matrices can be interpreted as log probabilities which would make them negative valued. It is also important to note that they represent a graph in which each point is connected to every other point that exists in the data space. This same network is then encoded in a matrix where the index i,k would represent the connection

between point i and point k.

$$\text{Message Graph} = \begin{bmatrix} i,i & i,k_1 & \dots & i,k_n \\ \vdots & & & \vdots \\ k_n,i & k_n,k_1 & \dots & k_n,k_n \end{bmatrix} \tag{3.7.1}$$

3.7.1 Similarity

In an iteration, the first messages that are sent are the responsibilities and are based on a similarity function **S**.

$$\mathbf{S}(i,k) = -\|x_i - x_k\|^2, \tag{3.7.2}$$

which is the negative of the squared Euclidean Distance.

Once the similarity function is implemented a similarity matrix can be defined. This matrix would be a graph of the similarities that exist between all data points. **R** and A zeros matrices that are the same size as **S** also need to be initialised.

3.7.2 Responsibility

Now, responsibility messages can be defined by:

$$r(i,k) \leftarrow s(i,k) - max_{k' \text{ s.t. } k' \neq k}\{a(i,k^{'}) + s(i,k^{'})\}, \tag{3.7.3}$$

which can be implemented using a nested loop that would iterate over each row i, then $\max(\mathbf{A}+\mathbf{S})$ for that row can be determined for every index that is not equal to k or i as the latter would entail sending messages to itself. The damping factor in Equation 3.7.3 is included to aid in numerical stability and can therefore be regarded as a learning rate that slowly converges. According to the original writers of the method, damping factor should be set between 0.5 and 1.

Note: The term $s(i,k)$ is equal to the matrix **S**.

3.7.3 Availability

When looking at availability, there are two distinct cases that arise. The first case applies to points that do not fall on the diagonal of **A**, this would include messages that are sent from one data point to all other points in the dataset. The update here would be equal to the responsibility that is assigned by point k to itself and the sum of the responsibilities that have been assigned to k by the other data points - this excludes the current point. Due to the presence of the *min* function, the following will only hold true for negative values:

$$a(i,k) \leftarrow min\{0, r(k,k) + \sum_{i^{'} \text{ s.t. } i^{'} \notin \{i,k\}} max\{0, r(i^{'},k)\}\}. \tag{3.7.4}$$

The second case to deal with when it comes to availability is for the points that are on the diagonal of **A**, which would be the availability value that a data point would send to itself. The value of this message is equal to the sum of all the positive responsibility values that would be sent to the point in question. This is formalised as:

$$a(k,k) \leftarrow \sum_{i^{'} \neq k} max(0, r(i^{'},k)). \tag{3.7.5}$$

3.7.4 Exemplars

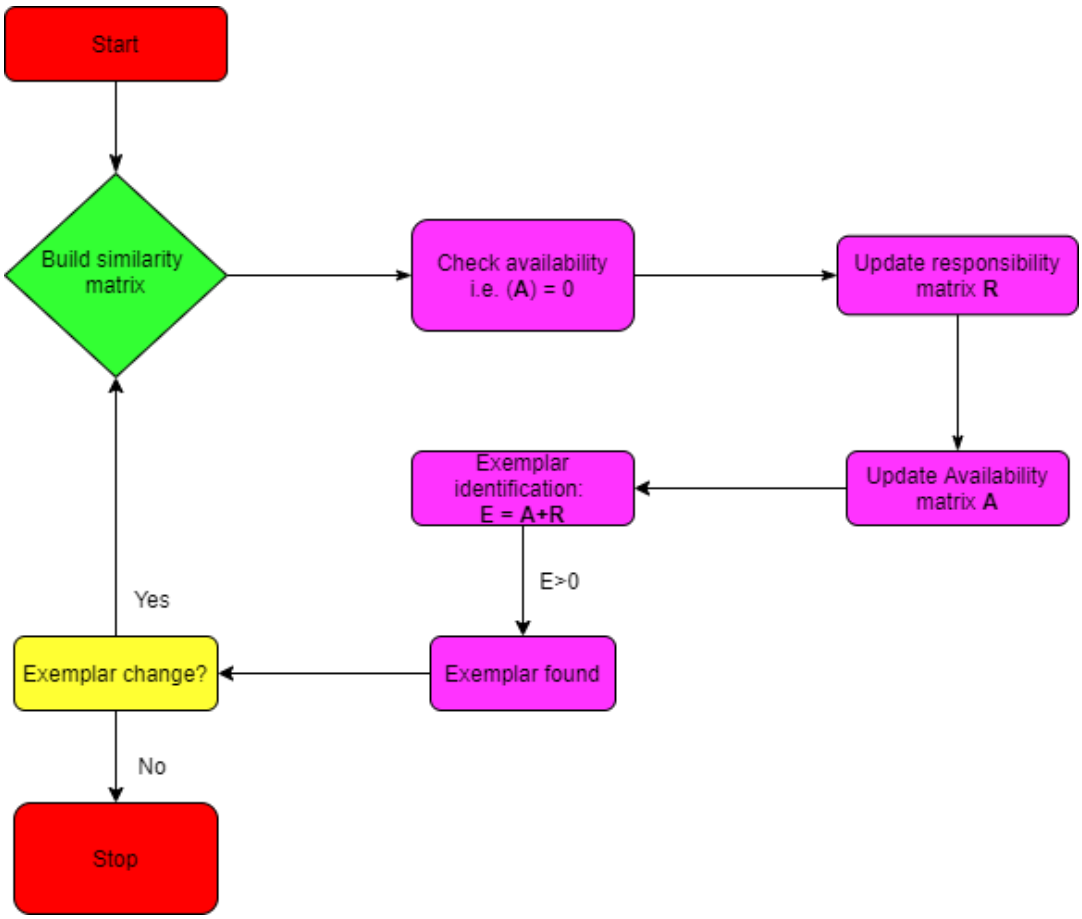
It is at this point in the process where the final exemplars can now be chosen by taking the maximum value of **A** + **R**,

$$\text{exemplar}(i,k) = max\{a(i^{'},k) + r(i^{'},k)\}. \tag{3.7.6}$$

3.7.5 Clustering

For the clustering itself, the algorithm requires the following inputs:

1. The number of iterations (this is necessary as the exemplars cannot change if the algorithm is to converge).
2. Damping Factor
3. Preference - This value will indicate how strongly a points thinks of itself as an exemplar. When there is limited a priori knowledge on the dataset, then the preference can simply be set to the median of the matrix **S**.



(a) Affinity Propagation Clustering Algorithm

Figure 10: A flow diagram providing an outline of the steps involved in the Affinity Propagation clustering process.

3.8 Spectral Clustering

Consider a dataset that has points $x_1, ..., x_n$ and a notion of similarity between all the data points represented by, $s_{ij} \geq 0$. In this case, the goal of clustering is to divide the data such that similar points fall in the same cluster while dissimilar points would be in different clusters. The spectral clustering process as elucidated by J. Shi & J. Malik proceeds as follows[17].

If the only information that is on hand is the similarity, then this can be used to reproduce a similarity graph $G = (V, E)$. In this graph the vertices v_i will represent the data points. If the similarity s_{ij} between two points

x_i and x_j is positive or larger than a predefined threshold then the corresponding vertices will be connected and the weight of the edge would be weighted by s_{ij}.

By representing the data like this, the task of clustering has been redefined. By looking for partitions of the similarity graph where the edges between groups have a low weight while the edges within a group have a high weighting, a solution has been found wherein points in different groups are dissimilar and points in the same group are similar.

3.8.1 Graph Notation

Let $G = (V, E)$ be an indirected and weighted graph that has a vertex set represented by $V = \{v_1, ..., v_n\}$. If graph G is weighted, then it follows that the edges between vertices v_i and v_j carries a non-negative weight such that $w_{ij} \geq 0$. Then the weighted adjacency matrix of this graph would be the matrix $W = (w_{ij})_{i,j=1,...,n}$. It follows that if v_i and v_j are not in fact connected then $w_{ij} = 0$. Furthermore, due to the undirected nature of the graph symmetry is required which would mean that $w_{ij} = w_{ji}$.

The degree of a vertex $v_i \in V$ is defined as:

$$d_i = \sum_{j=1}^{n} w_{ij}. \tag{3.8.1}$$

The values $d_1, ..., d_n$ from 3.8.1 will be on the diagonal of the degree matrix D. If the subset of vertices $A \subset V$ is given then its complement $V \backslash A$ by $\bar{A}$. The indicator vector is then defined as:

$$\mathbb{1}_A = (f_1, ..., f_n)' \in \mathbb{R}^n, \tag{3.8.2}$$

which has the entries $f_i = 1$ if $v_i \in A$ otherwise $f_i = 0$.
The shorthand, $i \in A$ for indices $\{i | v_i \in A\}$, has been introduced for the convenient handling of sums like $\sum_{i \in A} w_{ij}$.
For two sets $A, B \subset V$ that are not necessarily disjoint, the following can be defined:

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}. \tag{3.8.3}$$

The size of a subset $A \subset V$ can be measured in two different ways, these being:

$$|A| := \text{ number of vertices in A}$$

$$vol(A) := \sum_{i \in A}^{d_i}.$$

In equation (3.8.3.2) it is clear that $|A|$ would measure size by using the number of vertices, while $vol(A)$ would use the weights of the edges to measure size.
In a graph, a subset $A \subset V$ is only considered connected if any two vertices that exist in A can be joined by a path with the result being that all intermediate points too lie in A.
For a subset to be called a connected component, it needs to be connected and there can be no connections between the vertices in A and $\bar{A}$. If $A_i \cap A_j = \varnothing$ and $A_1 \cup ... \cup A_k = V$ then the sets $A_1, ..., A_k$ form a partition of the graph.

3.8.2 Similarity Graphs

When using pairwise similarities s_{ij} or pairwise distances d_{ij} to transform data points $x_1, .., x_n$ into a graph, there are a couple of methods to pick from. The goal of all these methods is to construct a model that would include the local neighbourhood relationships that exist between the data points.

3.8.2.1 ϵ - neighbourhood graph

The points that are connected in this type of graph are ones whole pairwise distances are less than ϵ. This would make ϵ - neighbourhood graphs unweighted due to the distances between connected points being relatively of the same scale, that is at most ϵ.

3.8.2.2 k-nearest neighbour graphs

A vertex v_i would be connected with a vertex v_j if the latter falls among the k-nearest neighbours of the former. This results in graphs that are directed due to the neighbourhood relationship being asymmetrical. If an undirected graph is desired then one can simply ignore the directions of the edges or only connect the vertices v_i and v_j if v_i is among the k-nearest neighbours of v_j and v_j is among the k-nearest neighbours of v_i. If the first option is selected then a k-nearest neighbours graph is produced, while the second would produce a mutual k-nearest neighbours graph. The choice of which option to use is irrelevant when considering the weighting of the edges, as in both cases the edges are weighted by the similarity of their endpoints.

3.8.2.3 The fully connected graph

All points that have a positive similarity are connected and the weight of their edges is determined by s_{ij}. This method is mostly used when mainly local neighbourhoods are already encoded into the similarity function. The reason for this being that similarities should model the local neighbourhood relationships that exist.

3.8.3 Graph Laplacian and their basic properties

Consider an indirected and weighted graph G, with a weight matrix W, where:

$$w_{ij} - w_{ji} > 0. \tag{3.8.4}$$

When dealing with the eigenvectors of a matrix, it is not necessarily assumed that they are normalised. They are however always ordered increasingly, respecting multiplicities. When talking about the "first k-eigenvectors", the eigenvalues that correspond to the k smallest eigenvectors are being referred to.

3.8.3.1 Unnormalised graph Laplacian

Define the unnormalised graph Laplacian matrix as:

$$L = D - W. \tag{3.8.5}$$

3.8.3.2 Normalised graph Laplacian

There are two matrices that are related to each other and are called normalised graph Laplacians. These are:

$$L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W,$$

where L_{sym} is a matrix that is symmetric and L_{rw} is related closely to a random walk.

3.8.4 Algorithms

To lay out the steps for the most common spectral clustering algorithms, assume that the data set in question consists of n points $x_1, ..., x_n$. Their pairwise similarities $s_{ij} = s(x_i, x_j)$ are measured by some symmetrical and non-negative similarity function which has a corresponding similarity matrix of $\mathbf{S} = (s_{ij})_{i,j=1,...,n}$.

3.8.4.1 Unnormalised Clustering

Inputs: The similarity matrix $S \in \mathbb{R}^{n \times n}$, number of clusters k.

1. Using one of the mentioned methods, construct a similarity graph and let the weighted adjacency matrix be W.
2. Compute the unnormalised Laplacian L.
3. Compute the first k eigenvectors $u_1, ..., u_k$ of L.
4. Let the vectors $u_1, ..., u_k$ be the columns in the matrix $U \in \mathbb{R}^{n \times k}$.
5. For $i = 1, ..., n$ let $y_i \in \mathbb{R}^k$ be the vector that corresponds to the i-th row of U.
6. Using the k-means algorithm cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}^k$ into the clusters $C_1, ..., C_k$.

Output: Clusters $A_1, ..., A_k$ with $A_i = \{j | y_j \in C_i\}$

3.8.4.2 Normalised Clustering

Inputs: The similarity matrix $S \in \mathbb{R}^{n \times n}$, number of clusters k.

1. Using one of the mentioned methods, construct a similarity graph and let the weighted adjacency matrix be W.
2. Compute the unnormalised Laplacian L.
3. Compute the first k eigenvectors $u_1, ..., u_k$ of the generalised eigen-problem $Lu = \lambda Du$
4. Let the vectors $u_1, ..., u_k$ be the columns in the matrix $U \in \mathbb{R}^{n \times k}$.
5. Normalise the rows of the matrix $U \in \mathbb{R}^{n \times k}$ to norm 1 by setting

$$u_{ij} = \frac{u_{ij}}{(\sum_k u_{ik}^2)^{1/2}}, \tag{3.8.6}$$

   therefore creating U

6. For $i = 1, ..., n$ let $y_i \in \mathbb{R}^k$ be the vector that corresponds to the i-th row of U.
7. Using the k-means algorithm cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}^k$ into the clusters $C_1, ..., C_k$.

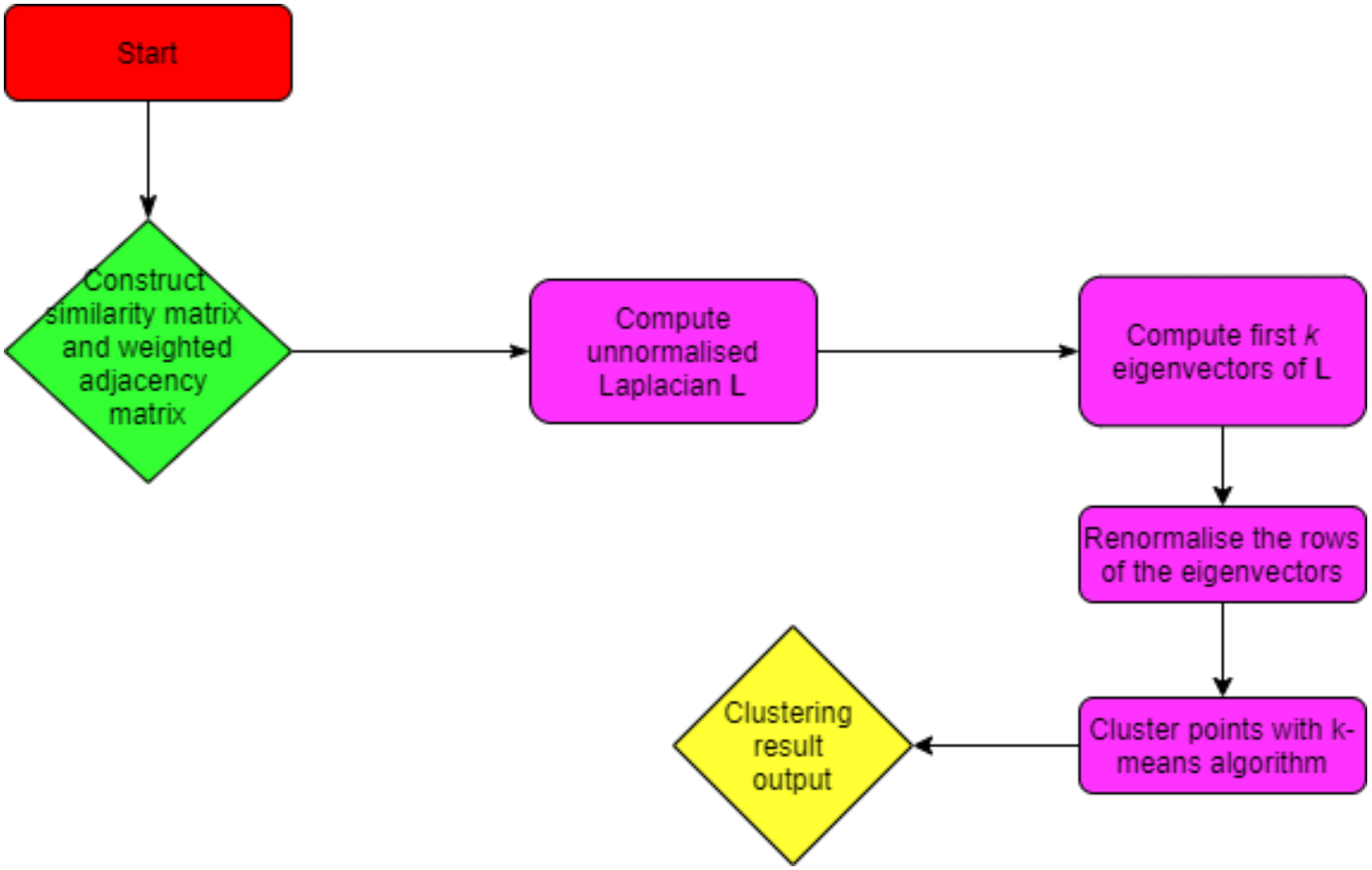
Output: Clusters $A_1, ..., A_k$ with $A_i = \{j | y_j \in C_i\}$

OR

Inputs: The similarity matrix $S \in \mathbb{R}^{n \times n}$, number of clusters k.

1. Using one of the mentioned methods, construct a similarity graph and let the weighted adjacency matrix be W.
2. Compute the unnormalised Laplacian L.
3. Compute the first k eigenvectors $u_1, ..., u_k$ of the generalised eigen-problem $Lu = \lambda Du$
4. Let the vectors $u_1, ..., u_k$ be the columns in the matrix $U \in \mathbb{R}^{n \times k}$.
5. For $i = 1, ..., n$ let $y_i \in \mathbb{R}^k$ be the vector that corresponds to the i-th row of U.
6. Using the k-means algorithm cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}^k$ into the clusters $C_1, ..., C_k$.

Output: Clusters $A_1, ..., A_k$ with $A_i = \{j | y_j \in C_i\}$



(a) Spectral Clustering Algorithm

Figure 11: A flow diagram providing an outline of the steps involved in the Spectral clustering process.

3.9 Mean Shift Clustering

This cluster method is non-parametric, meaning that it does not constrain the shape of the clusters and this particular method does not require prior knowledge of the cluster numbers for a dataset. It was presented by K. Fukunaga & L. Hostetler in the following way [24].

If n data points are given, $x_1, ..., x_n$ which is on a d-dimensional space $\mathbb{R}^d$, the multivariate kernel density estimate obtained with kernel $K(\mathbf{x})$ and window radius h is:

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right). \tag{3.9.1}$$

If a kernel is radically symmetrical then one only need define the profile of the kernel $k(\mathbf{x})$ which would satisfy the following:

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2), \tag{3.9.2}$$

where $c_{u,d}$ ensures the integration of $K(\mathbf{x})$ to 1 by being a normalisation constant. At the zeros of the gradient function $\nabla F(\mathbf{x}) = 0$ there are the modes of the density function.

Define the gradient of the density function estimator 3.9.1 as:

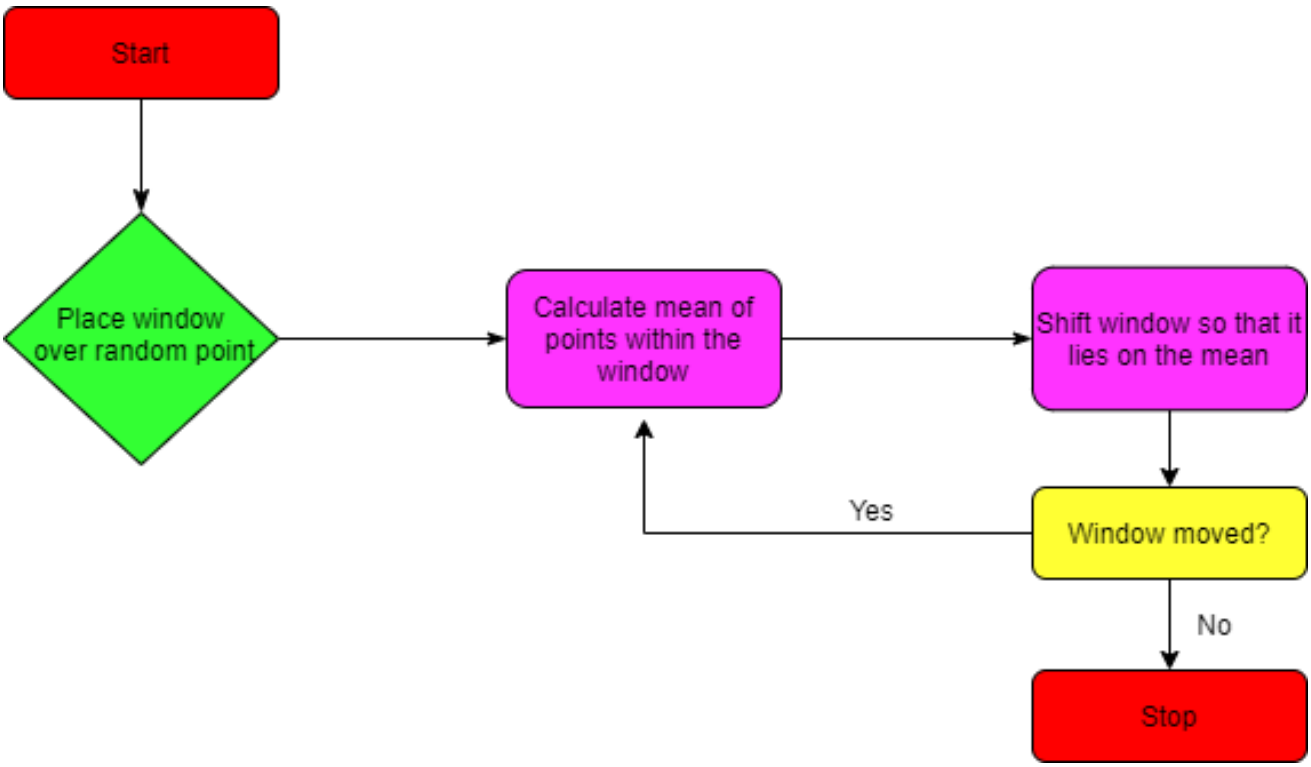
$$\begin{aligned} \nabla f(\mathbf{x}) &= \frac{2c_{k,d}}{nh^{d+2}} \textstyle\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g\left(\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\|^2\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\textstyle\sum_{i=1}^{n} g\left(\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\|^2\right)}{\sum_{i=1}^{n} g\left(\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\|^2\right)} - \mathbf{x}\right], \end{aligned} \tag{3.9.3}$$

where, $g(s) = -k'(s)$.The first term in Equation 3.9.3 is a proportional to the density estimate at $\mathbf{x}$ that is computed with the kernel $G(\mathbf{x}) = c_{g,d} g(\|\mathbf{x}\|^2)$, while the second term is what is defined as the Mean Shift.

The vector of the mean shift is one that always points in the direction of the maximum increase in the density. The mean shift procedure can now be defined by successively doing the following:

- computation of the mean shift vector which is $\mathbf{m}_h(\mathbf{x}^t)$.
- translation of the window, which is $\mathbf{x}^{t+1}. = \mathbf{x}^t + \mathbf{m}_h(\mathbf{x}^t)$.

This aforementioned procedure is guaranteed to converge to a point where the gradient of the density function would be equal to zero.



(a) Mean Shift Clustering Algorithm

Figure 12: A flow diagram providing an outline of the steps involved in the Mean Shift clustering process.

3.10 Advantages and Disadvantages of the Proposed Methods

Table 2: A summary of the advantages and disadvantages of the clustering methods that have been presented.

Advantages and Disadvantages of Methods		
Methods	**Advantages**	**Disadvantages**
K-means	• Relatively simple to implement • Scalable to large datasets • Convergence is guaranteed • Adapts easily to different and new datasets	• A value for k needs to be chosen manually • Random initialisation of centroids could skew results • Algorithm may not perform well if the underlying data has intuitive clusters that are different in size and have different densities • Is not efficient in dealing with outliers and noise • Does not perform well when dimensions of data are increased
Hierarchical	• Mathematically uncomplicated • Easy to implement computationally • The resulting dendrogram is visually easy to interpret • k does not need to be selected beforehand	• The solution that is found by the algorithm is rarely the global optimum, this is particularly problematic in higher dimensions when wrong solutions are harder to identify • As the data size increases so does the need for storage space • Time complexity becomes problematic with large datasets • There is no real mathematical objective with this algorithm which means that there is no point of convergence
Single Linkage	• Irregular cluster shapes can be detected as the algorithm makes no assumptions on the underlying cluster shapes	• The distance measure that is used here prefers to place points into existing clusters as opposed to creating new ones which can lead to chaining • Sensitive to outliers and noise
Complete Linkage	• Less susceptible than Single Linkage to outliers and noise	• Responds better to spherical shapes • Large clusters are more likely to be broken into smaller ones
Average Linkage	• Avoids chaining effect seen in Single Linkage • Less sensitive to noise and outliers	• Computationally takes more time than Single Linkage • More receptive to intuitive clusters that are globular in shape

Advantages and Disadvantages of Methods Continued		
Methods	**Advantages**	**Disadvantages**
Ward's Minimum Variance	• Is able to separate clusters well even when there is noise in between these clusters • Will produce desirable results when the underlying data has spherical multivariate normal distributions	• Struggles when the clusters are not all the same size • Responds very badly to intuitive clusters that are ellipsoidal
Fuzzy	• Able to identify clusters that overlap in the data space	• A priori specification of k is needed at the initiation of the algorithm • The usage of Euclidean Distance in the algorithms' calculations could produce an unequal weighting of the factors that underpin the data
DBSCAN	• Can efficiently distinguish between clusters of high and low density • Quite effective in identifying and separating outliers in a dataset	• Is less effective in separating multiple clusters that have similar densities
Gaussian Mixture Model	• No inherent bias towards spherical clusters. The algorithm is instead able to identify ellipsoidal clusters	• k needs to be known beforehand
Affinity Propagation	• k does not need to be known beforehand	• The time-complexity of this algorithm is substantially greater than other algorithms such as the k-means
Spectral Clustering	• Does not make a hard assumption on underlying cluster structure. Therefore non-spherical clusters can be found	• k needs to be known beforehand

3.11 Evaluating Cluster Validity

While non-parametric clustering may present a powerful method for data analysis and pattern discovery, without having true labels for the data it can be quite difficult to decide just how many clusters is the optimal number of clusters.

It is for this reason that there are a plethora of statistical tests that can help give an indication as to an optimal value for k [45]. Each of the tests presented below will look at certain features of the clusters produced by the different models to determine whether the result is suitable for the data or not. While these tests may give a good indication for the optimal value of k, they are not necessarily airtight. It is for this reason that more than one test will be used on the proposed dataset. Each test will be applied to the data and an optimal k will be chosen by taking all the results into account. This kind of analysis does not deal with absolutes. Instead, a broad analysis of the data and methods needs to be done if one is to select a model with some degree of confidence.

3.11.1 Sum of Squares Error (SSE)

Recall the Sum of Squares Error that was derived in equation 3.2.1. This same equation will be used when deciding what is the optimal value of k. To reiterate, the SSE for the k-means algorithm can be calculated as follows:

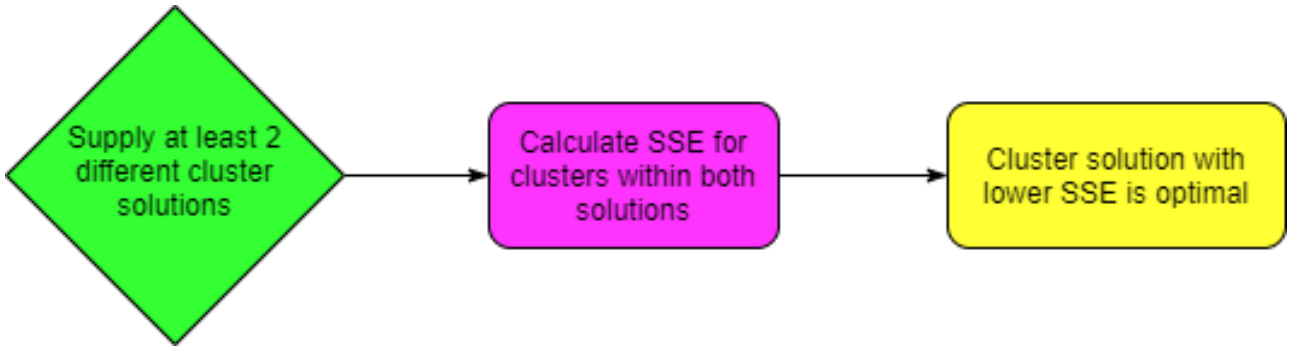
$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_i - \mu_j\|^2). \tag{3.11.1}$$

Taking this equation into account and equipped with at least two different clustering options, the model with the lower SSE is a relatively better solution for the dataset [46]. The easiest way to decrease this SSE value is to simply increase k. But, if k is increased too much then there is the danger of over-fitting the model[25].

Therefore, if this SSE index is to be used correctly a range of k needs to be tested. Naturally as k increases SSE will decrease, but the lowest value is not the definite optimal solution. It is suggested, rather that the optimal solution may exist at the *elbow* of the plotted SSE values [46]. This *elbow* value indicates where the decrease in SSE values is starting to peter out. SSE may still be decreasing, but after the elbow in the graph it is only decreasing minimally while k is still increasing at the same rate.

As demonstrated by T. Thinsungnoena *et al.* in [46], if the elbow is comprised of more than one point then it is advised to chose the point that is either at the start of the elbow region or in the middle. The corresponding k value at the end of the elbow region is potentially too high.

This index works best when it is applied to results obtained from the k-means method.

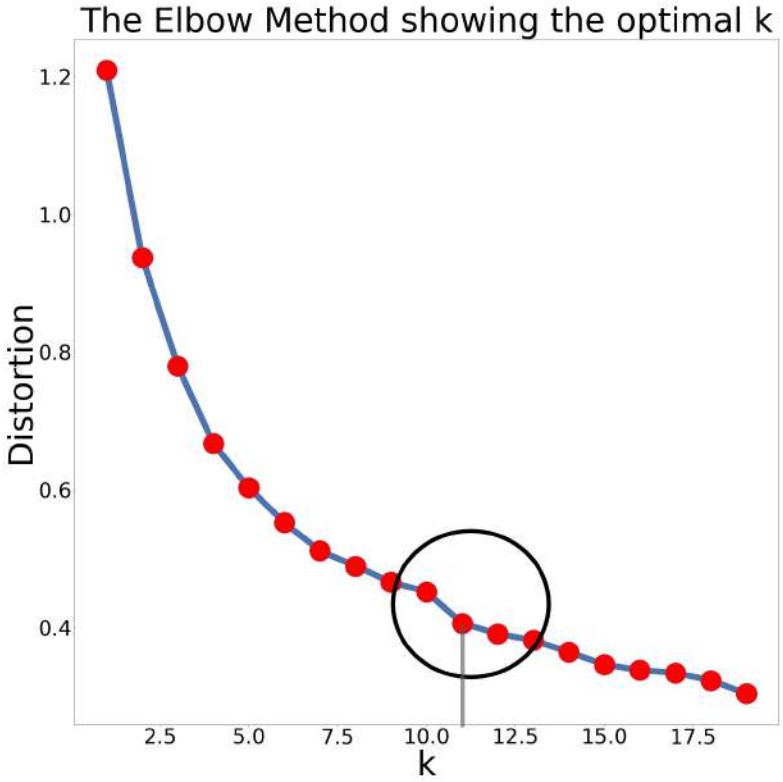


(a) *Sum-of-squares* error in model selection.

Figure 13: A flow diagram providing an outline of the steps involved in using the SSE metric for model validation.

When applying the process outlined in Figure 13a to select a value for k in cluster analysis, a graph similar to Figure 14a is produced.

[25] It makes sense that if each point was in its own cluster that the intra-cluster variation would be zero. While clustering methods aim to decrease intra-cluster variation, a solution where the chosen cluster number is too close to the amount of objects that are being clustered, is a degenerate and over-fitted solution.



(a) Hierarchical Clustering Algorithm

Figure 14: The graph produced when applying the SSE metric for the selection of k in cluster analysis. An optimal value for k can be found at the elbow of the curve. This area is outlined by the circle and relates to a value of 11 for k.

3.11.2 Silhouette Analysis

In Silhouette Analysis the distance between a point and the clusters that surrounds it, are of interest [47]. The value that is produced in this computation is called the **Silhouette Coefficient** and it has a range of $[-1, 1]$. If a value of $+1$ is observed, then the point that was tested is relatively far away from the cluster that surround it. This is a desirable result as it hints at the models' cohesion[26]. Conversely, if the silhouette coefficient is closer to -1 then the selected point is closer to its surrounding clusters than it is to its own. This indicates low cohesion in the resultant clusters. The last value of interest is one that is close to 0. This is a value that indicates neither low or high cohesion. It is a value that sits on the boundary and does not offer much information on the validity of the clusters.

It is worth noting, that these values and their respective analysis simply offer a guideline to interpret the generated silhouette coefficient. In reality, when testing a range of values for k on real-world data the silhouette coefficients produced for each model fit in a certain range of values. These do not necessarily span the length of

[26]Cohesion is a feature that ranks high in importance in Silhouette Analysis. As explained by Rousseeuw in [27], the cohesion of a cluster indicates the similarity of points in a cluster. A well-formed cluster will have points that are relatively similar while low cohesion will exist in an ill-fitting cluster.

$[-1, 1]$, but rather they exist within a subset of this range. In light of this, an optimal cluster number can be sought out by looking for the highest silhouette coefficient within this given range.

Given a cluster C_g, for the i-th object the following value needs to be calculated [47]:

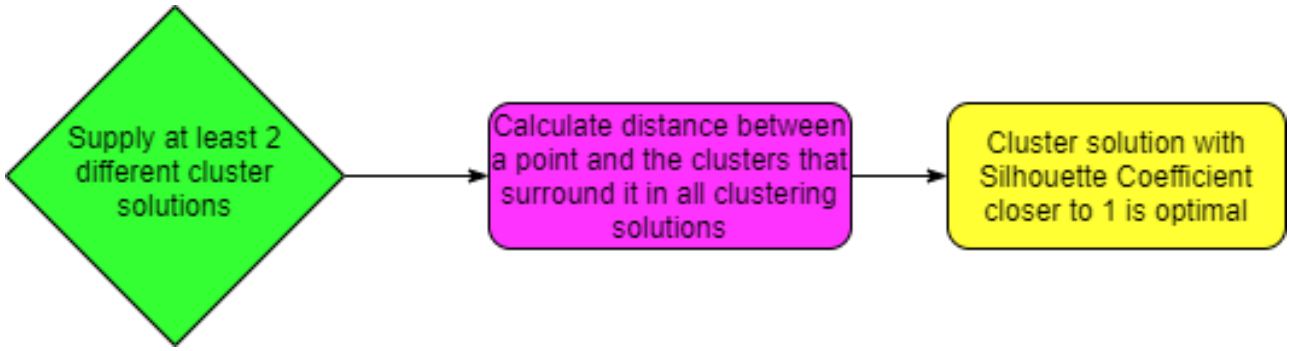
$$SW_i = \frac{b_i - a_i}{max\{a_i, b_i\}}, \tag{3.11.2}$$

where according to Rezankova in [47]:

$$ai = \frac{\sum_{j \in C_g (j \neq i)} D_{ij}}{n_g - 1}, b_i = min_{h \neq g} \left(\frac{\sum_{j \in C_h} D_{ij}}{n_h} \right). \tag{3.11.3}$$

Here, n_g and n_h is the number of objects that are placed in the g^{th} and h^{th} cluster respectively. The Silhouette Coefficient is therefore calculated by taking the average of the SW_i values:

$$SC = \frac{\sum_{i=1}^{n} SW_i}{n}. \tag{3.11.4}$$



(a) Silhouette analysis in model selection.

Figure 15: A flow diagram providing an outline of the steps involved in using Silhouette analysis for model validation.

3.11.3 Hartigan's Rule

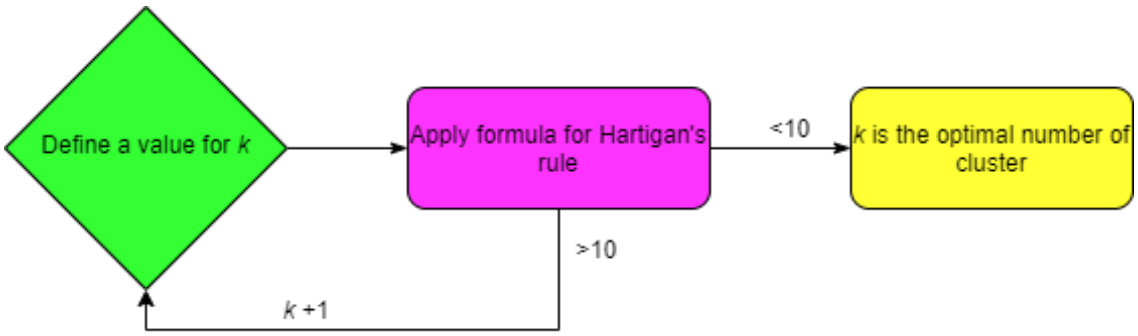
This method is not as complex as some of the other cluster validation techniques. It is simply a rule that was proposed by Hartigan in [28]. The idea is quite simple, if we define k as the number of clusters produced by the k-means algorithm and k+1 is one plus k amount of clusters then the following is calculated:

$$H(k) = (n - k - 1)\left[\frac{W(k)}{W(k+1)} - 1\right]. \tag{3.11.5}$$

If equation 3.11.5 is greater than 10, then Hartigan suggested that another cluster could be added. To calculate these values, the following *within cluster variation* equation will also be needed from the k-means algorithm:

$$\sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2. \tag{3.11.6}$$

Where x_i is a point that belongs to the cluster C_k having a mean of μ_k.



(a) Hartigan's Rule in model selection.

Figure 16: A flow diagram providing an outline of the steps involved in using Hartigan's Rule for model validation.

3.11.4 AIC and BIC

The Akaike Information Criterion was developed on the basis of information theory. The underlying idea is that information is lost when a model is used to try and describe real data. When AIC is calculated for multiple different models, the lowest value indicates which model is better [26]. This is because a low value represents a model in which a minimal amount of information is lost during its creation. In short, AIC tries to inform on how well the prevailing model explains/represents the data. To perform this calculation the following equation is used:

$$AIC = -2ln(l) + 2k, \tag{3.11.7}$$

where l is the maximum likelihood of the model[27] and k is the proposed number of clusters for the model being tested. The concepts of maximum likelihood were presented in Section 3.6.1.

The Bayesian Information Criterion is similar to the AIC in that it looks at maximum likelihood as a way to determine the quality of a model. But, it has a different penalty when it comes to parameters.

Parameters in the context of clustering is simply the amount of clusters chosen for each model. As stated previously, there is a danger of increasing the k value(parameters) to such a point that the model starts to over-fit. Both the AIC and BIC account for this in that they penalise a model if its complexity increases inordinately[28].

Calculation of BIC for a model requires the following equation:

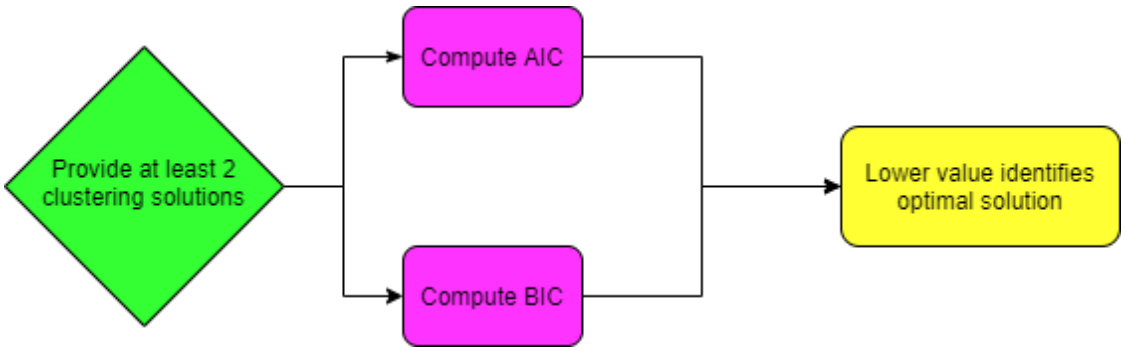
$$BIC = -2ln(l) + ln(n) \times k, \tag{3.11.8}$$

where n is the amount of objects(data points) to be clustered. Clearly this equation is almost identical to equation 3.11.7, with the exception of the last term. While AIC simply penalises parameters by multiplying them by 2, BIC goes one step further. This results in a harsher penalty when the complexity of a model increases.

As with the Sum of Squares Error, for both AIC and BIC there is no real value in only testing one or two models. In true unsupervised clustering a range of models needs to be examined. But this presents a range of AIC and BIC values to choose from. While low values are desirable, the lowest value is not necessarily the best model. It is far more likely that when plotted the AIC and BIC indices will produce graphs that contain multiple local minima and maxima. If this is the case then local minima may provide the values for an optimal k.

[27] The term maximum likelihood comes up frequently in the context of cluster validation. Maximum likelihood of a model derives its definition from information theory. It is a measure of how likely it is for the actual data to be produced if the only information at hand were the model itself. An effective clustering method aims to maximise this likelihood, thus the expression Maximum Likelihood [32].

[28] The more parameters a model has(the higher the k value) the more complex it becomes.



(a) AIC and BIC indices in model selection.

Figure 17: A flow diagram providing an outline of the steps involved in using AIC and BIC indices for model validation.

3.11.5 Calinski-Harabasz Index

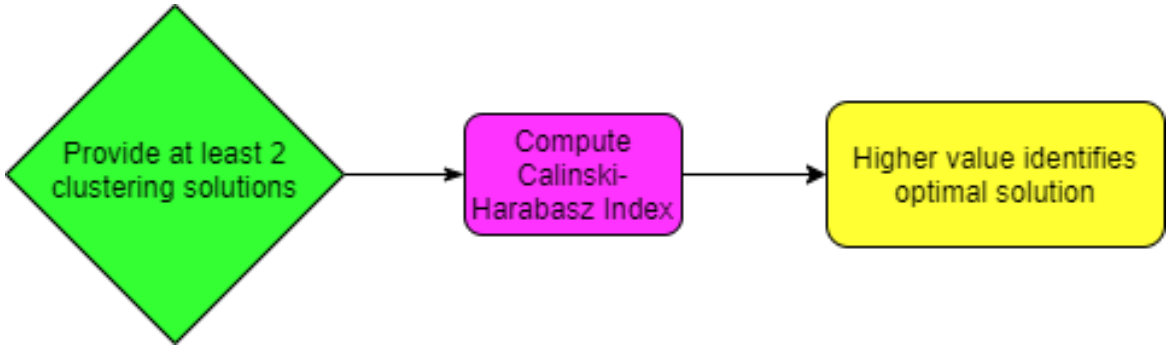
The Calinski-Harabasz Index evaluates a model by looking at how dense it is and how well-separated the clusters are[30]. A desirable model with dense clusters is one which has low intra-cluster variation, while well-separated clusters indicate a high inter-cluster variation[29].

Therefore, the CH index tests the validity of a model by calculating the average of intra-cluster sum of squares and inter-cluster sum of squares. This produces the following equation:

$$CH = \frac{\frac{\sum_i n_i d^2(c_i, c)}{k-1}}{\frac{\sum_i \sum_{x \in C_i} d^2(x, c_i)}{(n-k)}}, \tag{3.11.9}$$

where, n is the amount of objects to be clustered, c is the centre of the whole dataset, k is the number of clusters for the model, c_i is the centre of cluster C_i and x is an object.
Unlike the aforementioned cluster validation techniques, with the CH index a model with a high value is desirable. If a range of values for k are tested, it is not necessarily best to look at the highest CH values. Rather peaks in the plotted values should be investigated.

[29]High inter-cluster variation simply means that cluster are different to each other. This directly relates to the aim of clustering, which is to group together objects that are similar while clusters that represent object of different feature are relatively far apart.



(a) Calinski-Harabasz index in model selection.

Figure 18: A flow diagram providing an outline of the steps involved in using the Calinski-Harabasz index for model validation.

3.11.6 Davies-Bouldin Index

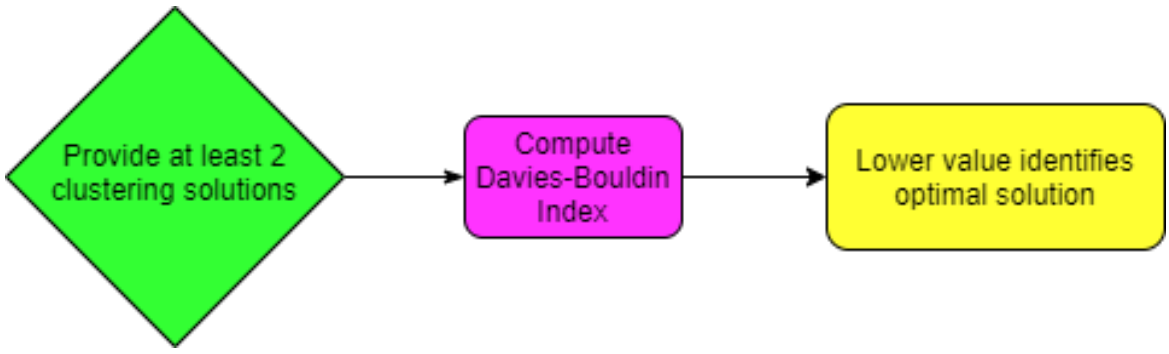
The Davies-Bouldin Index, measures the fit of a model by examining how distinct the clusters are[30]. This goes back to the idea that clusters should not be similar in a well-formed model. Instead, points that are dissimilar should not exist in the same cluster and furthermore, these clusters should not be close to each other.

Liu and Xiong in [30] explain that the DB index works by selecting a cluster C and then testing the similarity of this cluster to all the other clusters in a model. The highest value is assigned to cluster C and then this is done for each cluster. The average of all these values is calculated to produce the DB index:

$$DB = \frac{1}{NC}\sum_{i} max_{j,j\neq i}\left\{\left(\frac{1}{n_i}\sum_{x\in C_i} d(x,c_i) + \frac{1}{n_j}\sum_{x\in C_j} d(x,c_j)\right)\frac{1}{d(c_i,c_j)}\right\}, \tag{3.11.10}$$

where, n_i is the number of objects in C_i, k is the number of clusters for the model, c_i is the centre of cluster C_i, c_j is the centre of cluster C_j and x is an object.

As the aim of clustering is to ensure distinct and different clusters, a low DB index is sought after for an effective model.



(a) Davies-Bouldin index in model selection.

Figure 19: A flow diagram providing an outline of the steps involved in using the Davies-Bouldin index for model validation.

3.11.7 Gap Statistic

The sum of the intra-cluster distance between the points in a given cluster C_k which has n_k points can be calculated by:

$$D_k = \sum_{x_i \in C_k} \sum_{x_j \in C_k} \|x_i - x_j\|^2 = 2n_k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2. \tag{3.11.11}$$

If normalisation is added to the intra-cluster sums of squares then the result is measure of how compact the clusters are:

$$N_k = \sum_{k=1}^{K} \frac{1}{2n_k} D_k. \tag{3.11.12}$$

The variance method in Equation (3.11.12) is how the elbow method for the selection of k was built. While the method itself is robust, it can be quite difficult to select a single value for k when given a range of models. The gap statistic as proposed by R. Tibshirani *et al.*, attempts to formalise this selection and thus reduce a portion of the human error that exists within clustering [31].

This formalisation is done by attempting to standardise the comparison of the null reference distribution[30] of the data with $logW_k$. It is for this reason that an estimate was proposed for the optimal k. This is a value that would result in a $logW_k$ that falls the farthest below the reference curve in question. Formally, this is written as:

$$Gap_n k = E_n^*\{logW_k\} - logW_k, \tag{3.11.13}$$

where E_n^* is the expected value for a sample, whose size is n, that was taken from the reference distribution. An estimate for k would then be a value which maximises that value of $Gap_n k$ with the sample from the reference distribution taken into account. This estimate is general in nature which means that it can be applied to a multitude of distance measures $d_{ii'}$ and clustering methods.

The motivation for this method would require the clustering of n data points that are in p dimensions with k different cluster centres. If the assumption is made that these cluster centres are aligned in an equally spaced fashion then the expectation of the $log(W_k)$ would approximate to:

$$log\left(\frac{pn}{12}\right) - \left(\frac{2}{p}\right) log(k) + constant. \tag{3.11.14}$$

If a more accurate representation of the cluster number k is K then the $log(W_k)$ would be expected to decrease at a faster rate than its expected rate,

$$\left(\frac{2}{p}\right) log(k) \text{ for } k \leq K. \tag{3.11.15}$$

If $k > K$ then it would be a matter of adding an extra unnecessary cluster centre into the middle of a cloud that is essentially uniform. With simple algebra, it is easy to see that the $log(W_k)$ would decrease more slowly than its expected rate with these values for k. This translates to the gap statistic being at its biggest when $k = K$.

To create the reference datasets that are required, uniform sampling needs to be done within the bounding box of the original datasets.

To estimate $E_n^*\{log_k\}$ the average of B copies $logW_k^*$ needs to be computed. Each of these can be computed by taking a Monte Carlo sample $X_1^*, ..., X_n^*$ which would be drawn from the reference distribution.

The standard deviation $sd(k)$ of the $logW_k^*$ which is from the B Monte Carlo replicates can be turned into the following quantity if simulation error is accounted for,

$$s_k = \sqrt{1 + \frac{1}{B}} sd(k). \tag{3.11.16}$$

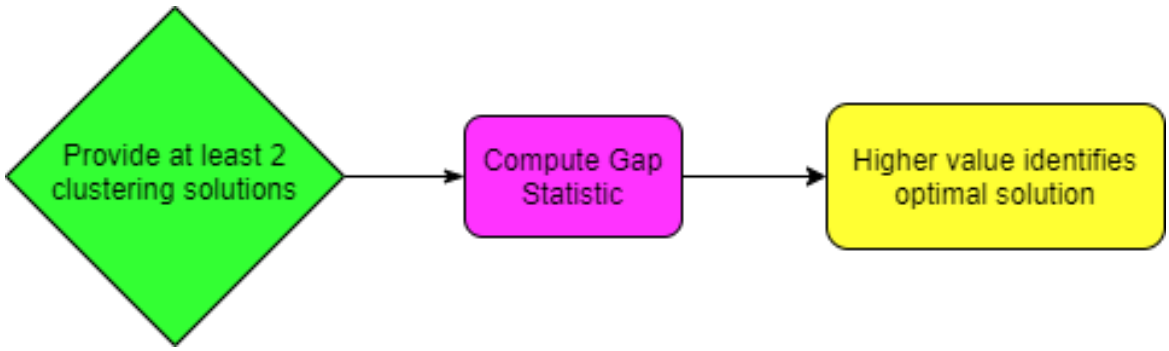
[30] This kind of distribution is one in which there is no obvious clustering.

This means that the optimal number of cluster K is the smallest value for k such that:

$$Gap(k) \geq Gap(k+1) - s_{k+1}. \tag{3.11.17}$$

3.11.7.1 Implementing the algorithm

1. Cluster the relevant data for $k = 1, ..., k_{max}$ and compute W_k for each.
2. Generate the B reference datasets and cluster these without vary cluster number $k = 1, ..., k_{max}$. Compute the Gap statistic using:
$$Gap(k) = \left(\frac{1}{B}\right) \sum_{b=1}^{B} logW^*_{kb} - logW_k. \tag{3.11.18}$$
3. Using:
$$\overline{w} = \left(\frac{1}{B}\right) \sum_{b} logW^*_{kb}, \tag{3.11.19}$$
compute the standard deviation,
$$sd(k) = \left[\left(\frac{1}{B}\right) \sum_{b} (logW^*_{kb} - \overline{w})^2\right]^{\frac{1}{2}}. \tag{3.11.20}$$
Then define:
$$s_k = \sqrt{1 + \frac{1}{B}}\, sd(k). \tag{3.11.21}$$
4. The optimal k would be the smallest value such that,
$$Gap(k) \geq Gap(k+1) - s_{k+1}. \tag{3.11.22}$$



(a) Gap Statistic in model selection.

Figure 20: A flow diagram providing an outline of the steps involved in using the Gap statistic for model validation.

3.12 Comparing the Results Obtained from each Method

Once an optimal k has been chosen for each dataset, and the aforementioned methods have been employed to cluster the data at said k, is there a way to compare the results in a quantifiable manner? An obvious answer is to compare

these results to external information that alludes to the true nature of the divisions that exist within the data. In the case of this project, it would be at this point that the true functions of the genes in the cassava plant would come into play. But, this information is not at hand. If it were, then the entire clustering process would not be necessary.

It is therefore proposed that the optimal models produced by each clustering method be tested against each other. This will require indices that are similar to the ones that have been presented in the previous section. The difference here being that an optimal result from one method will be chosen as the *true* clustering result and then the other models will be compared to it. This process will be repeated for each model. There also exists a technique called consensus clustering. This does not involve comparing models from different methods. Instead, the same model is calculated multiple times but only portions of the full dataset will be clustered each time. The assignment results from each of these calculations are then compared. Performing this step will provide a *consensus* as to which cluster an object belongs to, regardless of a change of parameters(only clustering portions of the dataset) or the amount of times the calculation is done[31].

This is not to say that the *true* nature of the data will be discovered during this process. But it will give an indication as to which methods agree on which assignments when it comes to the objects in question.

Consider this, if four out of ten models agree that a certain subset of the genes in question belong in the same cluster and thus share a differential expression pattern across tolerant and susceptible hosts. Then would it not be safe to say that, mathematically speaking, these points are similar to some degree? This is exactly a desirable result as the underlying assumption invoked when applying cluster analysis to differential gene expression data, is that genes with a similar expression profile are either co-expressed or share a common biological pathway/process. The methods that follow will be employed to do just that.

3.12.1 Adjusted-Rand Index

Without knowing the *true* partitions of the data, one can test the validity of the results obtained form clustering by searching for a measure of agreement between the proposed methods. This would entail taking the results obtained from the methods and then using indices or tests to determine just how much one method agrees with the other.

One such method is the Adjusted Rand Index as proposed by L. Hubert & P. Arabie [48]. To understand how this Index works, the Rand Index first needs to be derived and described.

Given a set of n objects

$$S = \{O_1, ..., O_n\}. \tag{3.12.1}$$

Suppose:

$$U = \{U_1, ..., U_R\}, V = \{V_1, ..., V_C\}, \tag{3.12.2}$$

which are two different partitions of the objects in S such that,

$$\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j \text{ and } u_i \cap u_i^{'} = \emptyset = v_j \cap v_j^{'}, \tag{3.12.3}$$

for,

$$1 \leq i \neq i^{'} \leq R \text{ and } 1 \leq j \neq j^{'} \leq C. \tag{3.12.4}$$

Suppose that U is some external criterion that has been offered for the data while V is the result that has been obtained from clustering. Furthermore, the following variables are defined:

- a number of pairs of objects in the same class in U and in the same cluster in V.

[31] As all of the clustering methods presented require a somewhat randomised initialisation step, there is a degree of change that can be observed if a method is run more than once, with the same parameters(k value) on the same dataset. This is why it is valuable to repeat the entire clustering process multiple times to see if the result lands up being the same. This offers a degree of confidence in object assignment within a chosen clustering method.

- b number of pairs of objects in the same class in U but not in the same cluster in V.
- c number of pair of objects in the same cluster in V but not in the same class in U.
- d number of pairs of objects that are in different classes and different clusters in both partitions.

The values a and d can be thought of as agreement indices while b and c indicate disagreements. The table below gives a simplified view of the variables that have been defined above.

	Same Class in U	Same Cluster in V
a	✓	✓
b	✓	×
c	×	✓
d	×	×

The Rand Index can now be defined, which is:

$$0 \leq \frac{a+d}{a+b+c+d} \leq 1. \tag{3.12.5}$$

With 0 being full disagreement while a value of 1 would be indicative of full agreement. There is however a problem that arises with the original Rand Index, this being that the expected value of the 2 random partitions does not take a constant value.

This is why the Adjusted Rand Index will be used as it assumes a generalised hypergeometric distribution[32] to account for the randomness of the model. This means that the U and V partitions are randomly picked in such a way the number of the objects in clusters and classes remains fixed throughout.

Now, let n_{ij} be the number of objects that are both in class u_i and cluster v_j. While n_i and n_j would be the number of objects that are in u_i and v_j respectively.

At this point the general form of an index which has a constant expected value[33] needs be defined. This is,

$$\frac{\text{index} - \text{expected values}}{\text{max index} - \text{expected value}} \leq 1. \tag{3.12.6}$$

The above equation would be equal to 0 when the expected value and index itself are equal.

Using the hypergeometric model it can therefore be shown that:

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right] = \frac{\left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}} \tag{3.12.7}$$

While the expression a + d can be simplified to:

$$\sum_{i,j}\binom{n_{ij}}{2}, \tag{3.12.8}$$

which is a linear transformation.

Therefore, the Adjusted Rand Index can be defined as:

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \frac{\left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \frac{\left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}}} \tag{3.12.9}$$

[32] A hypergeometric Distribution is a discrete probability distribution in which the success of a random draw without replacement for a population of finite size is described.

[33] An expected values is simply the probability-weighted average of all possible values in the set.

3.12.2 Mutual Information Based Scores

Assuming that there exits two labelling assignments for the same N objects. These being U and V, then their entropy as described by S. Romano *et al.* would be defined as the amount of uncertainty for a partition set and it would be defined as [49]:

$$H(U) = -\sum_{i=1}|U|P_i log(P_i). \tag{3.12.10}$$

Where,

$$P_i = \frac{|U_i|}{N} \tag{3.12.11}$$

is the probability that a randomly picked object from U would fall into the class U_i.
The same is true for V:

$$H(V) = -\sum_{j=1}|V|P_j^{'} log(P_j^{'}), \tag{3.12.12}$$

with,

$$P_j^{'} = \frac{|V_j|}{N}. \tag{3.12.13}$$

The mutual information between U and V would then be calculated using the following:

$$MI(U,V) = \sum_{i=1}|U|\sum_{j=1}|V|P_{i,j}\log(P_{i,j}P_iP_j^{'}). \tag{3.12.14}$$

Where,

$$P_{i,j} = \frac{|U_i \cap V_j|}{N}, \tag{3.12.15}$$

would be the probability that a randomly picked object would fall into both classes U_i and V_j.
If equation (3.12.14) were to be normalised then the result would be:

$$NMI(U,V) = MI(U,V)mean(H(U),H(V)). \tag{3.12.16}$$

Now the expected value for the Mutual Information Score needs to be calculated. This can be done with the following:

$$a_i = |U_i| \text{ and } b_j = |V_j|, \tag{3.12.17}$$

with a_i being the number of elements in in U_i while b_i is the number of elements in V_j.
Then,

$$\begin{aligned} E[MI(U,V)] = &\sum_{i=1}|U|\sum_{j=1}|V|... \\ &\sum_{n_{ij}=(a_i+b_i-N)+min(a_i,b_i)} n_{ij}N\log(Nn_{ij}a_ib_j)a_i!b_j! \\ &(N-a_i)!(N-b_j)!N!n_{ij}!(a_i-n_{ij})!(b_j-n_{ij})!(N-a_i-b_j+n_{ij})!. \end{aligned} \tag{3.12.18}$$

With this, the Adjusted Mutual Information Score can be calculated in much the same way as the Adjusted Rand Index:

$$AMI(U,V) = \frac{[MI(U,V) - E[MI(U,V)]]}{mean(H(U),H(V)) - E[MI(U,V)]}. \tag{3.12.19}$$

The mean that is used in Equation (3.12.19) is dependant on the data that is being studied. One such mean that can be used is the arithmetic mean.

One issue that arises with this measure is that neither the Mutual Information Score nor the Normalised Mutual Information Score are adjusted for chance. This means that the score will most likely increase with the number of clusters which will not necessarily indicate the actual mutual information that exists between the two label assignments.

3.12.3 Homogeneity, Completeness and V-measure

Here an intuitive metric needs to be defined using conditional entropy analysis [34]. For this, two objectives of cluster analysis need to be defined, these being:

- Homogeneity - Each cluster only contains members that belong to the same class.
- Completeness - Every member of a given class is assigned to the same cluster.

To calculate the homogeneity according to A. Rosenberg *et al.* of a label assignment define [34]:

$$h = 1 - \frac{H(C,K)}{H(C)}, \tag{3.12.20}$$

where,

$$H(C,K) = -\sum_{k=1}^{K}\sum_{c=1}^{C}\frac{a_{ck}}{N}\log\left(\frac{a_{ck}}{\sum_{c=1}^{C}a_{ck}}\right), \tag{3.12.21}$$

and,

$$H(C) = -\sum_{c=1}^{C}\frac{\sum_{k=1}^{K}a_{ck}}{C}\log\left(\frac{\sum_{k=1}^{K}a_{ck}}{C}\right). \tag{3.12.22}$$

For completeness,

$$c = 1 - \frac{H(C,K)}{H(K)}, \tag{3.12.23}$$

where,

$$H(C,K) = -\sum_{c=1}^{C}\sum_{k=1}^{K}\frac{a_{ck}}{N}\log\left(\frac{a_{ck}}{\sum_{k=1}^{K}a_{ck}}\right), \tag{3.12.24}$$

and,

$$H(K) = -\sum_{k=1}^{K}\frac{\sum_{c=1}^{C}a_{ck}}{C}\log\left(\frac{\sum_{c=1}^{C}a_{ck}}{C}\right). \tag{3.12.25}$$

With this, the weighted V_β can now be defined as the harmonic mean of homogeneity and completeness. The result being:

$$V_\beta = \frac{(1+\beta)hc}{\beta h + c}, \tag{3.12.26}$$

where β can be adjusted to favour either completeness or homogeneity.

3.12.4 Fowlkes-Mallows Scores

For the Fowlkes-Mallows Score as proposed by E. Fowlkes & C. Mallows, define N as the number of data points that are being sorted into K different clusters in the assignments A_1 and A_2 [50].

Then, define the matrix M such that:

$$M = [m_{ij}]_{k\times k}. \tag{3.12.27}$$

Where m_{ij} will define the points that fall in the i^{th} cluster of A_1 and the j^{th} cluster of A_2.

The Fowlkes-Mallows Score for the parameter k can be given by use of the following:

$$B_k = \frac{T_k}{\sqrt{P_k Q_k}}, \tag{3.12.28}$$

$$T_k = (\sum_{i=1}^{k}\sum_{j=1}^{k} m_{ij}^2) - N, \tag{3.12.29}$$

$$P_k = (\sum_{i=1}^{k}(\sum_{j=1}^{k} m_{ij})^2) - N, \tag{3.12.30}$$

$$Q_k = (\sum_{j=1}^{k}(\sum_{i=1}^{k} m_{ij})^2) - N. \tag{3.12.31}$$

The following terms also need to be defined in the context of the aforementioned conventions,

- True Positive(TP) - The number of pairs of data points that are in the came cluster in both A_1 and A_2.
- False Positive(FP) - The number of pairs of data points that are in the same cluster in A_1 but not in A_2.
- False Negative(FN) - The number of pairs of data points that are in the same cluster in A_2 but not in A_1.
- True Negative(TN) - The number of pairs of data points that are neither in the same cluster in A_1 nor in A_2.

Therefore,

$$TP + FP + FN + TN = \frac{n(n-1)}{2}, \tag{3.12.32}$$

and the Fowkles-Mallows Score can now finally be defined as:

$$\begin{aligned} B_k &= \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \\ &= \sqrt{\left(\frac{TP}{TP+FP}\right)\left(\frac{TP}{TP+FN}\right)} \\ \Rightarrow B_k &= \sqrt{\text{Precision} \times \text{Recall}}. \end{aligned} \tag{3.12.33}$$

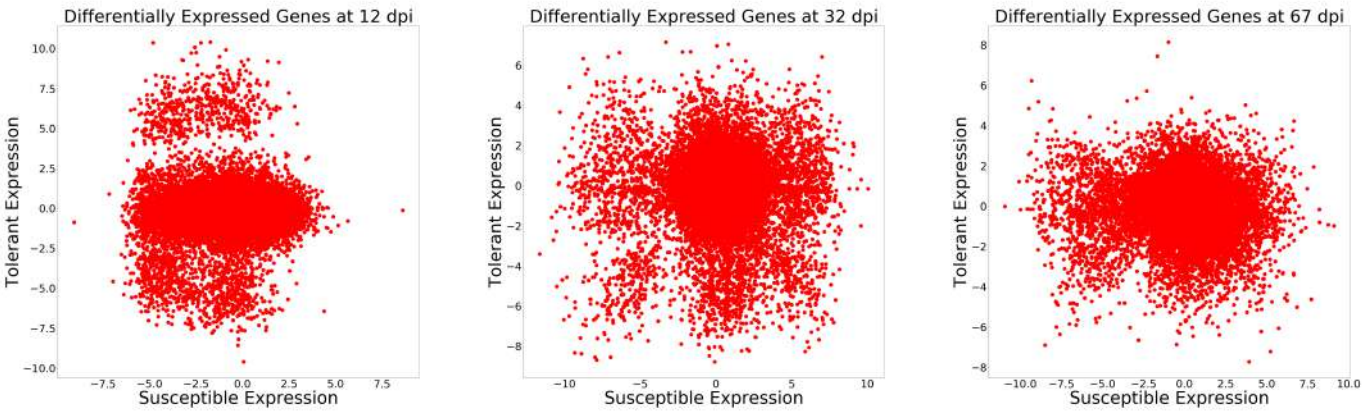
It is now clear that the Fowlkes-Mallows is simply the geometric mean of the precision and the recall.

One attractive aspect of this score is that there is no assumption made regarding the structure of the underlying clusters.

4 Results

4.1 Preparation of the Data

Figures 21a, 21b and 21c below shows the raw datasets at B12, B32 and B67 respectively.



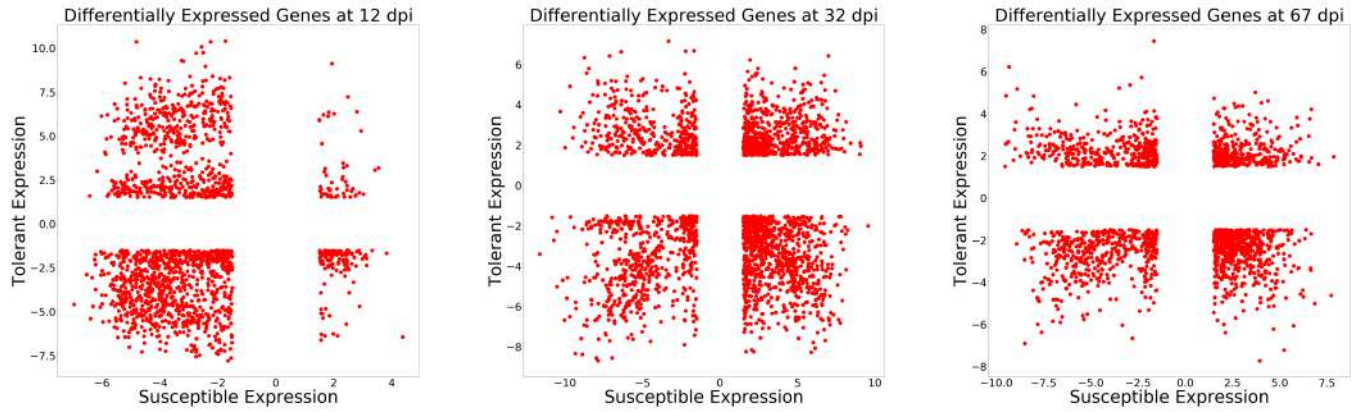
(a) B12 (b) B32 (c) B67

Figure 21: Plots depicting the differentially expressed genes of the *Cassava* landrace plant at 12, 32 and 67 Days Post Infection respectively. The host tolerant to the South African cassava mosaic virus - SACMV is represented on the y-axis while the susceptible host is represented on the x-axis in Figure 21a, 21b and 21c.

Only biologically significant genes will be clustered with the proposed methods. This is achieved by excluding genes whose differential expression(log2FoldChange value) falls within the following range:

$$-1.5 < log2FoldChange < 1.5. \tag{4.1.1}$$

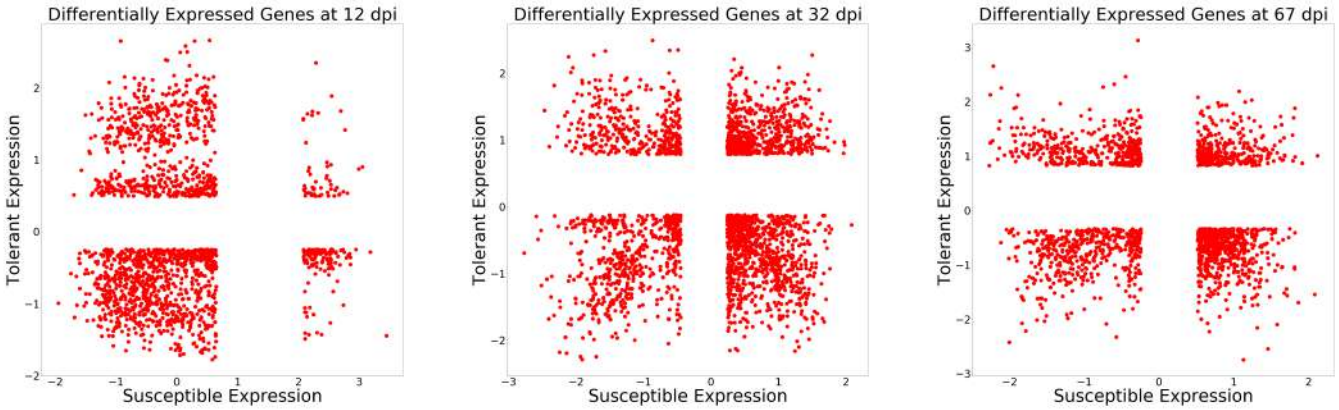
Implementation of Equation 4.1.1 produces the datasets as depicted in Figure 22a, 22b and 22c.



(a) B12 (b) B32 (c) B67

Figure 22: Plots depicting the differentially expressed genes of the *Cassava* landrace plant at 12, 32 and 67 Days Post Infection respectively. The host tolerant to the South African cassava mosaic virus - SACMV is represented on the y-axis while the susceptible host is represented on the x-axis in Figure 22a, 22b and 22c. Genes who had a differential expression between the range given by Equation 4.1.1 were excluded as they are not deemed biologically significant.

Lastly, the genes whose differential expression were deemed biologically significant were normalised thus producing Figures 23a, 23b and 23c for the B12, B32 and B67 dataset respectively.



(a) B12 (b) B32 (c) B67

Figure 23: Plots depicting the differentially expressed genes of the *Cassava* landrace plant at 12, 32 and 67 Days Post Infection respectively. The host tolerant to the South African cassava mosaic virus - SACMV is represented on the y-axis while the susceptible host is represented on the x-axis in Figure 23a, 23b and 23c. Differentially expressed genes were normalised using Equation3.1.3 to reduce the impact of a large range on the *Euclidean distance* calculations.

4.2 Decision of Optimal k

Table 3 below provides a summary of the options for k as were provided by implementing the tests in Section 3.11. These tests were implemented in *Python* 3.7.4 with the aid of the *scikit-learn* package[35].

Table 3: Table showing the options for k as found by the statistical tests presented in Section 3.11.

		SSE	SA	HR	AIC	BIC	CH	DB	Gap	k
B12	Q1	8, 10	4, 7, 14	8	3, 12, 14	3, 12, 14	8	7, 14, 16	7, 14, 18	14
	Q2	8	16	23	21	4, 18, 21	8, 18	17, 22	8 , 22	22
	Q3	14	9, 19	38	12, 27	7	4, 15, 25	9, 23, 26	3	25
	Q4	13, 15	12, 14	14	12, 18	4	13	4, 19	9, 14, 19	14
B32	Q1	21	8, 21	33	10, 21	6	21	24	3	21
	Q2	20	16, 25	26	16, 24	4	15, 21	25	3	25
	Q3	22, 25	9, 17 or 23	38	9	5	23	10, 17 or 22	2, 13 or 23	23
	Q4	19, 21	10, 16, 18, 21	41	7, 16, 25	7, 11, 16	19, 23, 26	16, 18	2, 23	21
B67	Q1	16, 23	17, 20	21	11, 24	3, 21	18	20, 24	3	21
	Q2	16, 25	16	28	16, 21	9	11, 17, 22	15, 26	11, 20, 24, 26	26
	Q3	15, 25	21	17	12, 14	4	20, 22, 24	20, 24	2, 25	24
	Q4	11, 19	18, 23	25	14, 27	4, 11	21, 24	24, 27	27	24

With data wherein there exists poorly defined natural groupings, the majority of the statistical tests in Section 3.11 will not provide only one option for an optimal k. There will instead be a couple of values for k that are

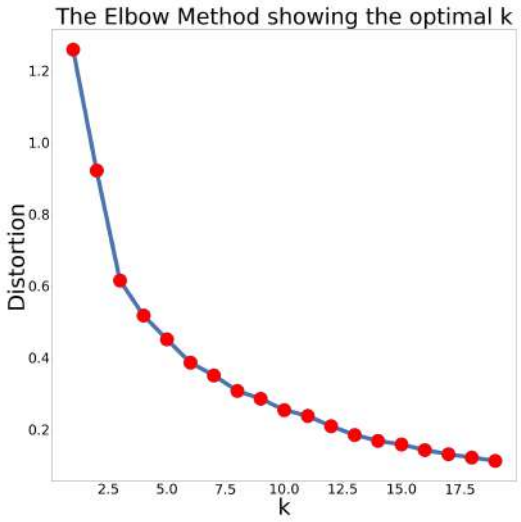
proposed for each model. This is why there exist more than one option for k in each column. The first column in the above table refers to the data set - either B12, B32 or B67. The second column indicates the quadrants in all three of the datasets. The following columns provide results from the *Sum of squares error*, Silhouette Analysis, Hartigan's Rule, AIC, BIC, Calinski-Harabasz Index, Davies-Bouldin Index and Gap Statistic respectively. The last column give the value of k that was chosen for each quadrant in each dataset by taking all 8 tests into account.

To reduce these options down to a single value for k, a consensus was sought after among the potential values for each quadrant at each time step. This is comparable to looking for the mode of a dataset. But it is slightly more complex than that as the mode for each quadrant need be an intelligent option when compared to the resulting clustering for said value of k.

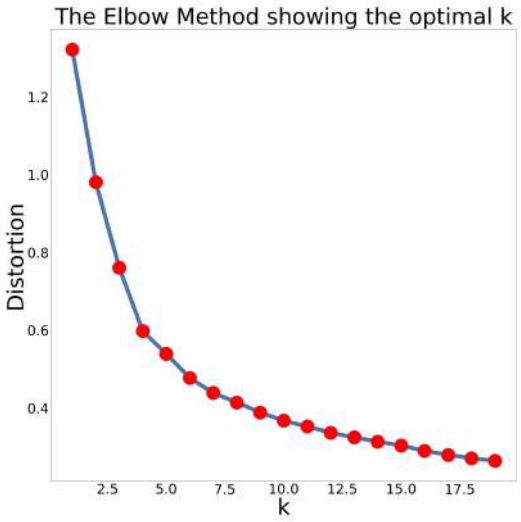
Another factor that needs to be considered when evaluating the above table is that all of the tests, excluding the AIC and BIC, are executed by performing k-means clustering. This implies that these tests presume that the optimal k for a model will be one that produces clusters that are circular and evenly sized. This is not necessarily the type of clusters that fit best to a dataset of this nature.

The figures that follow are graphical depictions of the results in the above table. The evaluation of these figures led to the decision of an optimal k which is shown in the final column.

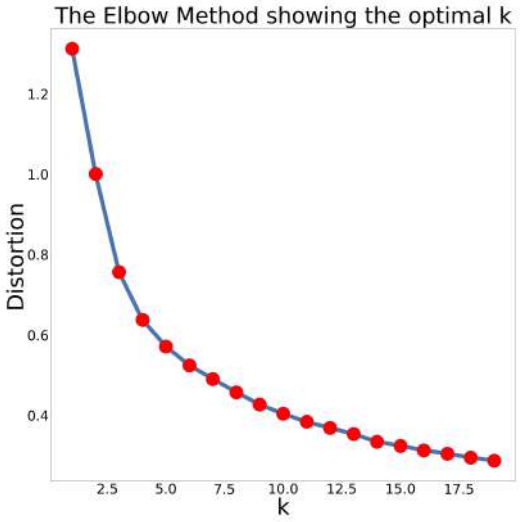
4.2.1 Sum of Squares Error for B12 Dataset



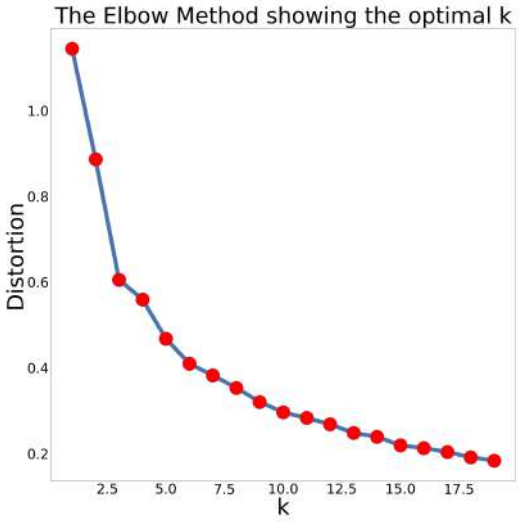
(a) Quadrant 1



(b) Quadrant 2



(c) Quadrant 3



(d) Quadrant 4

Figure 24: Plots depicting the SSE for B12 dataset. Optimal values for k can be found at the *elbow* of each plot.

Recall that the sum of squares error gives an indication as to the variation within a cluster. It is therefore desirable to have a low SSE value as this would indicate a model wherein points that are numerically similar are grouped or clustered together. Referring to Figure 24a, a global minimum does not necessarily indicate the optimal k. Rather, local minimums are sought out at the *elbow* of the graph. In the figure in question, this would be either at k values of 8 or 10[34]. The superior of these options can only be determined when compared to the other methods used to calculate an optimal k.

[34]These *elbows* in the graph are easier to see when the result is magnified. But they exist where there is a distinct drop in the distortion between two points, followed by only a slight drop in distortion corresponding to the next point. This pattern creates the elbow. Either the first or second point making up a three-point elbow are selected.

4.2.2 Silhouette Analysis for B12 Dataset - Quadrant 1

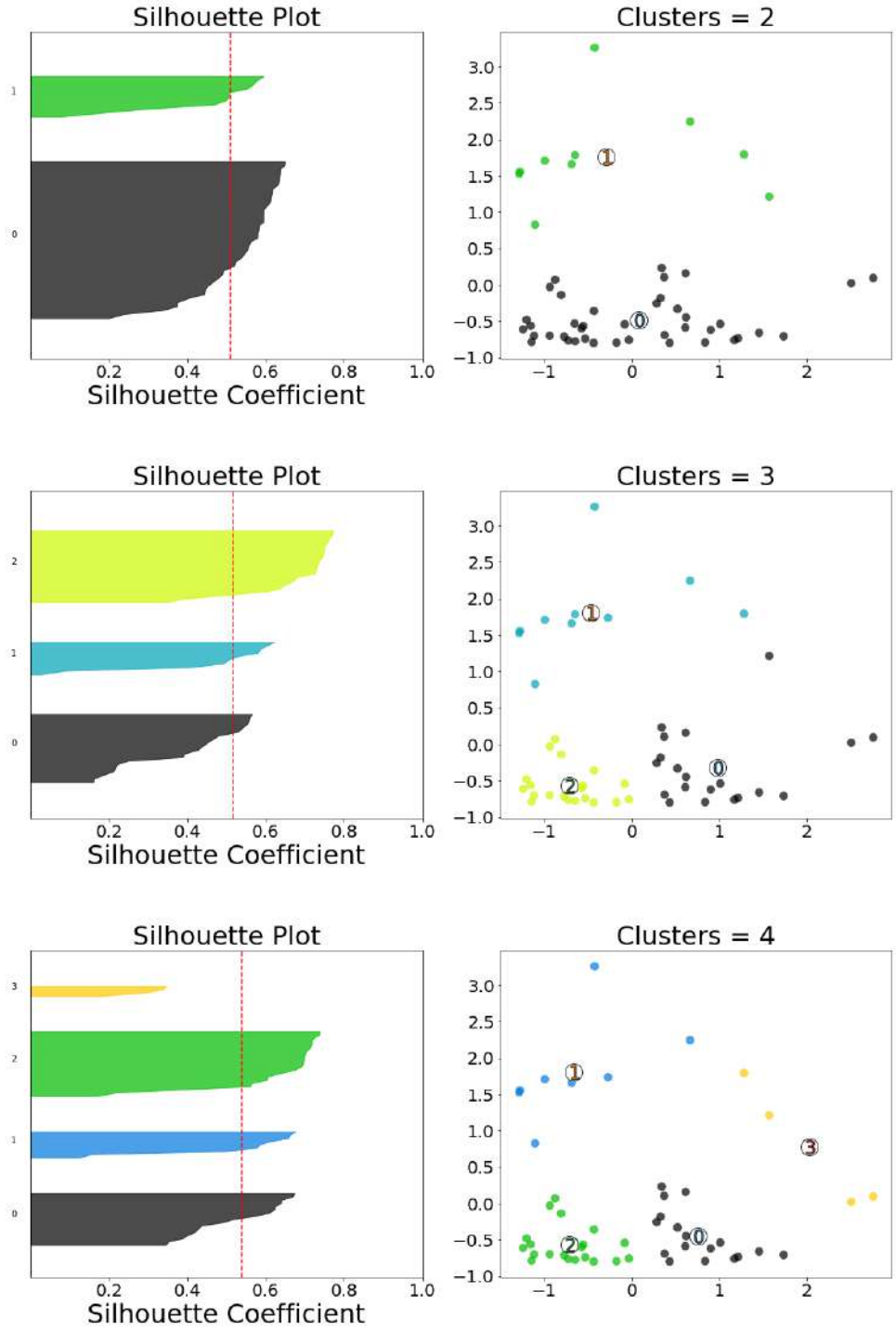


Figure 25: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 2, 3 and 4 clusters respectively.

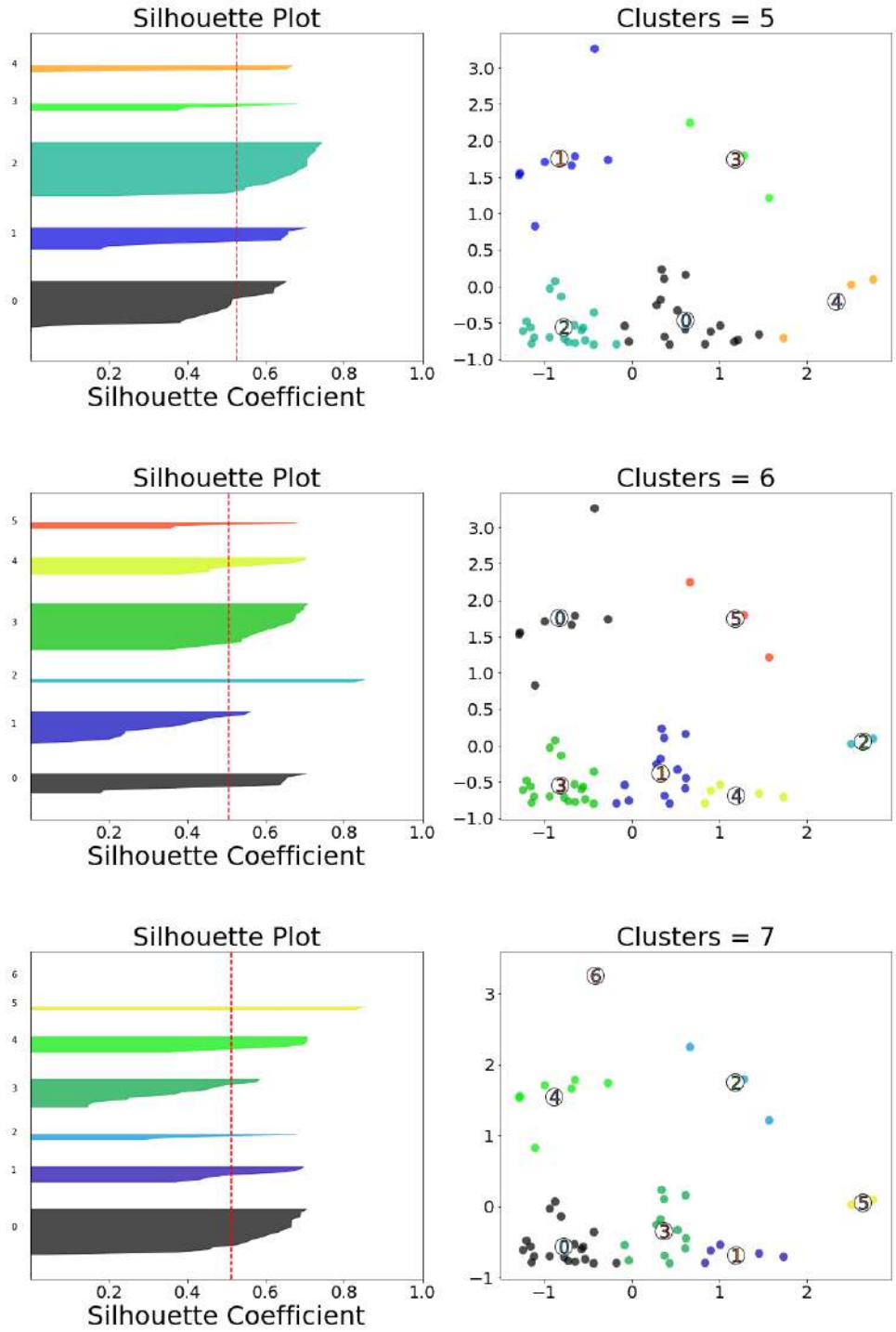


Figure 26: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 5, 6 and 7 clusters respectively.

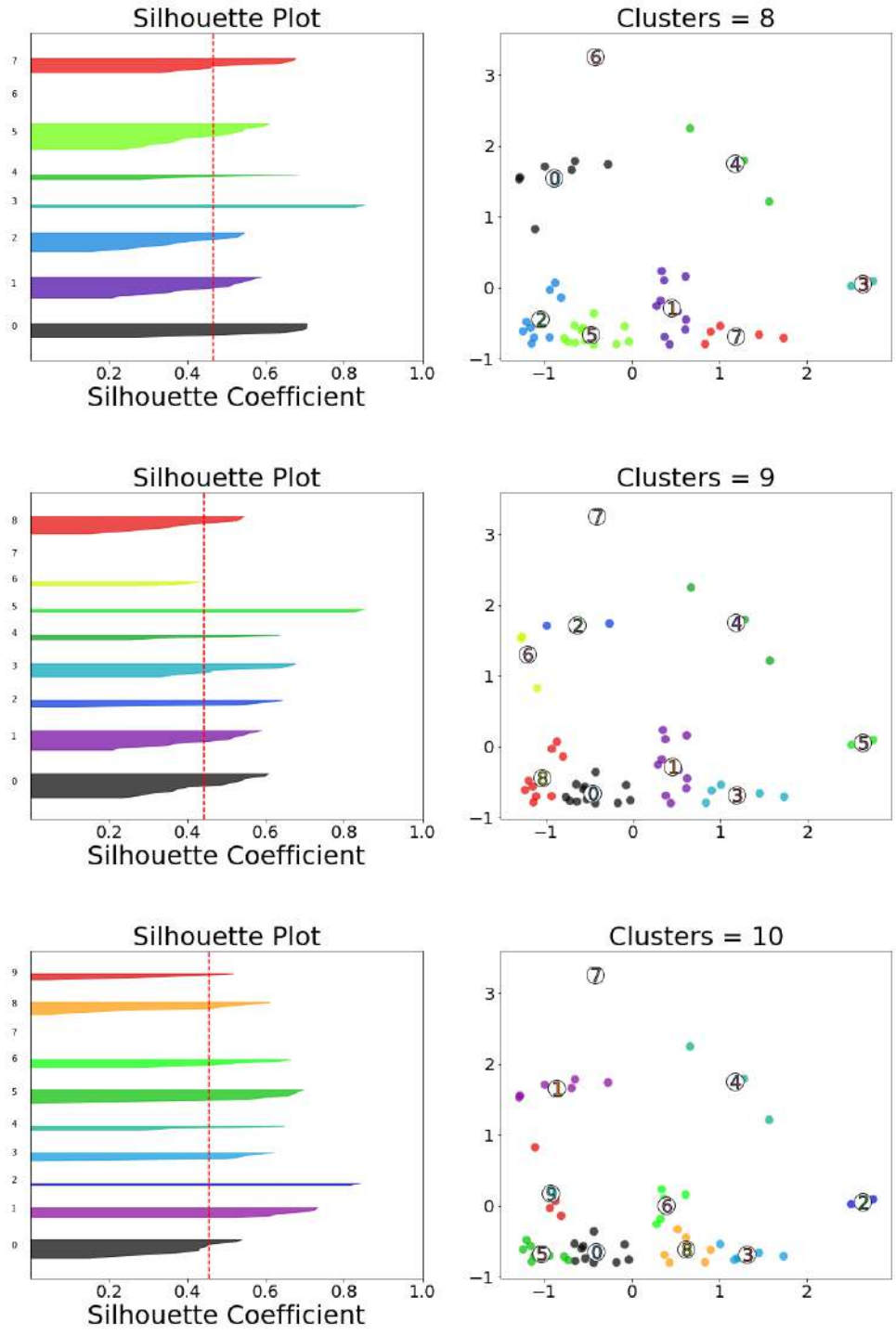


Figure 27: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 8, 9 and 10 clusters respectively.

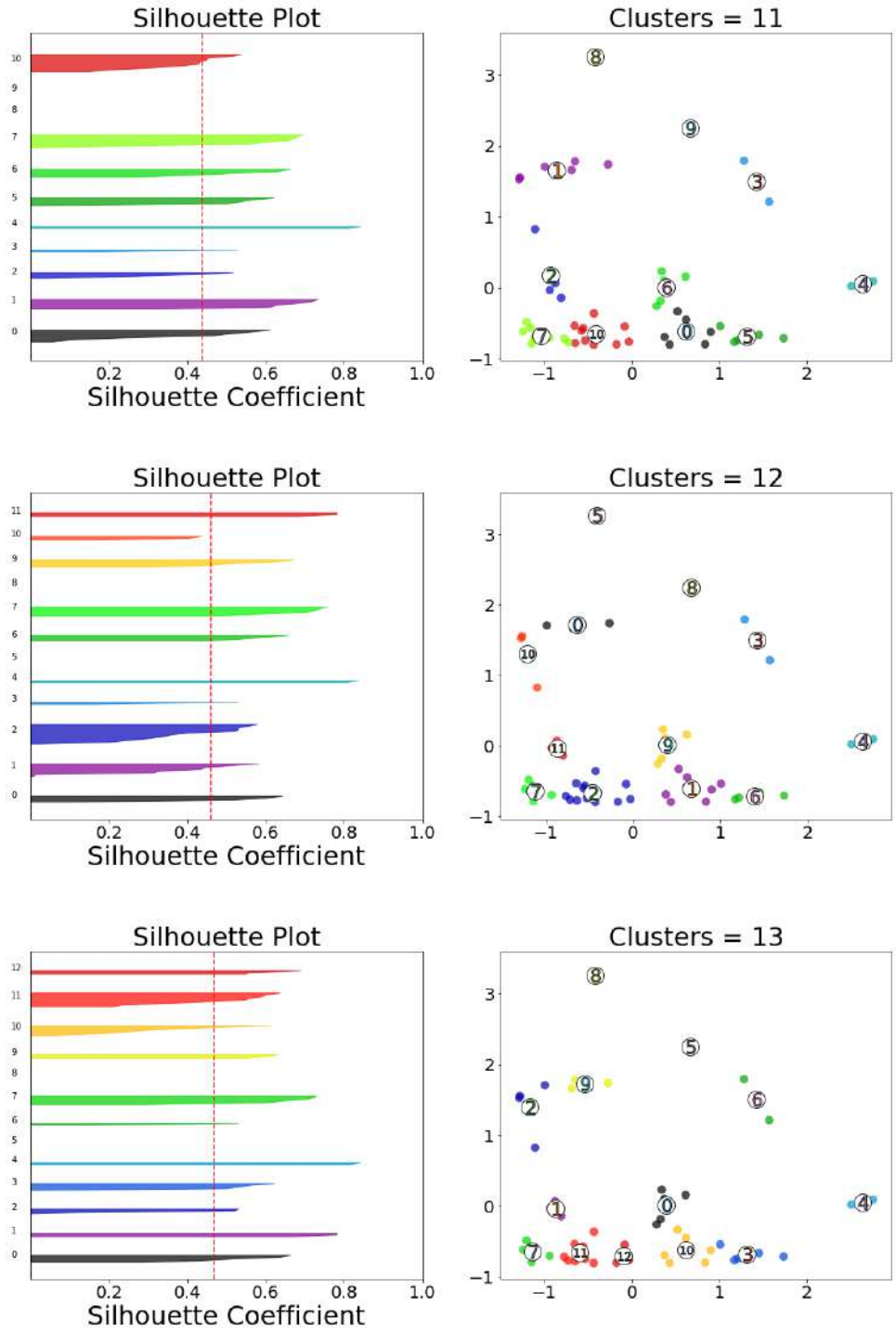


Figure 28: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 11, 12 and 13 clusters respectively.

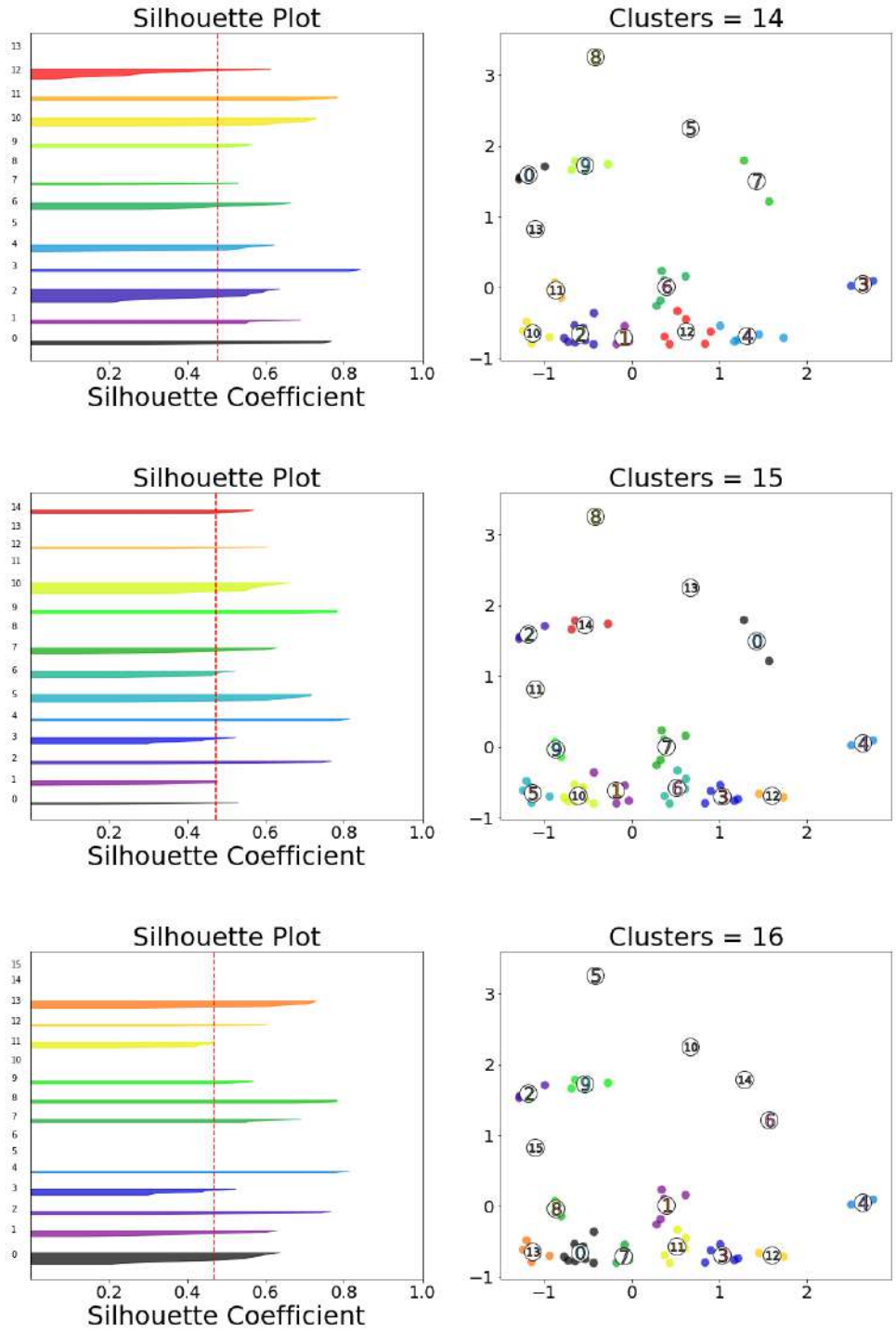


Figure 29: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 14, 15 and 16 clusters respectively.

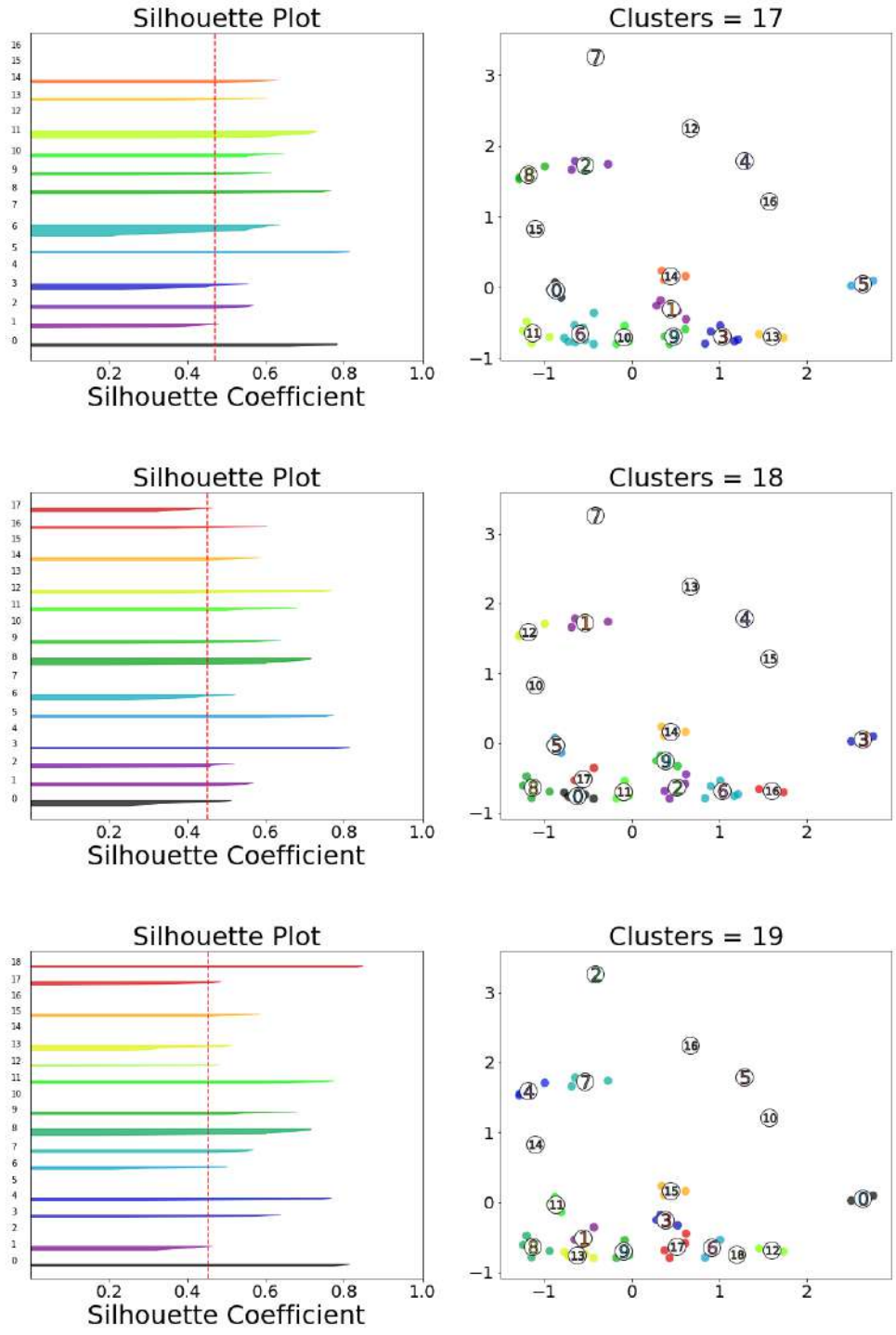


Figure 30: The above graphs depict the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plots on the left are visual representations of the Silhouette, while the plots on the left are the resulting clusters. These are for 17, 18 and 19 clusters respectively.

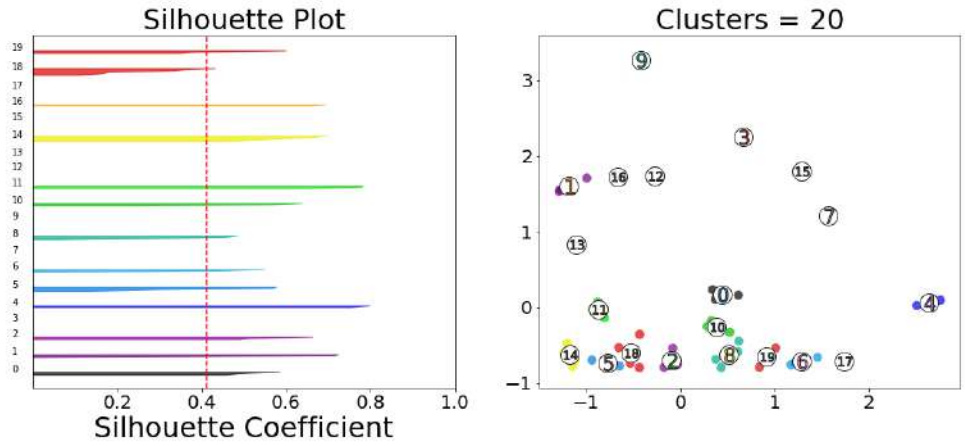


Figure 31: The above graph depicts the Silhouette Analysis for Quadrant 1 of the B12 dataset. The plot on the left is visual representation of the Silhouette, while the plot on the left is the resulting clusters. These is for 20 clusters.

Table 4: Average Silhouette Score calculated for each value of k in Quadrant 1 of the B12 dataset.

k	Average Silhouette Score
2	0.5109038582009556
3	0.5170351572649495
4	0.5382962735652664
5	0.5249613280237251
6	0.5068372097500163
7	0.5134739533064956
8	0.46645947363662293
9	0.4419197986151517
10	0.4549954674385096
11	0.4387255686767596
12	0.46048265483415707
13	0.46784784820145603
14	0.47818264811521133
15	0.47287916972947514
16	0.4697351482684045
17	0.4709027803697369
18	0.4522785881845947
19	0.4540656799487794
20	0.4108042824257553

When it comes to interpreting Silhouette plots and their corresponding scores, it is not as straightforward as looking for a minimum or maximum value. Instead, The average silhouette score for a given model needs to be analysed in conjunction with the silhouette plot along with the resulting clustering for that model. A silhouette score is calculated for each cluster in a model and then an average score is calculated by using all these values. The cohesion[35] of a clustering model is the feature that is evaluated during silhouette analysis. A desirable model is one that has relatively high cohesion, this would be indicated by a high average silhouette score. Using only this bit of information, the optimal k for the dataset B12 in quadrant 1 according to this method would be 4. Figure 25c depicts the silhouette analysis and corresponding k-means plot that was used to calculate this initial optimal k value.By examining the corresponding plot for k being equal to 4, it is relatively clear why this test has settled on such a k. There does indeed appear to be 4 broad clusters in this particular quadrant. These have been loosely segmented by the different clusters. While the result produced from this clustering may be worth evaluating in a biological sense, it should not necessarily be taken as the sole result from silhouette analysis.

This is because there are some limitations that are imposed on silhouette analysis by the very nature of the clusters that it ranks as optimal. An optimal model with well-formed clusters according to silhouette analysis is

[35]Recall that cohesion in cluster analysis indicates how close a point is to its own cluster as well as how far away it is from other clusters i.e. clusters that are dissimilar to the one that the point belongs to.

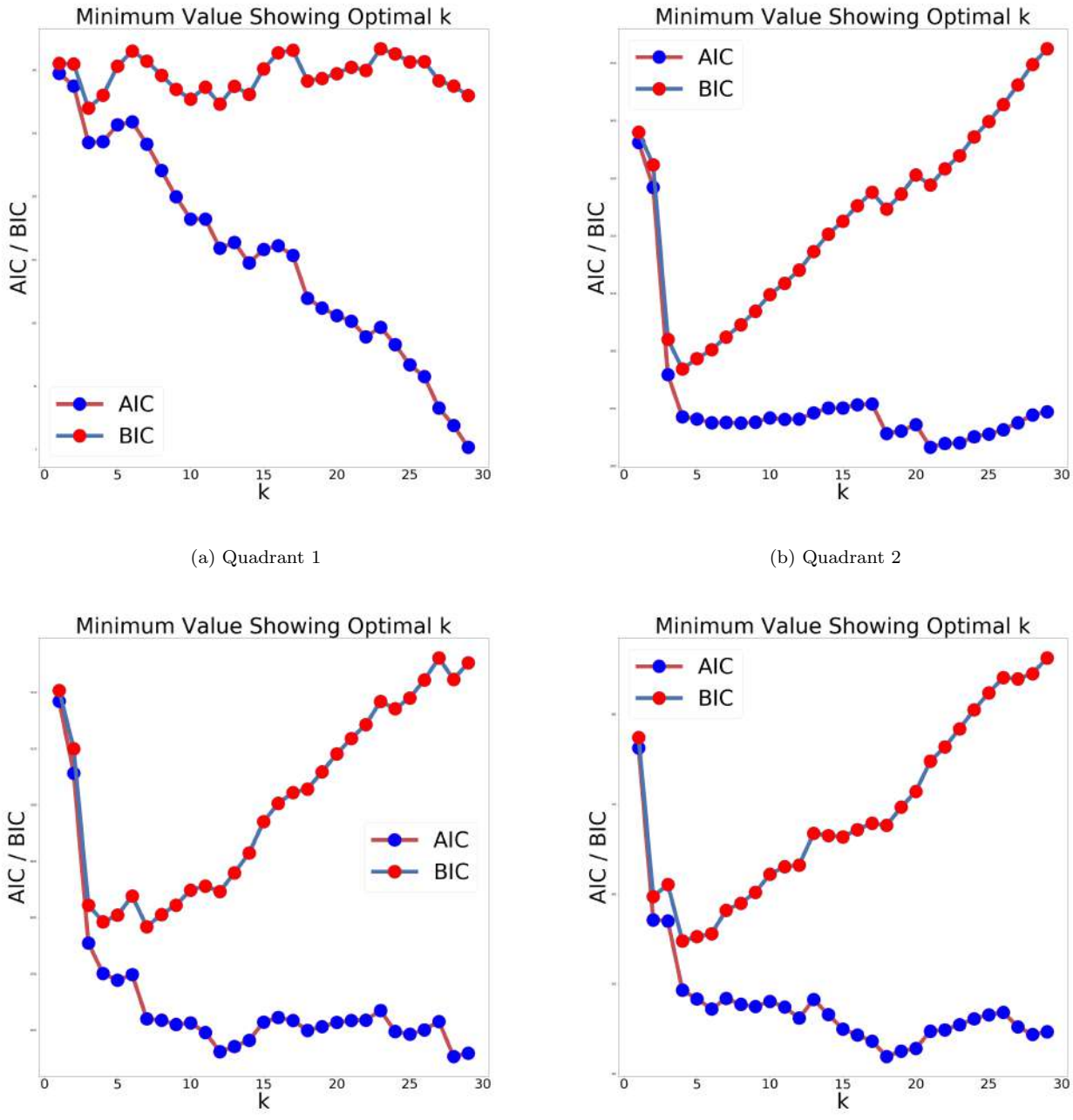
one that has:

- Silhouette plots for each cluster falling above the red line, which indicates the average silhouette score for this value of k. This feature confirms that each cluster has at least a relatively high level of cohesion when compared to the average silhouette score for the whole model.
- The size of each silhouette plot being relatively similar[36].

These points certainly provide a good guideline upon which to evaluate the prevailing clusters in a dataset. But they cannot be taken as absolute and they cannot exist in isolation. One reason for this being that the data being tested is *real-world* data. It does not necessarily have clusters that are the same size and there may be outliers present in the data, that could be interfering with the accuracy of the clustering method. If only the tenants of data science were to be observed then an argument could be made for the removal of these troublesome outliers[52]. But when it comes to data analysis, one need always bear in mind the nature of the data that is being dealt with. In this case, the data is indicative of gene expression values in the cell of the Cassava plant. Points that may appear to be outliers in the distribution are not simply errors produced during the data collection process. They simply indicate a rather large spread in gene expression data. These points cannot be removed as they too provide information as to the mecha-nations of the organism. With this in mind, one could look for models that deal with these outliers not by forcing them into a clusters in order to achieve evenly sized clusters, but rather allow these points to exist in their own cluster. Further more, this method of clustering may be useful in the later stages of gene analysis. If new data points were to be observed that share the expression profile of these genes that exist in their own clusters, then there is already defined groups/clusters for these potential points/genes to be compared with. If this new logic is applied while maintaining that a high average silhouette score is desirable for optimal k selection, then 7 or 14 clusters should also be considered as candidates. These models can be found in Figure 26c and 29a respectively.

[36]This is generally accepted as a feature of a *good* clustering model as clustering methods tend to assume that the prevailing data has a distribution that will allow for evenly sized and spread clusters [51].

4.2.3 AIC/BIC for B12 Dataset



(a) Quadrant 1

(b) Quadrant 2

(c) Quadrant 3

(d) Quadrant 4

Figure 32: Plots depicting the AIC and BIC Scores for the data in the B12 dataset. The optimal values for k can be found at the local minima in each Quadrant.

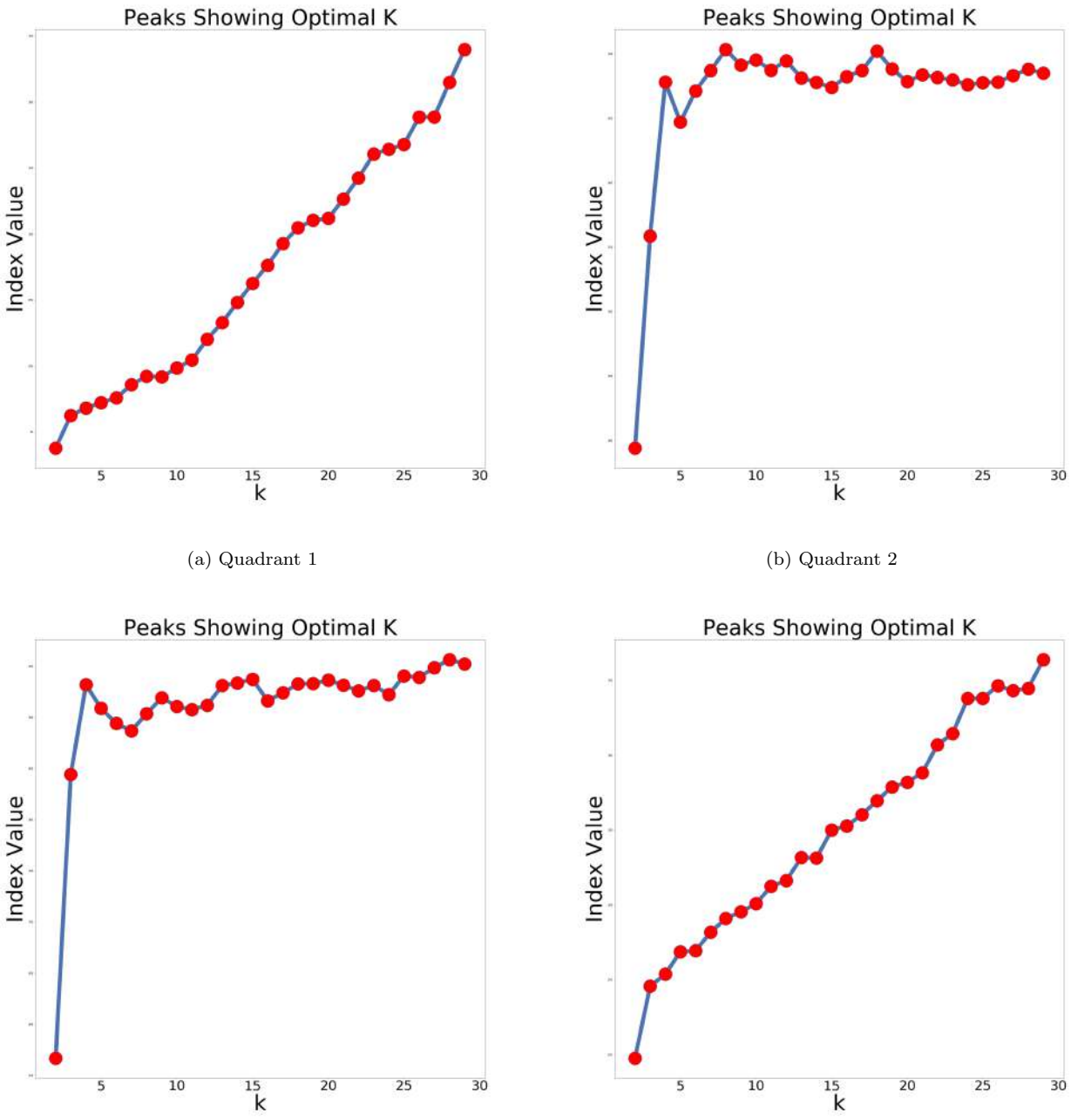
Recall that the Akaike and Bayesian Information Criterion both seek out models that are able to explain the data without losing too much information in the process. This is done using the equations derived in 3.11.7 and 3.11.8. The former of these two has a penalty factor of $2k$ while the latter performed a more stringent penalisation by multiplying by the parameter k.

It is therefore expected that when graphing the AIC and BIC values for a dataset one will find that the AIC value for a model will continue to decrease[37] without being hampered too much by the steady increase of k. BIC values will increase once it seems that the model is becoming too complex for the data that it is representing. This is an unfavourable outcome as unnecessarily large amounts of information are lost in overly complex clustering models [26].

This information on the nature of Information Criterion testing means that an optimal model for a dataset should exist at the minimum values of an AIC and BIC mapping. While a global minimum may theoretically provide the best possible model, this is not always the case in practice and therefore global and local minimums shall be considered. In figure (32a) these minimums would be a k value of 3, 12 or 14. These represent points in the graph which are characterised by a sharp dip preceding the point along with an increase directly after the point. The significance of this being that the value for k at a local minima is a better fit than the model for k-1 and k+1.

[37] A low AIC or BIC value indicates a model that has lost little information during its creation.

4.2.4 Calinskv-Harabasz Score for B12 Dataset



(a) Quadrant 1

(b) Quadrant 2

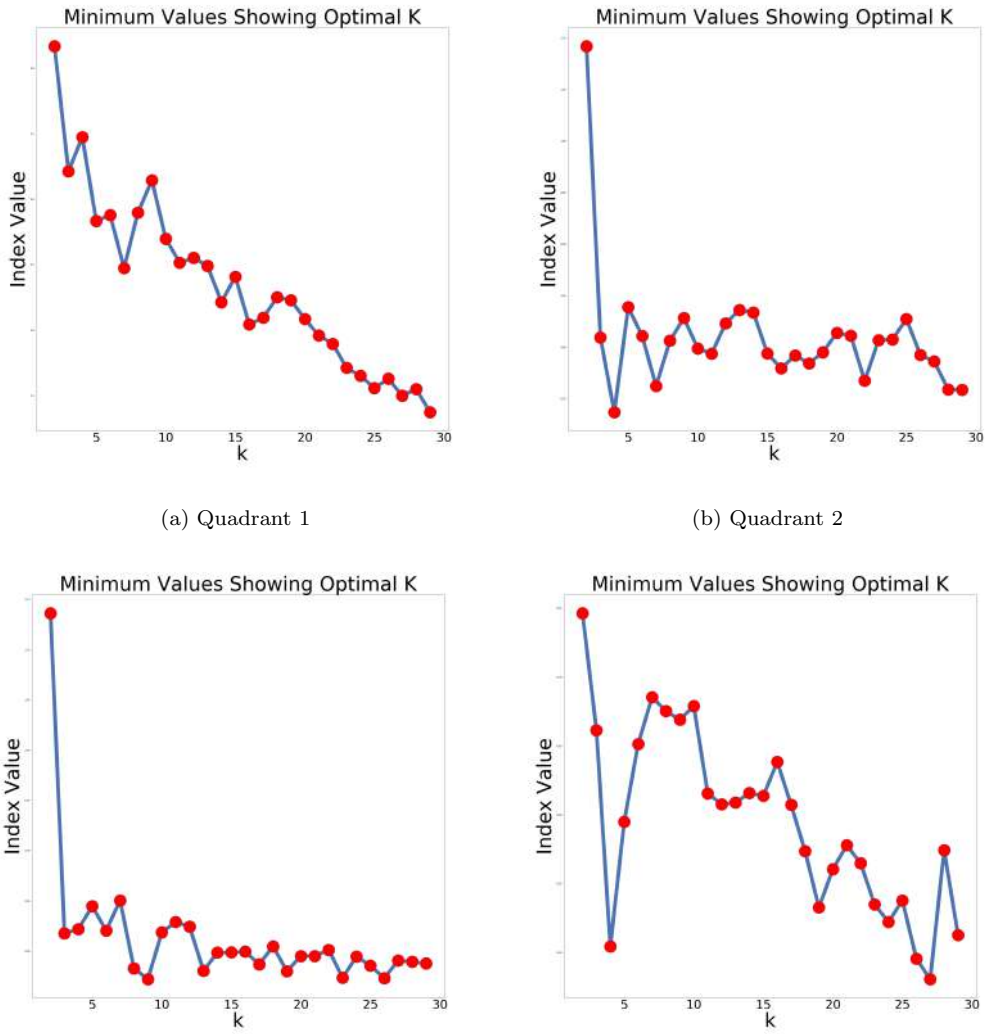
(c) Quadrant 3

(d) Quadrant 4

Figure 33: Plots depicting the Calinsky-Harabasz Score for the data in the B12 dataset. The optimal values for k can be found at the local minima in each Quadrant.

The Calinski-Harabasz Index is primarily concerned with identifying models that have dense and well-separated clusters. These models would be identifiable by a high CH value, which would exist at the peaks of the plots above. Figure (33b) suggests that an optimal k values for data in quadrant 2 of the B12 dataset would be either 8 or 18.

4.2.5 Davies-Bouldin Score for B12 Dataset



(a) Quadrant 1

(b) Quadrant 2

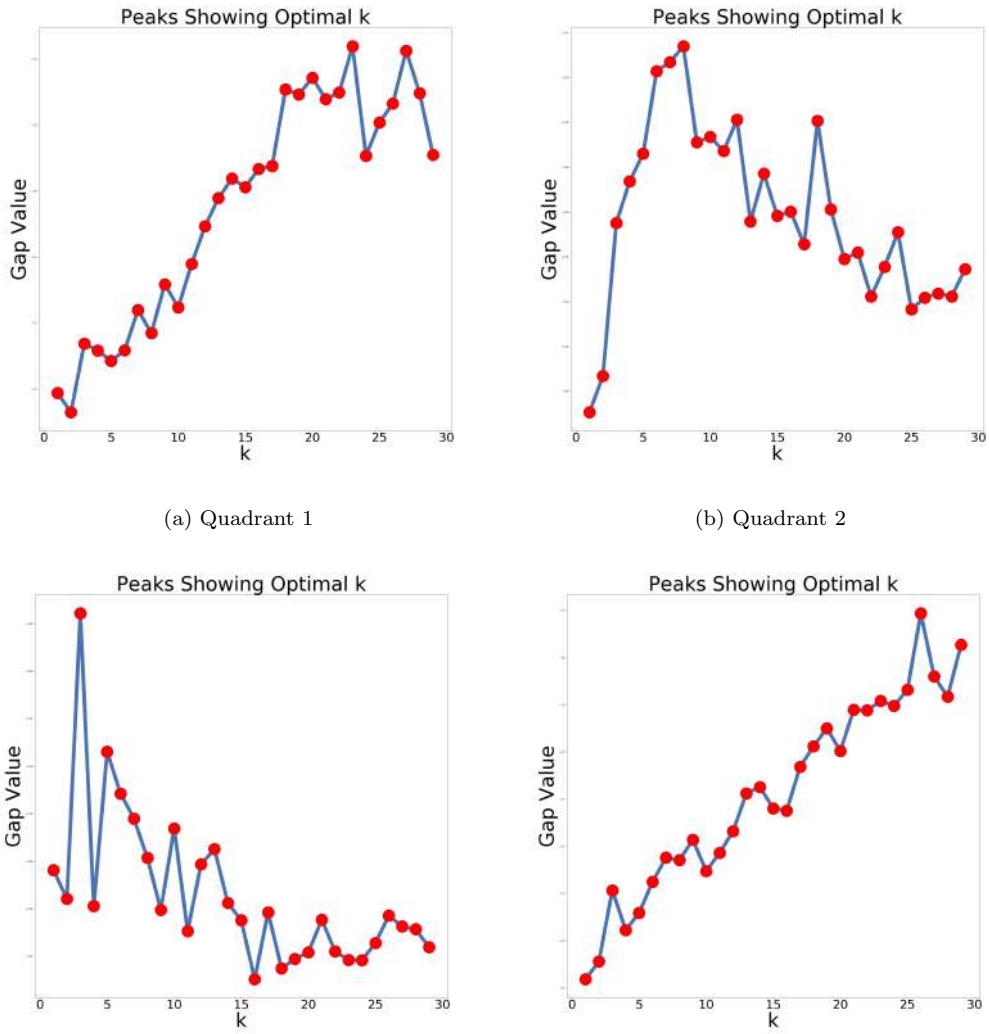
(c) Quadrant 3

(d) Quadrant 4

Figure 34: Plots depicting the Davies-Bouldin Score for the data in the B12 dataset. The optimal values for k can be found at the local minima in each Quadrant.

The Davies-Bouldin Index considers the inter and intra cluster variation when examining goodness of fit. Relevant calculations would then seek out a model that has a high inter cluster variation coupled with a low intra-cluster variation. These together would produce a low DB value in an optimal model.
In figure(34a) a k of 7, 14 or 16 produces relatively low DB values. With this example a distinct limitation of these indices is brought to light. This being that if the underlying data is not naturally well segmented then an index like the Davies-Bouldin index will continue to decrease as the cluster number increases. The reason for this being that considerably smaller clusters are more likely to display low intra-cluster variation, making these the more favourable models in the scope of the index. Therefore these points were selected as they are distinct local minima. This relation being identified by a sharp decrease in the index directly before the point in question along with a sharp increase in the score directly after.

4.2.6 Gap Statistic for B12 Dataset



(a) Quadrant 1

(b) Quadrant 2

(c) Quadrant 3

(d) Quadrant 4

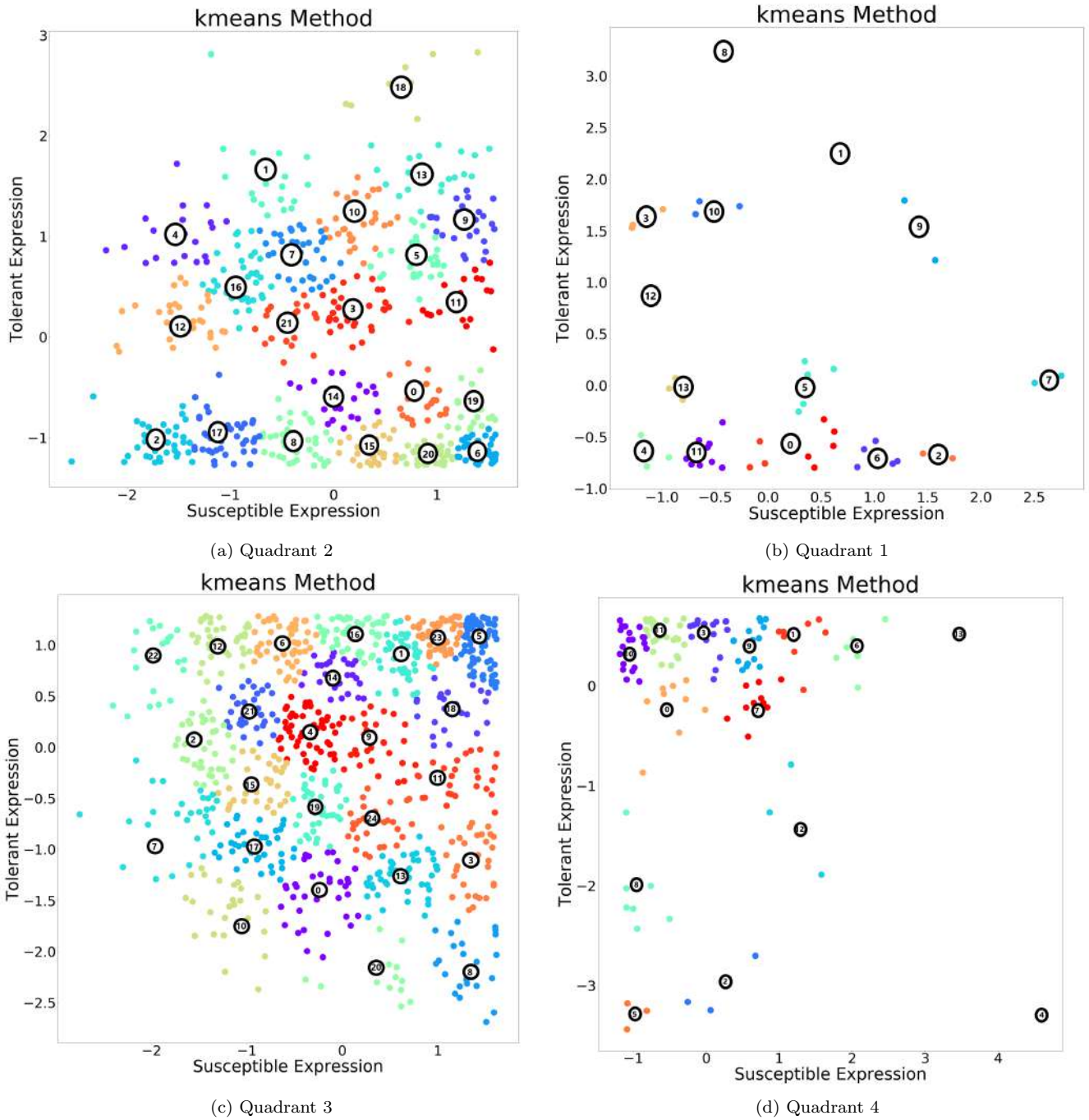
Figure 35: Plots depicting the Gap Statistic for the data in the B12 dataset. The optimal values for k can be found at the local minima in each Quadrant.

The Gap Statistic sought to formalise the way in which an optimal k was selected. R. Tibshirani *et al.* proposed seeking out the value for k which maximises Equation 3.11.13, [31]. This is interpreted as a peak in the above graphs. The authors did however concede that a single peak is not always present in *real-world* data, in which case they suggested seeking out the value for k where the increase in the gap statistic starts to plateau.

The data that has been used in this project produces Gap statistic graphs that are more jagged than ones which produce a single peak or a value which signifies the onset of a plateau. Therefore the local maximums in each of the graphs were considered in conjunction with the results from the aforementioned statistical tests. In Figure 35a the possible peaks that were available for k was 7, 14 or 18.

4.3 k-means Method

In analysing the results produced by the k-means algorithm that was implemented with the aid of the *scikit-learn* package in *Python 3.7.4* [35], the concept of a centroid needs to be readdressed. The centroids defined by the algorithm are simply points, either ones that are included in the dataset or not, that are right in the centre of the resulting clusters. This definition of a centroid gives some insight into the nature of the clusters that the k-means algorithm aims to produce. These clusters, being ones that are of a relatively even size and spherical [53].



(a) Quadrant 2

(b) Quadrant 1

(c) Quadrant 3

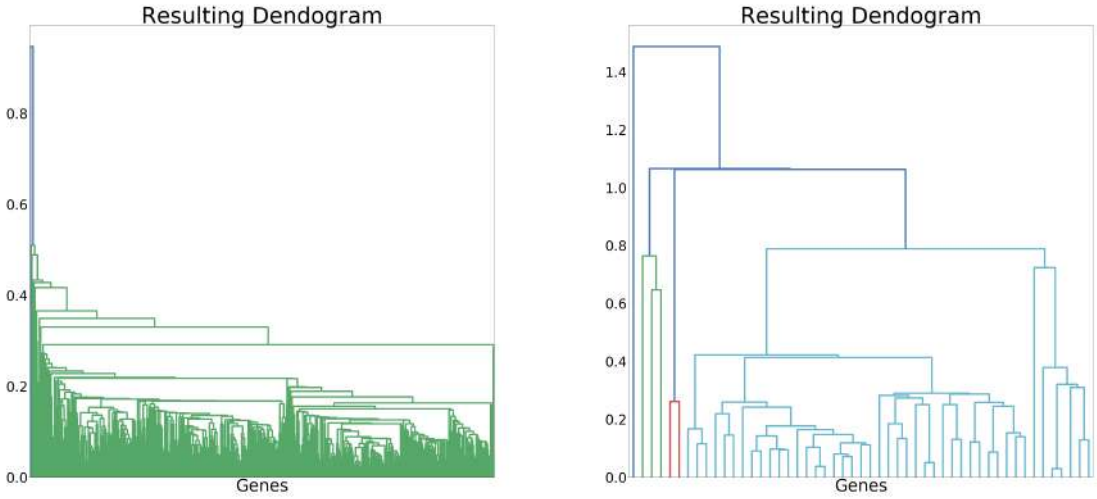
(d) Quadrant 4

Figure 36: Plots depicting the results when applying the k-means clustering algorithm to the B12 dataset.

This can clearly be seen when looking at the clusters produced when applying the k-means algorithm to the data in quadrant 1 of the B12 dataset.
Figure (36a), clearly depicts the spherical nature of the resulting clusters. The centroid of each cluster is depicted by the white circles. By taking these points as the centres of the clusters, it is easy to trace a circle which encompasses the points that belong to that cluster.
While the clusters are relatively spherical in shape, they are not similar in size[38]

4.4 Hierarchical Clustering - Single Linkage

In single linkage clustering the distance measure is calculated by using the two **most** similar points in the respective clusters. This will result in a lack of compactness of the clusters as a chaining effect should be observed, producing large clusters that encompass seemingly disparate points.



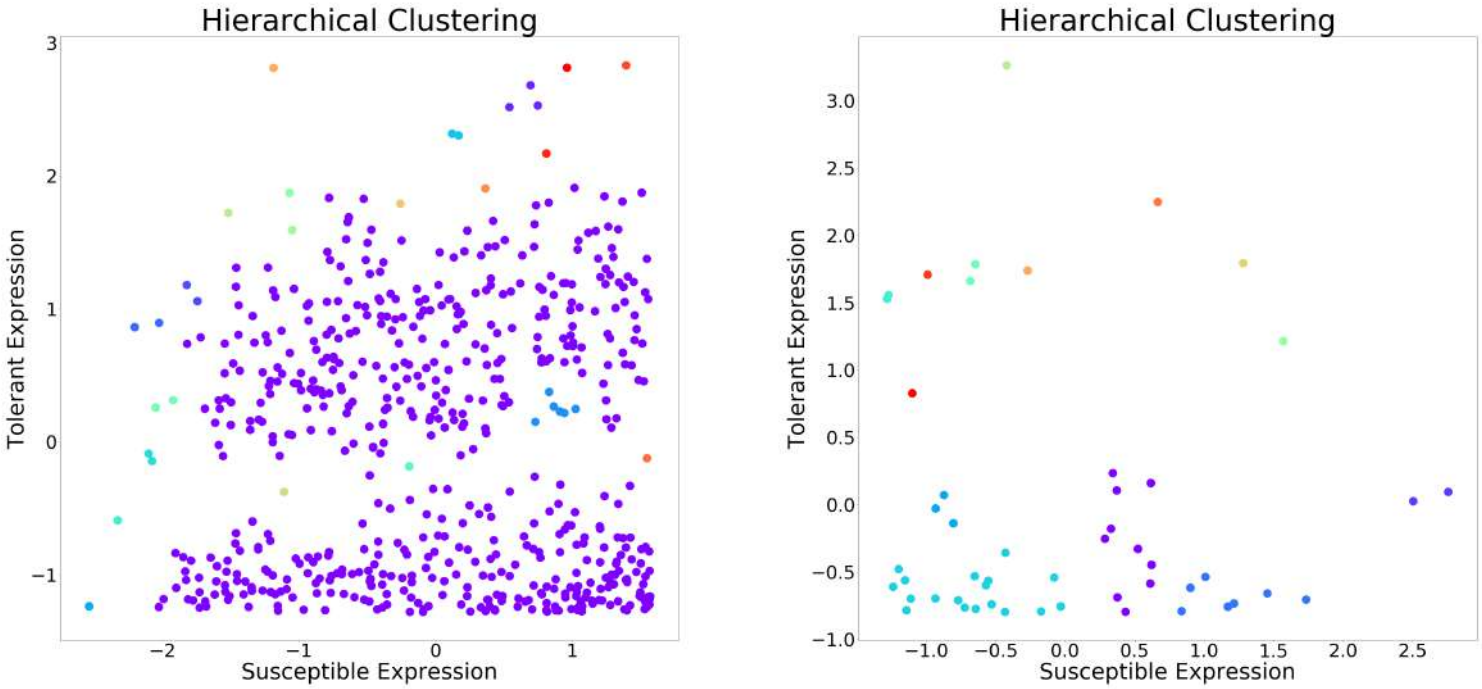
(a) Quadrant 2- Dendrogram

(b) Quadrant 1- Dendrogram

Figure 37: Plots depicting the results from applying Hierarchical clustering using single linkage to Quadrant 1 and 2 of the B12 dataset. The resulting dendrogram is displayed for Quadrant 2 in Figure 37a and Quadrant 1 in Figure 37b. Genes are represented on the x-axis, with the distance between the points plotted on the y-axis.

The chaining effect mentioned above can be observed best in Figure 38a and 37a, where there appears to be one vast cluster that encompasses a large range of points. This type of linkage appears to work relatively well with small sets of data as is evident in Figure 38b. But it does not yield useful results when the dataset size increases. The dendrogram produced here is also mostly unintelligible.
The selected value for k in Quadrant 2 of dataset B12 was 22. To find this value in Figure 37a, one would need to drag a horizontal line from the top of the graph down until there are 22 vertical lines at a right angle to it. This would only be possible if the graph were to be magnified a substantial amount. Furthermore, even if the points and clusters were at a more visually accessible size there would still exist the one large cluster that dominates the results space.

[38]The size of a cluster does not refer to the amount of points in the cluster. Rather it speaks of the area of the graph that a particular cluster occupies.



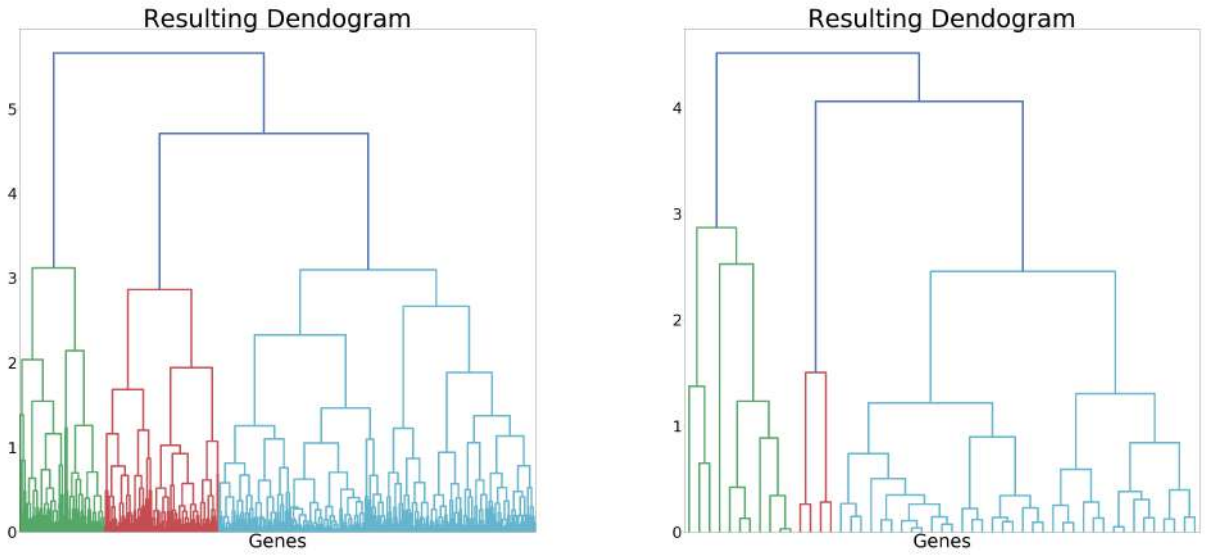
(a) Quadrant 2

(b) Quadrant 1

Figure 38: Plots illustrating the clusters produced by applying Hierarchical clustering with single linkage to Quadrant 1 and 2 of the B12 dataset. The value for k used can be found in Table 3.

4.5 Hierarchical Clustering - Complete Linkage

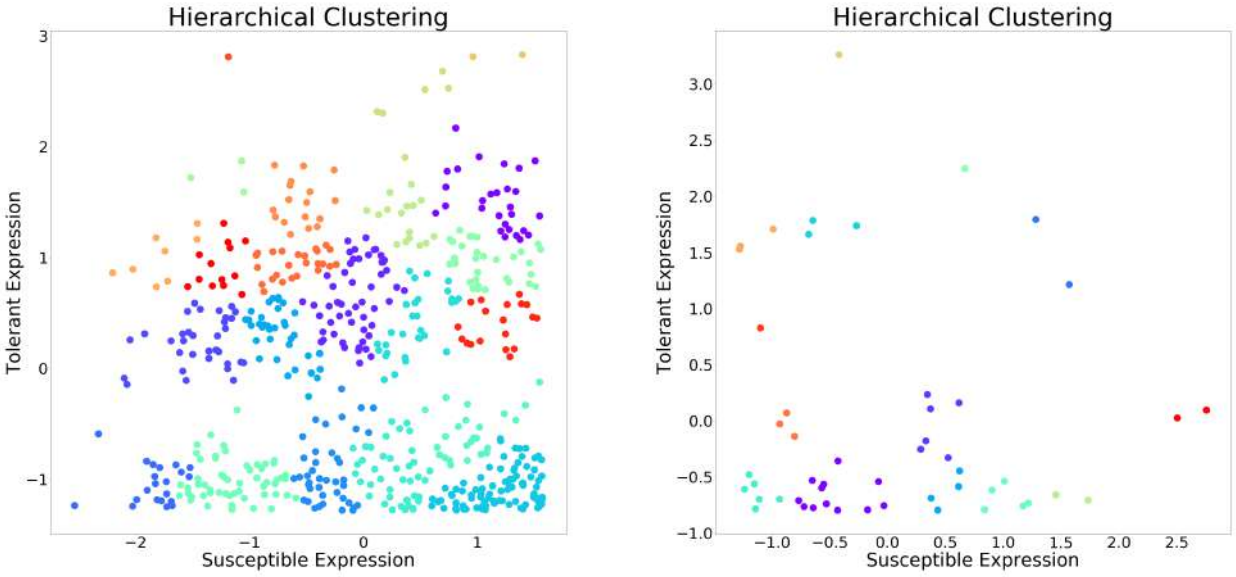
When computing distance with complete linkage, the **least** similar points in two clusters are used. This corrects the chaining issue that is seen in single linkage clustering and produces more compact clusters.
If Figure 37a and Figure 39a are to be compared, then it becomes quite clear that there is no longer the issue with chaining. The resulting dendrogram for complete linkage is also far more legible than its single linkage counterpart. But, compactness is not the only sought after feature of a suitable clustering model. For clusters to be well-formed, they need to be compact as well as relatively far apart. In a small dataset this is an easier task as is seen with Figure 40b, where the data is nicely dispersed allowing for the spacial separation of differing clusters. In contrast, Figure 40a has more data which has naturally created dense regions of points. Here the lack of separation of clusters is evident as there are overlap regions where it is unclear which cluster a set of points rightly belongs. This produces an effective visualisation of the issue that arises with complete linkage. This issue being that complete linkage suffers from crowding.



(a) Quadrant 2- Dendrogram

(b) Quadrant 1- Dendrogram

Figure 39: Plots depicting the results from applying Hierarchical clustering using complete linkage to Quadrant 1 and 2 of the B12 dataset. The resulting dendrogram is displayed for Quadrant 2 in Figure 39a and Quadrant 1 in Figure 39b. Genes are represented on the x-axis, with the distance between the points plotted on the y-axis.



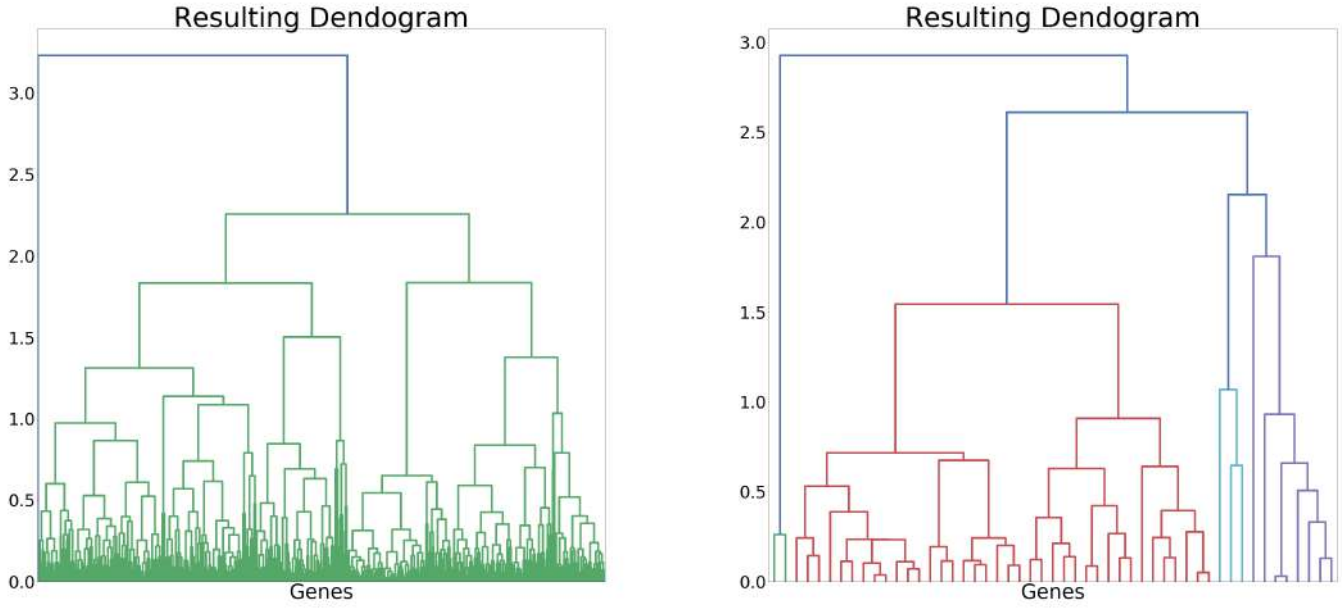
(a) Quadrant 2

(b) Quadrant 1

Figure 40: Plots illustrating the clusters produced by applying Hierarchical clustering with complete linkage to Quadrant 1 and 2 of the B12 dataset. The value for k used can be found in Table 3.

4.6 Hierarchical Clustering - Average Linkage

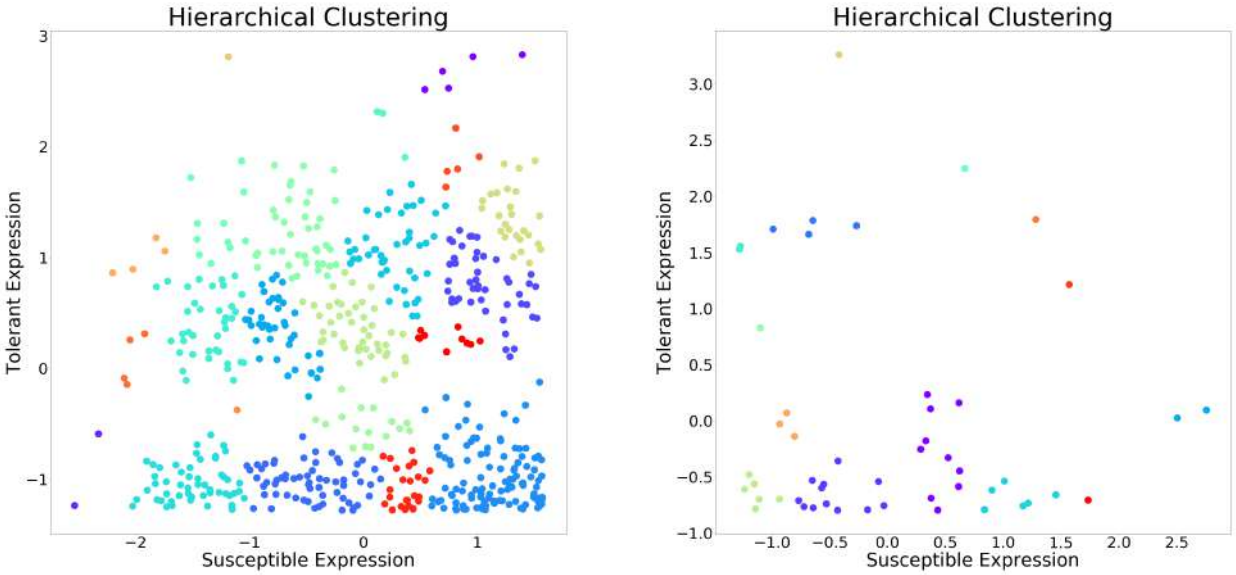
When computing the distance measure with average linkage clustering, more than a single set of points is used as with the previous distance measures that have been presented. Instead, the average distance is calculated by taking the distance between each point in one cluster with each point in its neighbouring cluster. This distance measure was proposed in an attempt to correct the issue of chaining and crowding which is present in single and complete linkage respectively. The idea being that a balance should be found resulting in clusters that are relatively compact while being relatively far enough apart.
If the dendrogram in Figure 41a is compared with the previous dendrogram as produced by single linkage Figure 37a, it is clear that chaining is not an issue with Average linkage.
The second problem that this type of linkage aims to address is crowding. As mentioned with the results in Complete linkage, crowding does not present a significant problem when the data being clustered is relatively small, as is the case in Quadrant 1(Figure 40b). Therefore Quadrant 2(Figure 42a) needs to be examined to evaluate whether the crowding effect has diminished between complete and average linkage.
Upon initial examination Figure 40a and Figure 42a do not appear to differ substantially. There are however some subtle differences that indicate an improvement in terms of spacial separation between clusters. This can be seen in the upper right corner and lower left corner of Figure 42a. In these areas of the graph, average linkage has separated disparate points into different clusters producing both relative compactness and spacial separation. This effect is less visible in areas of high density, such as in the centre of the graph, but the clusters in Figure 42a do appear to be more regular in shape which lends itself to spacial separation.



(a) Quadrant 2 - Dendrogram

(b) Quadrant 1 - Dendrogram

Figure 41: Plots depicting the results from applying Hierarchical clustering using average linkage to Quadrant 1 and 2 of the B12 dataset. The resulting dendrogram is displayed for Quadrant 2 in Figure 41a and Quadrant 1 in Figure 41b. Genes are represented on the x-axis, with the distance between the points plotted on the y-axis.



(a) Quadrant 2

(b) Quadrant 1

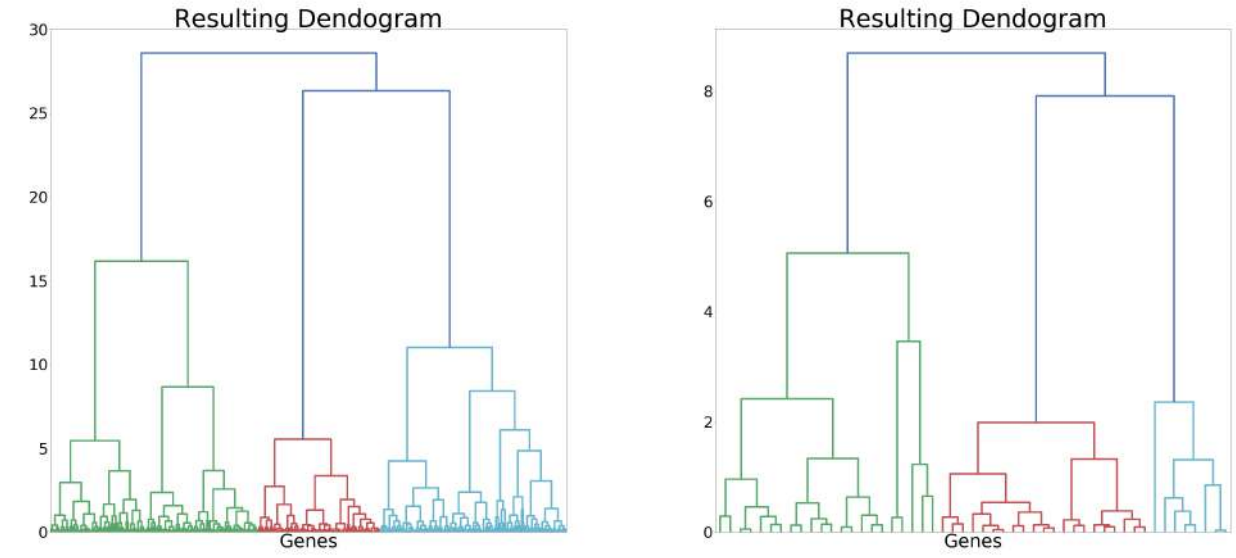
Figure 42: Plots illustrating the clusters produced by applying Hierarchical clustering with average linkage to Quadrant 1 and 2 of the B12 dataset. The value for k used can be found in Table 3.

4.7 Hierarchical Clustering - Ward Linkage

When clustering is governed by Wards minimum variance the spatial distance between clusters is less important than the potential increase in the sum of the squares that would be produced by the merging of the clusters in question [43]. At the imitation of Hierarchical clustering the sum of squares is equal to zero. This value increases as clusters are merged. Wards minimum variance seeks to merge clusters only if the cost of this merge does not unduly increase the sum of squares.

The result of this internal logic is that Ward Linkage will tend to produce clusters that are of more of less equal size. This effect may not be evident when examining Figure(ward quad 2) in isolation. But if Figure 44a is compared to Figure 40a and Figure 42a, then it is clear that the clusters produced here are closer to being equal in size than the previous results.

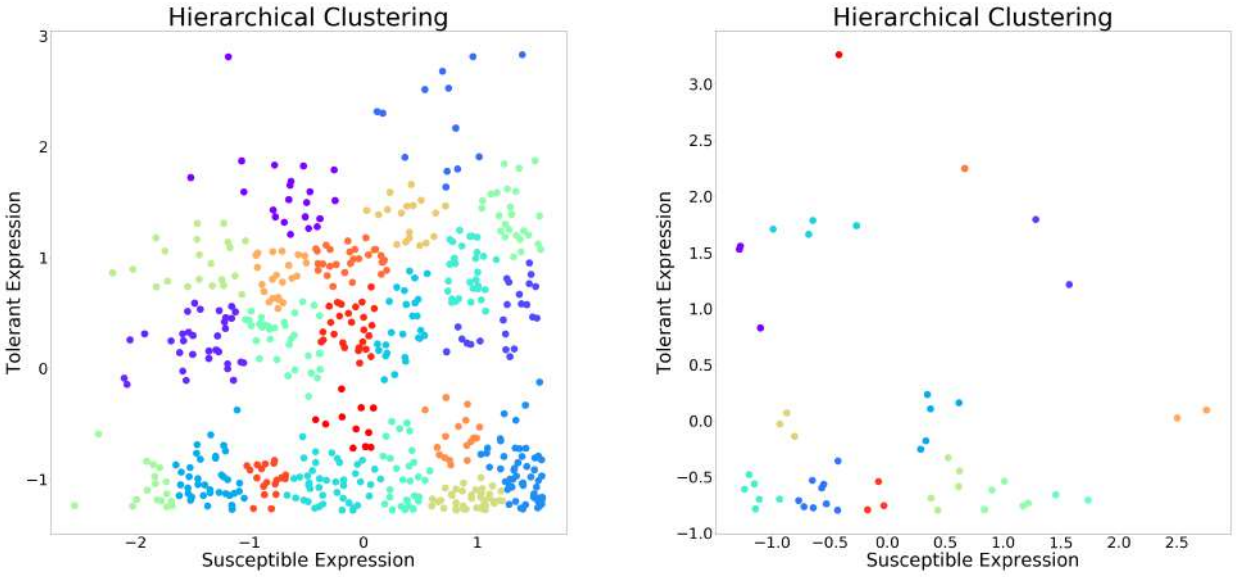
This would be desirable if it was known that the natural groupings within the data tend to be equal in size, but if there are outliers in the data then Wards Minimum variance would rather merge these with other smaller clusters than leave them in their own grouping. This effect is visible at the edges of the graph where potentials outliers have been included in clusters that appear to be relatively far from the point in question.



(a) Quadrant 2- Dendrogram

(b) Quadrant 1- Dendrogram

Figure 43: Plots depicting the results from applying Hierarchical clustering using Ward's linkage to Quadrant 1 and 2 of the B12 dataset. The resulting dendrogram is displayed for Quadrant 2 in Figure 43a and Quadrant 1 in Figure 43b. Genes are represented on the x-axis, with the distance between the points plotted on the y-axis.



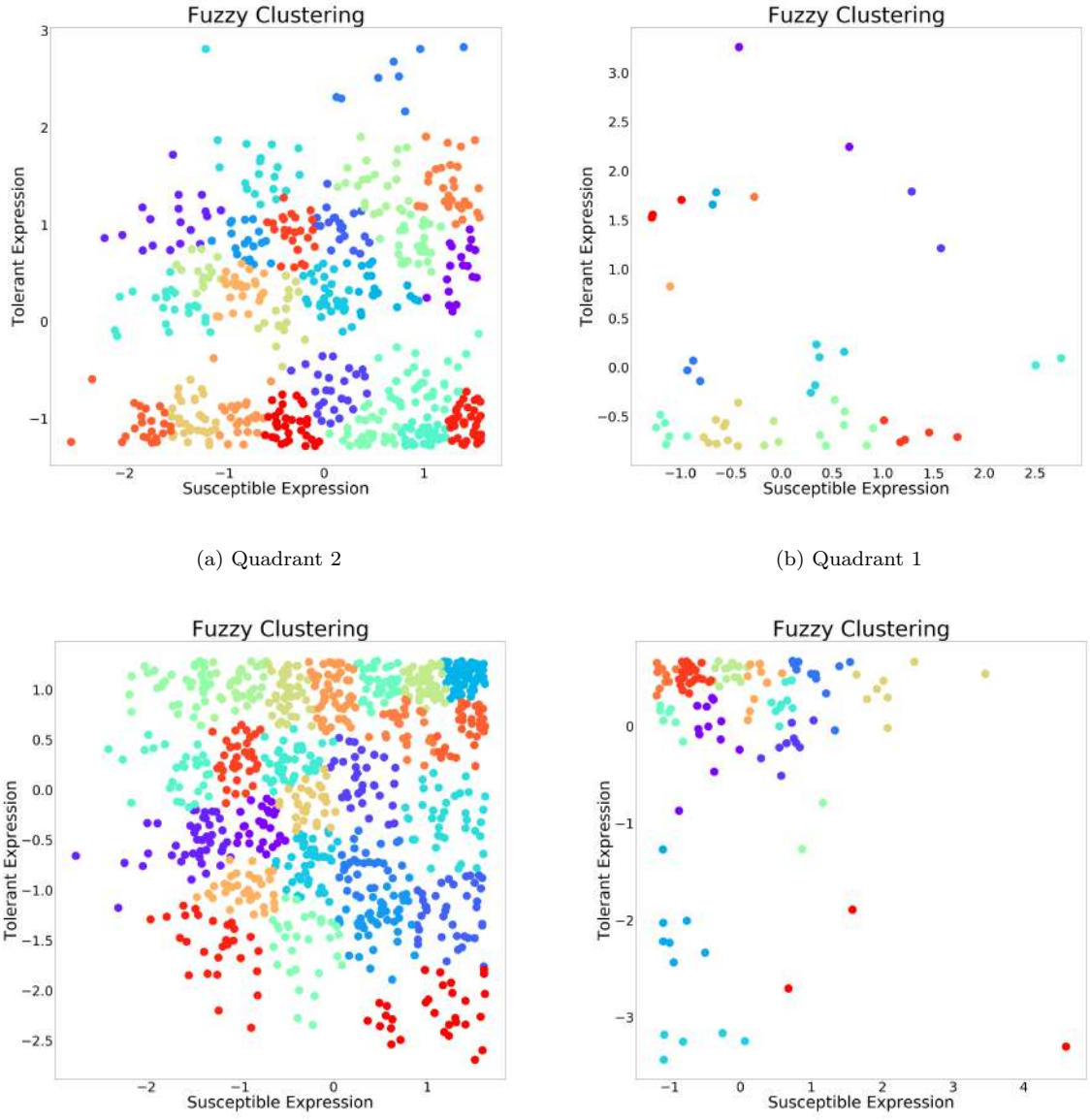
(a) Quadrant 2

(b) Quadrant 1

Figure 44: Plots illustrating the clusters produced by applying Hierarchical clustering with Ward's linkage to Quadrant 1 and 2 of the B12 dataset. The value for k used can be found in Table 3.

4.8 Fuzzy Clustering

Unlike the results that have been presented thus far, Fuzzy clustering does not seek to perform hard clustering on the given data. Instead points can belong to more than one cluster which is achieved by assigning a point a membership value which relates to how much it can belong to each of the potential clusters.
In practice this is not graphically represented by points having a colour scale which would describe the varying membership that has been assigned to each cluster. Instead a *fuzzy* classification of data would allow points on the boundary of two clusters to be assigned to the cluster with which it has the highest membership value, even if this assignment breaks the spherical shape of a cluster. The clusters produced therefore being non-spherical clusters with *fuzzy* edges. This feature is desirable when there are categorical or features overlap in the underlying data.



(a) Quadrant 2

(b) Quadrant 1

(c) Quadrant 3

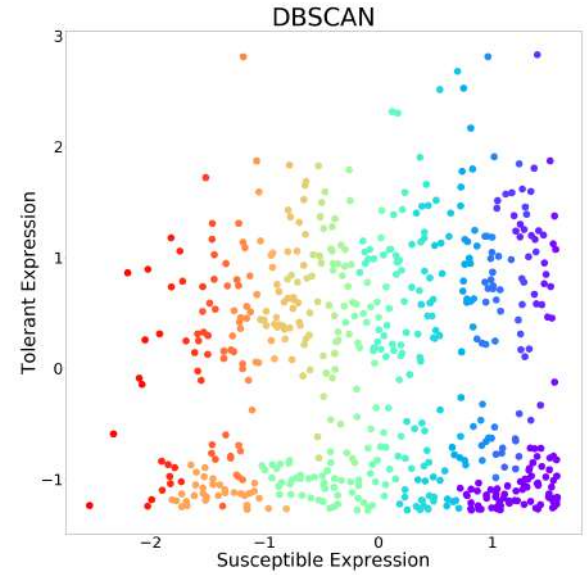
(d) Quadrant 4

Figure 45: Plots depicting the results when Fuzzy Clustering is applied to the B12 dataset.

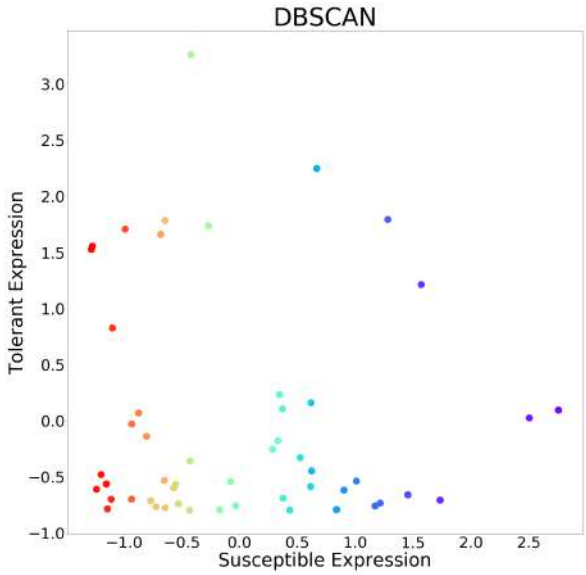
The effect of this *fuzzy* clustering is best seen in areas of high density such as the centre of the graph in Figure 45a. The clusters that have been produced here merge slightly with each other at the edges, clearly showing how a point could potentially belong to either of the clusters in its vicinity.

4.9 Density Based Clustering

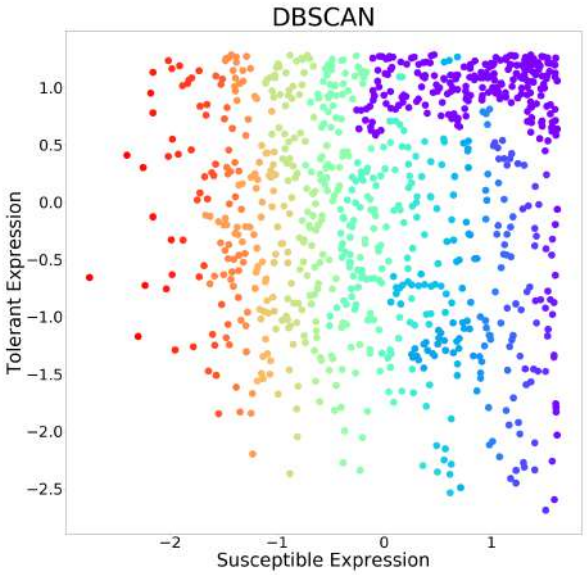
Density Based Clustering is achieved by making a distinction between core points and noise points. Core points exist in areas of high density and are thus clustered together. These areas/clusters are then separated by areas of low density that are populated by noise points. The logic implemented here is sound if there is some certainty in the existence and frequency of the noise within the data.
While the data that is being tested here may have a rather large dispersion, there are not distinct areas of high density that would indicate a formal grouping. Instead the data is more dispersed throughout the graph with areas of relatively higher and lower densities. There is no clear division between points that could be considered core points and noise. When the DBSCAN algorithm is applied here the result is large clusters that span substantial portions of the active area as well as many small clusters which the algorithm considers to be noise. Referring to Table 2, it was expected that the DBSCAN algorithm may perform badly if the underlying data contained natural divisions of similar densities. By examining Figures 46a and 46c below, while near the origin there may be an area of slightly higher density the remainder of the active space contains a relatively even spread of points which would pre-empt the DBSCAN performing poorly.
This method falls into the category of non-parametric clustering and therefore the number of clusters k was not supplied. The algorithm instead sought to *learn* a value for k that would suit the underlying data. In Figure 46a this k was found to be 206. This value is not only far higher than the value supplied by the statistical tests in Section 3 - which found k to be 22 for Quadrant 2 in the B12 dataset - but there are only 591 points in this part of the graph. This means that the algorithm found a value for k which is almost half the total points provided. A result that is not desirable as it means that a fair amount of points exist within their own cluster. If these points where in fact noise then the results would not seem so obscure, but there is not enough information to totally exclude these points from further analysis simply because they could be considered noise according to this particular algorithm.



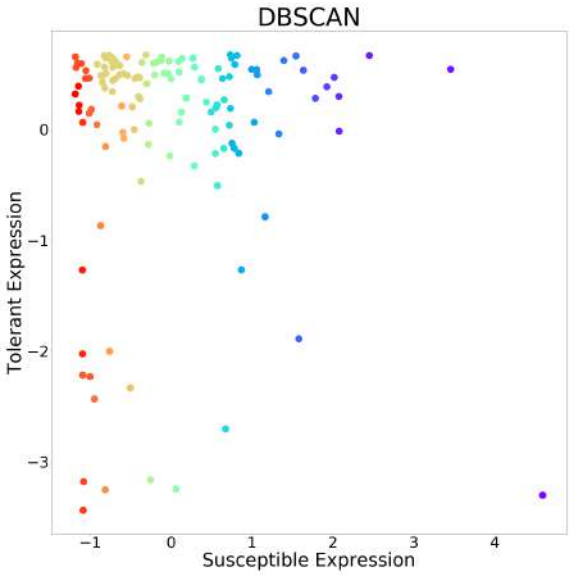
(a) Quadrant 2



(b) Quadrant 1



(c) Quadrant 3

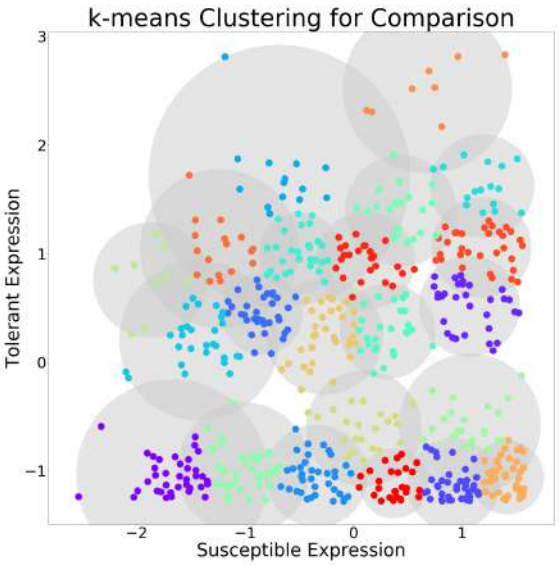


(d) Quadrant 4

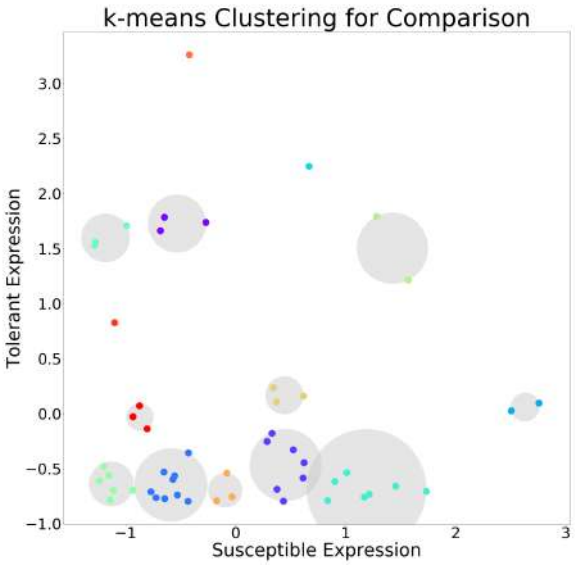
Figure 46: Plots depicting the results when DBSCAN clustering is applied to the B12 dataset.

4.10 Gaussian Mixture Model

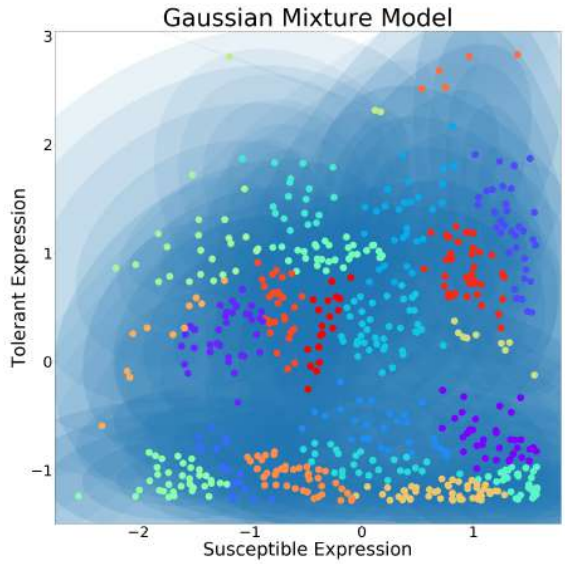
A distinct advantage of the Gaussian Mixture Model is that it does not have an inherent bias towards circular clusters. Specifically this model is able to identify clusters that are ellipsoidal in shape. Like fuzzy clustering, the Gaussian Mixture Model is also able to produce fuzzy clustering assignments and not hard assignment.



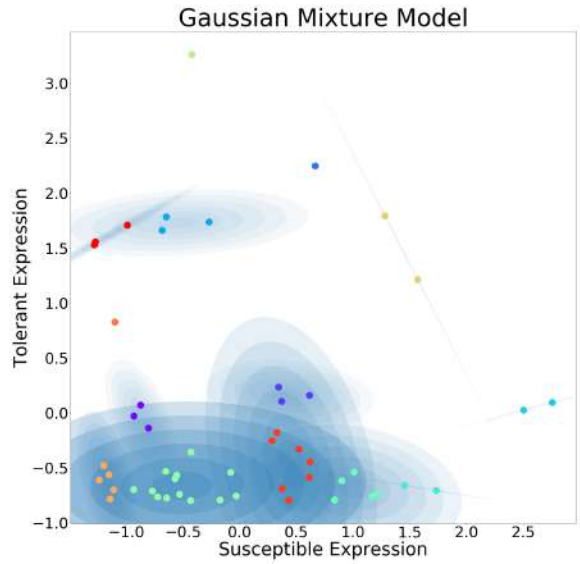
(a) Quadrant 2



(b) Quadrant 1



(c) Quadrant 2



(d) Quadrant 1

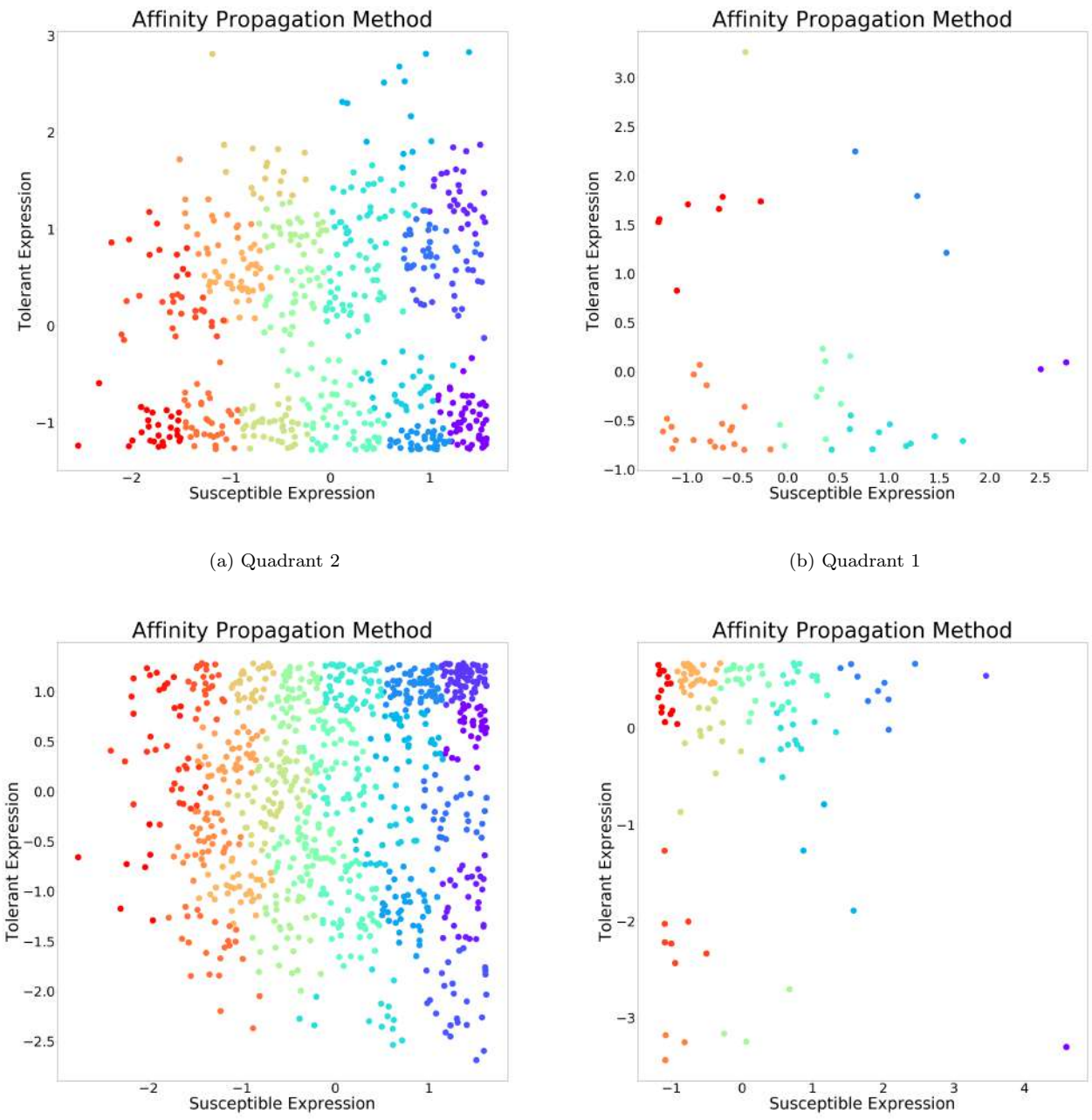
Figure 47: Plots depicting the results when applying the Gaussian Mixture Model to Quadrant 1 and 2 of dataset B12(Figure 47d and Figure 47c respectively). The clustering results for k-means have been supplied for comparison, these being Figures 47b and 47a for Quadrant 1 and 2 respectively.

The result of these is that clusters produced by this model tend to be more flexible in shape and size than a traditional method such as the k-means method.
When evaluating Figure 47c, it is clear that the space is not dominated by spherical clusters of even shape. Instead, there are clusters that vary in size as well as clusters that have distinct ellipsoidal shapes. While this flexibility could be considered an improvement on the more rigid methods that have been discussed, there are groupings in Figures 47a and 56a with more data points which seem visually inaccurate. An example of this is the thin ellipsoidal cluster on the left side of Figure 56a which groups together the dark yellow points. These points span a relatively large range and it is possible that in this instance more compact and spherical clusters would be more fitting(when taking into account the dispersion of the data in this area).

4.11 Affinity Propagation

The Affinity Propagation Algorithm seeks to identify the points in a dataset that would make suitable exemplars of the clusters in the data. These exemplars would be individuals that best represent the cluster in which they exist. Furthermore, the algorithm uses these points to learn a value for k that would best describe the underlying data[39]. In Figure 48a the algorithm found k to be 21 this is just one short of the value found for k in Section 3. The similarity of these values could statistically validate the use of this algorithm for the given dataset.
There is however a drawback with Affinity Propagation that could prove problematic. This being that the algorithm does not necessarily converge to a global optima. It is possible that the solution presented above is simply one suboptimal solution that is available for the data.

[39] k here would be equal to the number of exemplars in the given dataset.



(a) Quadrant 2

(b) Quadrant 1

(c) Quadrant 3

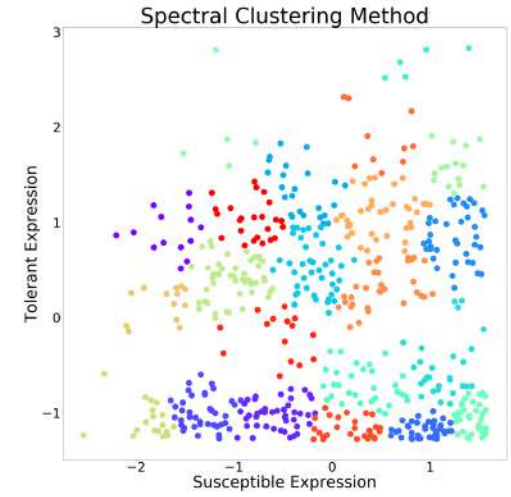
(d) Quadrant 4

Figure 48: Plots depicting the results when applying the Affinity Propagation clustering method to the B12 dataset.

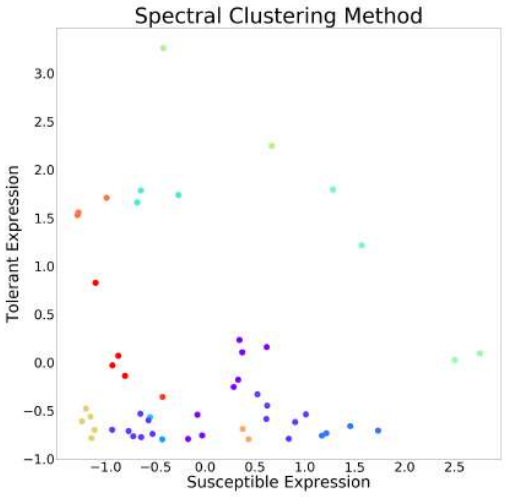
4.12 Spectral Clustering

Spectral clustering uses the notion of weighting and similarity to cluster a given data set. The goal here is simply to ensure that points that are similar are placed in the same cluster while dissimilar points exist in different clusters. If

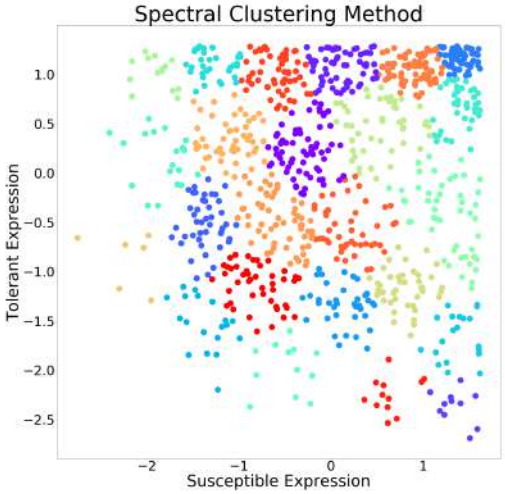
in a similarity graph the vertices represent the data points,where these vertices are connected only if the similarity of the points exceeds a certain threshold and the weighting of the edge is defined by the corresponding similarity between the points. Then a solution is found wherein similar points are grouped together. This method of defining similarity therefore does not inherently tend toward clusters of a predefined shape. Instead the similarity of points takes precedent over the prevailing cluster shape and size. In the below examples a value of k (as found in Table 3) was supplied for each Quadrant of the data and then the algorithm was applied. For larger datasets (Figure 49a and Figure 49c) the clusters appear to be relatively well-behaved. This is to say that they are relatively compact and appear to group together similar points(if position is to be used to define similarity). Conversely when a smaller dataset in provided, as is the case in Figure 49b and Figure 49d, then a few clusters are produced that group together seemingly disparate points[40].



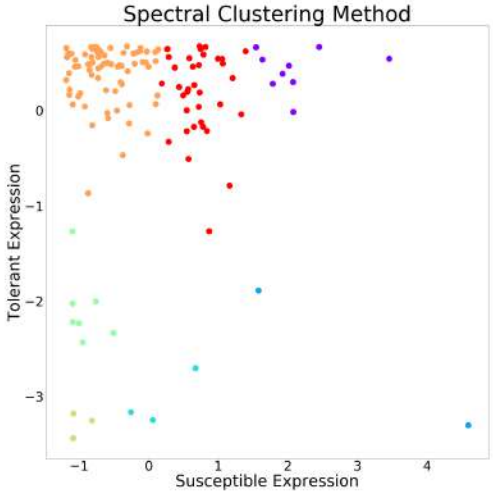
(a) Quadrant 2



(b) Quadrant 1



(c) Quadrant 3



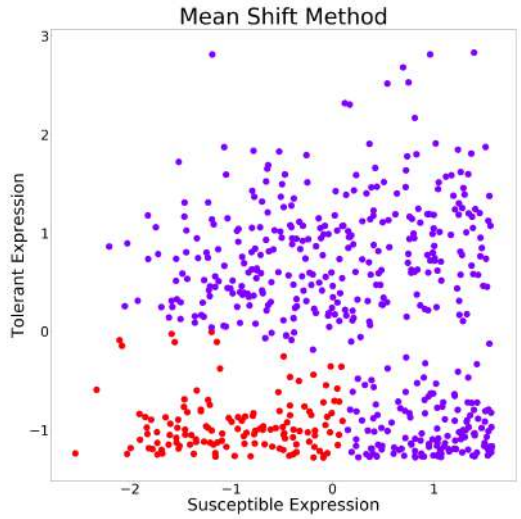
(d) Quadrant 4

Figure 49: Plots depicting the results when applying Spectral Clustering to the B12 dataset.

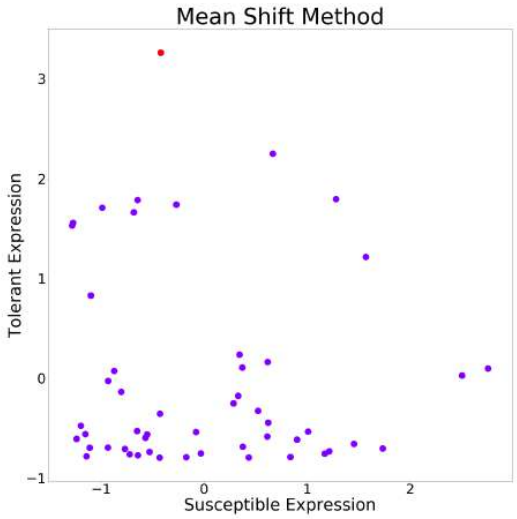
[40] Points that are positionally substantially separated.

4.13 Mean Shift Clustering

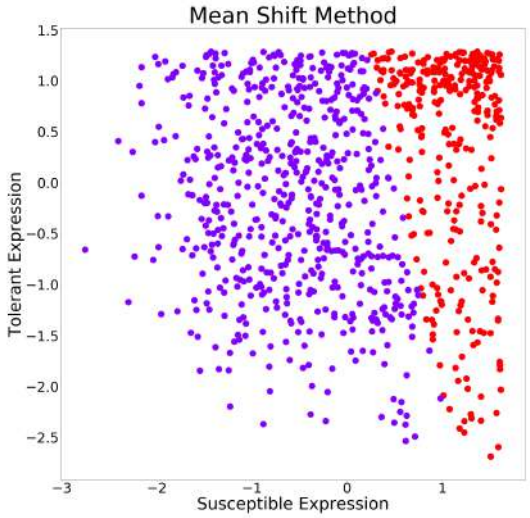
The Mean Shift method is an unsupervised clustering technique and therefore does not require a value for k to initiate the algorithm. It instead uses the density of the dataset in an attempt to learn a value for k. This is incredibly useful if a value for k is not known, but if the underlying data is not naturally separated into distinct areas of differing densities, then the method could produce somewhat erroneous results. Looking at the below graphs, this method has found a value for k of 2 in Figure 50a. This is substantially different from the value found in Section 3 which was 22. The result of this reduced value for k is large clusters that group together points that span vast portions of the available space. This same result is seen in Figure 50c, where only 2 clusters have been created for the given dataset.



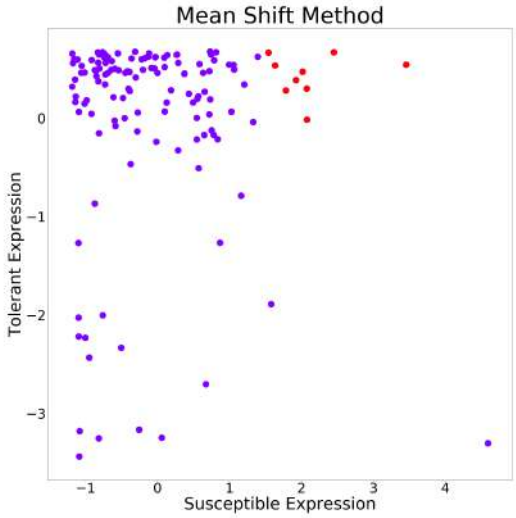
(a) Quadrant 2



(b) Quadrant 1



(c) Quadrant 3



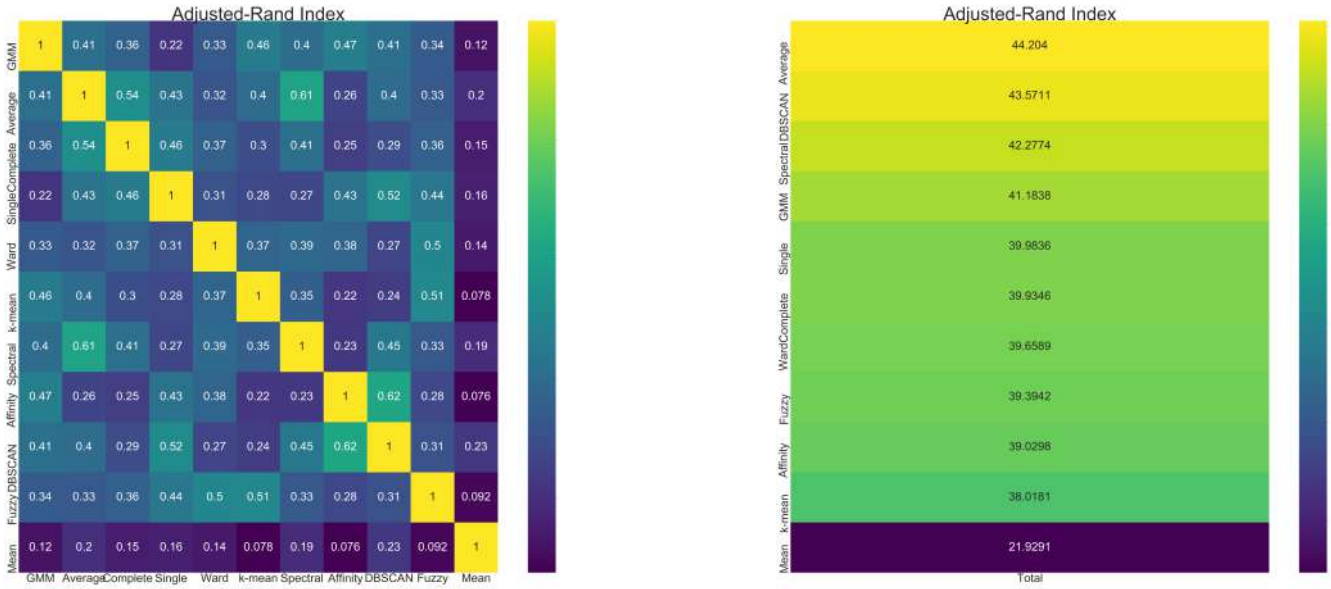
(d) Quadrant 4

Figure 50: Plots depicting the results when applying Mean Shift clustering to the B12 dataset.

4.14 Comparison of Results Obtained from Clustering

4.14.1 Adjusted-Rand Index - B12

The adjusted Rand Index attempts to test the consensus between the different clustering methods by checking where the methods agree and disagree on clustering assignments. The below graphs were created by iteratively letting each clustering result be the *true* assignment and then testing each other method against that to see how much consensus there exists between the results. A value closer to 1 would indicate a high level of consensus, while values closer to zero indicate less agreement. Looking at the below graphs Figure 51a in particular, it would seem that the highest level of agreement exists between DBSCAN and Affinity Propagation as well as Spectral Clustering and Average linkage in Quadrant 1 of the B12 dataset.



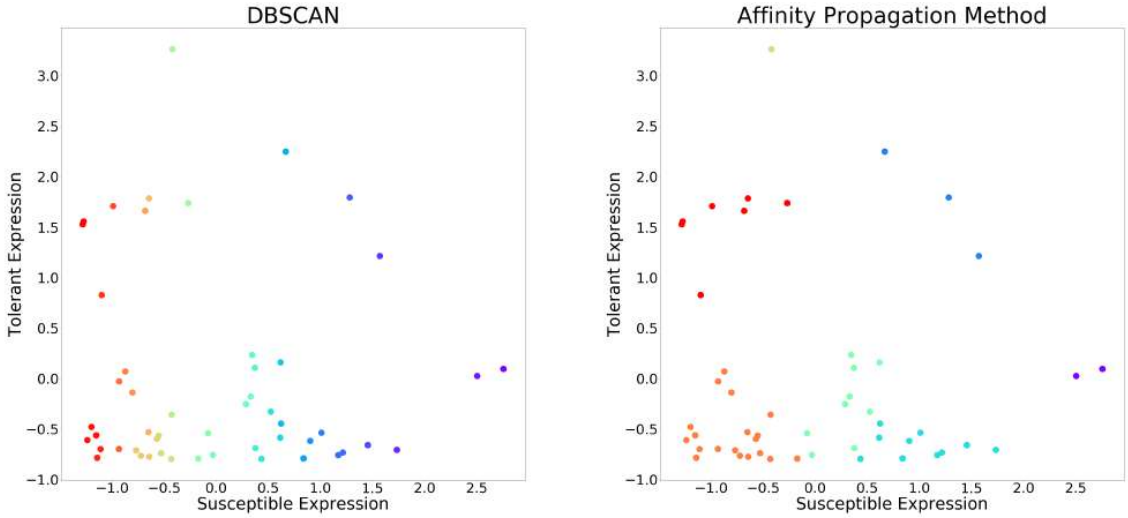
(a) All Methods Quadrant 1

(b) Concensus Totals Quadrant 1

Figure 51: Plots providing the Adjusted-Rand Index values when comparing the clustering methods proposed in Section 3. Figure 51a provides the index values between the clustering methods, while Figure 51b provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 1 of the B12 dataset.

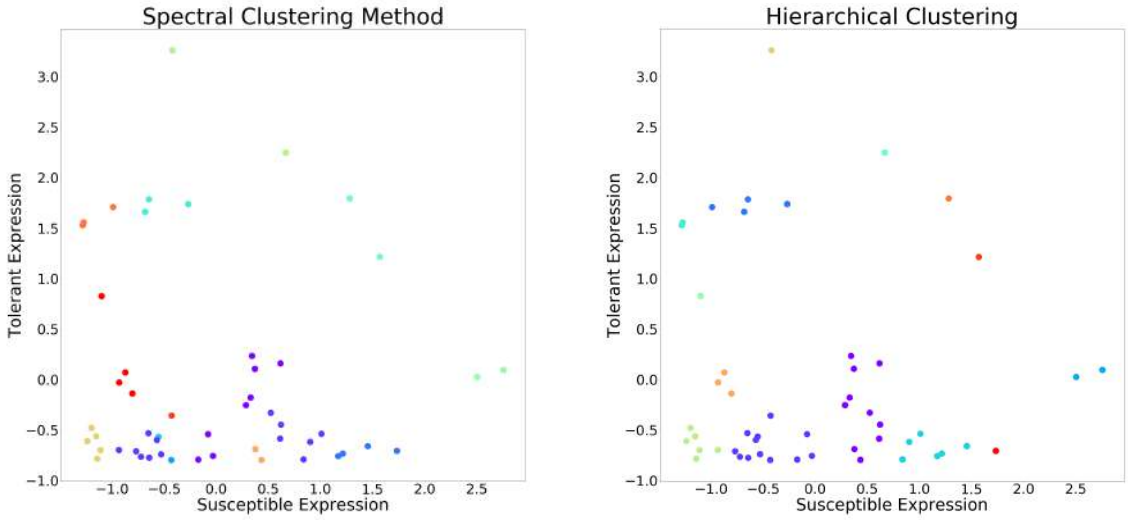
Looking at the results for DBSCAN and Affinity Propagation in Figure 52b and Figure 52a respectively, it is clear that the results produced by both are similar. DBSCAN found k to be 5[41] in Quadrant 1 of the B12 dataset and Affinity Propagation found k to be 7. Furthermore, the data assignment into these clusters was similar with each method thus resulting in a relatively high Adjusted-Rand Index.

[41] Recall that the DBSCAN is an unsupervised clustering technique and not a statistical test to determine k. It therefore finds a value for k without the need for this information to be supplied beforehand.



(a) Quadrant 1 (b) Quadrant 1

Figure 52: Plots depicting the clustering results in Quadrant 1 of the B12 dataset when applying the DBSCAN algorithm and Affinity Propagation method respectively.

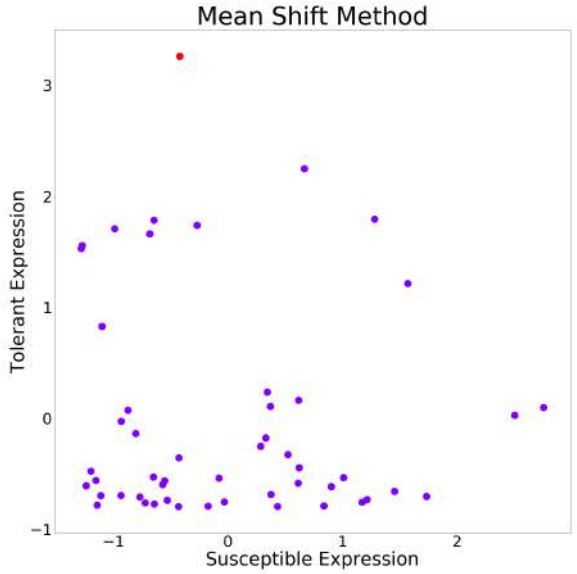


(a) Quadrant 1 (b) Quadrant 1

Figure 53: Plots depicting the clustering results in Quadrant 1 of the B12 dataset when applying the Spectral Clustering and Hierarchical Clustering with average linkage respectively.

With the case of consensus between Spectral Clustering and Average linkage, the similarity between Figure 53a and Figure 53b is not as strong as the relation between DBSCAN and Affinity Propagation but it is clear that there is a fair amount of agreement between these methods with this set of data. Agreement between these two methods would seem plausible when the way in which they define *similarity* is considered. In Spectral clustering

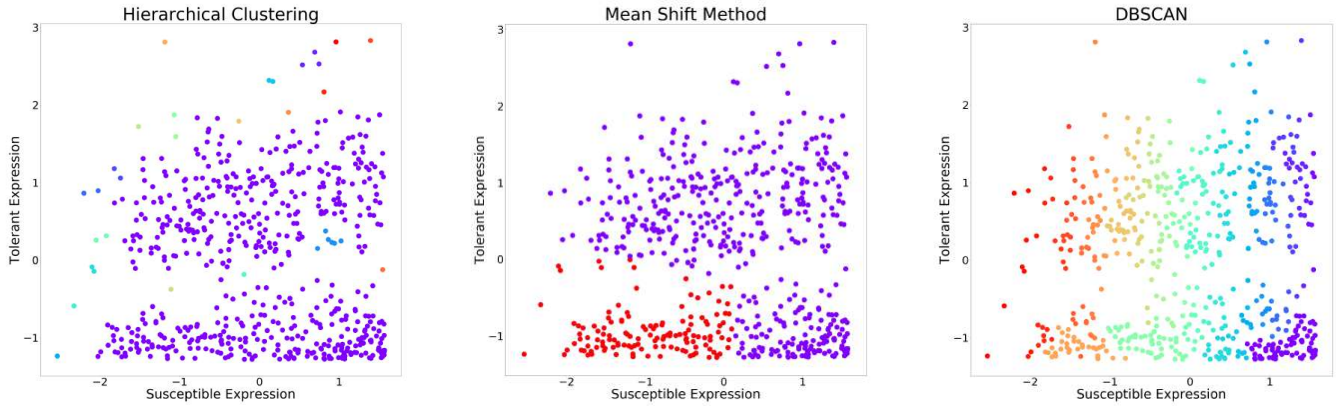
the similarity between 2 points(where they are in the data space) is the factor that is used when deciding which clusters to merge and in Average Linkage all points in a cluster are used in the measurement of the distance between clusters which then relates to the similarity of said clusters.
When looking at the level of agreement between all the clusters, as depicted in Figure 51b, Average Linkage seems to have the highest level of consensus across the board. But this total value of agreement is not very high when looking at the scale of the other totals. 10 out of the 11 of the methods have total values that fall within a range of 0.9. This is not markedly significant and if the actual clustering is examined alongside these values it is clear that the differences in clustering results are not stark, with the exception of Mean Shift Clustering. Figure54a shows one large cluster near the origin of the graph which could account for this divergence in agreement with the other methods.



(a) Mean Shift Quadrant 1

Figure 54: Plot depicting the result when applying Mean Shift clustering to Quadrant 1 of the B12 dataset. Here the algorithm found k to be 5.

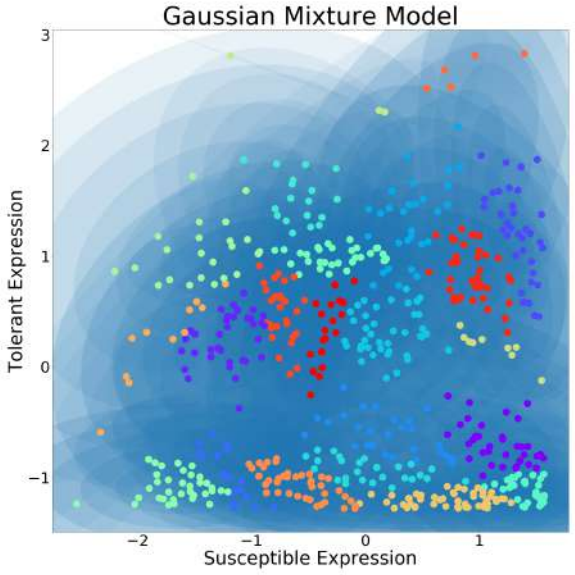
This lack of consensus between the Mean Shift Algorithm and the other methods suggests that a method which produces larger clusters will have a lower Adjusted-Rand Index when compared to methods that produce relatively smaller clusters. This result is seen again in Quadrant 2 of dataset B12 (Figure 57b) where the methods with the total lowest consensus are Single Linkage, Mean Shift and DBSCAN, all of which have relatively large clusters that dominate the data space (Figure 55a, Figure 55b and Figure 55c respectively).



(a) Single Linkage Quadrant 2 (b) Mean Shift Quadrant 2 (c) DBSCAN Quadrant 2

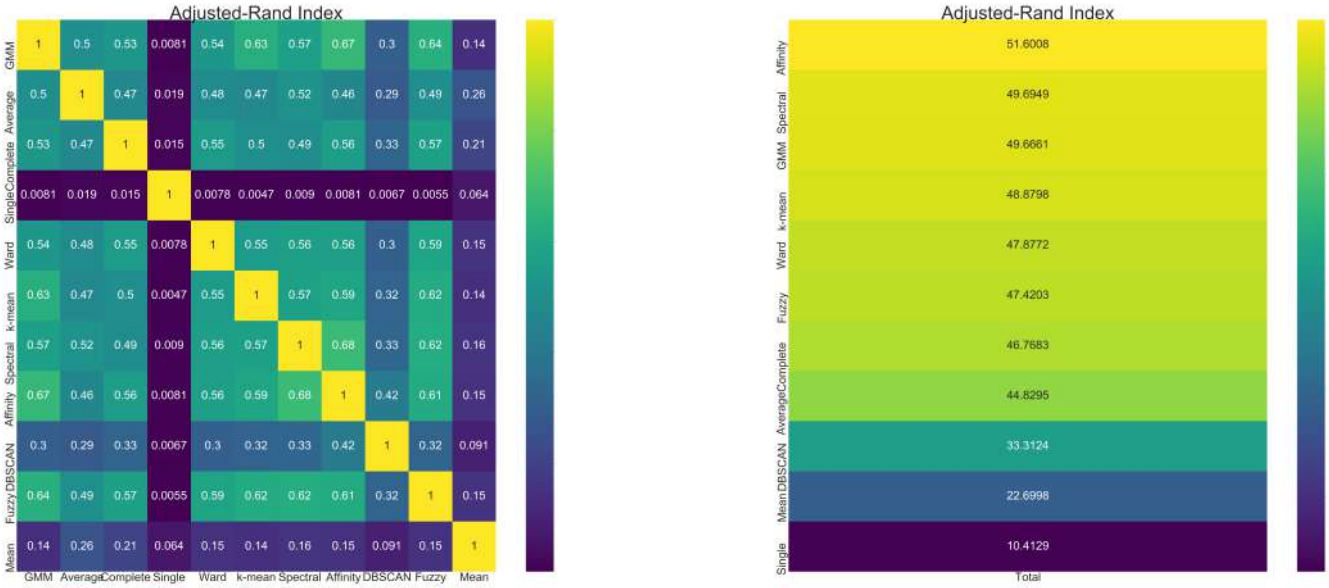
Figure 55: Plots depicting the results when applying Hierarchical Clustering with single linkage, Mean Shift Clustering and the DBSCAN algorithm to Quadrant 2 of the B12 dataset respectively.

The last feature of the Adjusted-Rand Index which is visible in the results here is that it does not make a strong assumption on the structure of the cluster. This means, that spherical clusters can be compared to more irregular shaped clusters, as is the case when comparing results from the Gaussian Mixture Model. Despite the existence of ellipsoidal clusters in the results of the Gaussian Mixture model in Figure 56a, there is still a fair amount of agreement between this model and the other methods which cannot specifically identify ellipses in the data.



(a) Quadrant 3

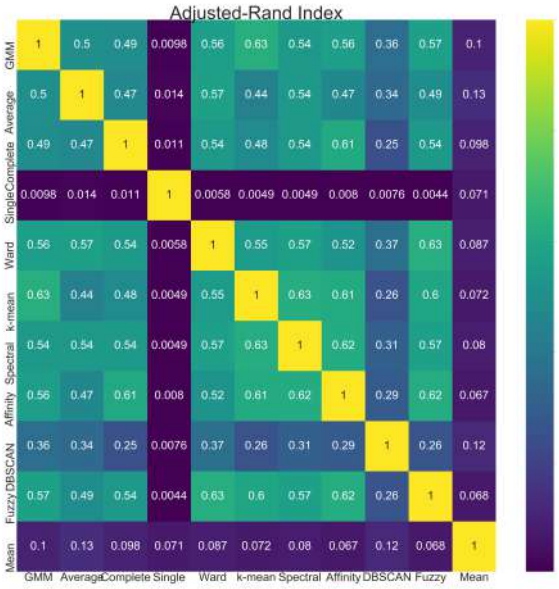
Figure 56: Plot depicting the result when applying the Gaussian Mixture Model to Quadrant 3 of the B12 dataset.



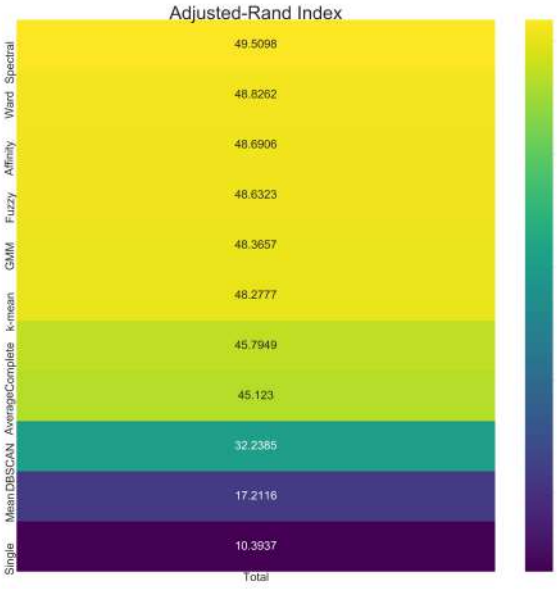
(a) All Methods Quadrant 2

(b) Consensus Totals Quadrant 2

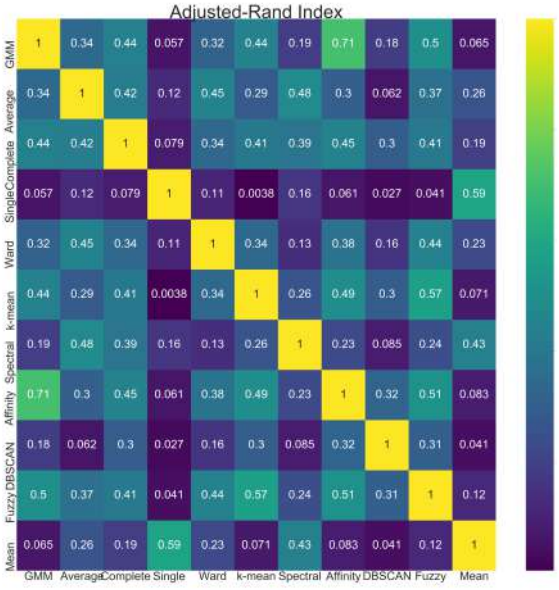
Figure 57: Plots providing the Adjusted-Rand Index values when comparing the clustering methods proposed in Section 3. Figure 57a provides the index values between the clustering methods, while Figure 57b provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 2 of the B12 dataset.



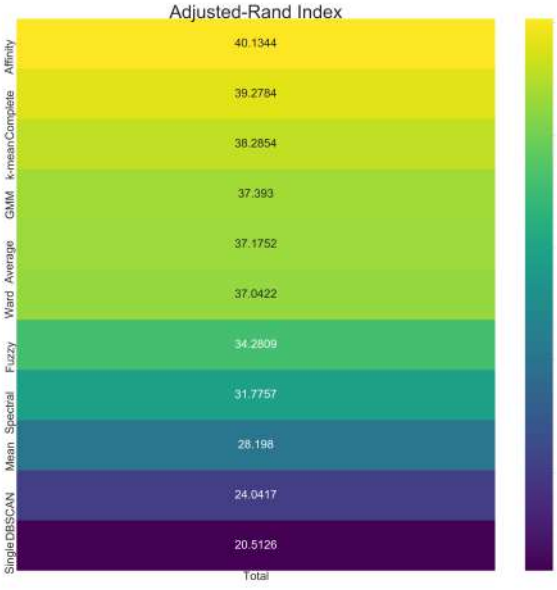
(a) All Methods Quadrant 3



(b) Concensus Totals Quadrant 3



(c) All Methods Quadrant 4



(d) Consensus Totals Quadrant 4

Figure 58: Plots providing the Adjusted-Rand Index values when comparing the clustering methods proposed in Section 3. Figures 58a and 58c provides the index values between the clustering methods, while Figures 58b and 58d provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 3 and 4 of the B12 dataset.

4.14.2 Mutual Information Based Score - B12

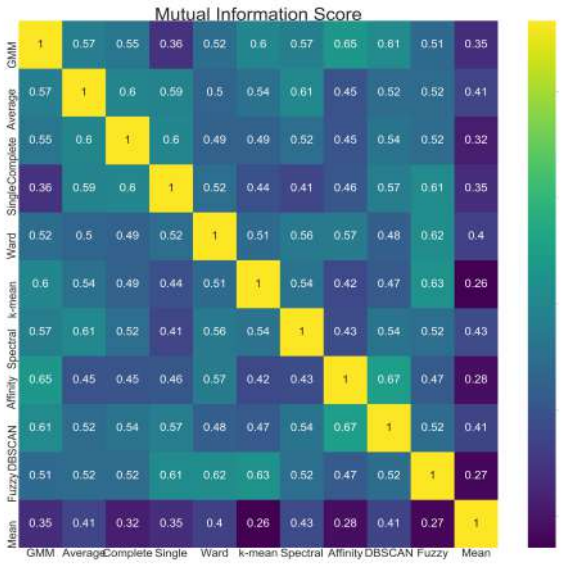
The Adjusted Mutual Information Score that was used here tests the agreement between the assignments that are made by two different clustering techniques. This method has been adjusted for chance (unlike the Normalised Mutual Information Score and Mutual Information Score) and therefore will not naturally increase with the number of clusters - k.
Looking at Figure 59a, DBSCAN and Affinity Propagation have the highest level of agreement (as was the case with the Adjusted Rand Index). This is an expected result as both these indices are simply measuring where methods agree on their assignments. According to M. Warrens & H. Van der Hoef, the Adjusted Rand Index is better suited for methods that produce clusters that are large and equal in size, while the Adjusted Mutual Information Score is more appropriate for methods that produce unequal sized clusters that include small clusters [54].
Looking at Table 5 and Table 6 alongside Figure 59a, it is fitting that the highest Adjusted Mutual Information Score exists between these two. Not only do they have similar groupings (as was seen in Figure 52a and 52b), but they both have clusters that are unequal in size including small clusters.

Table 5: DBSCAN Clusters

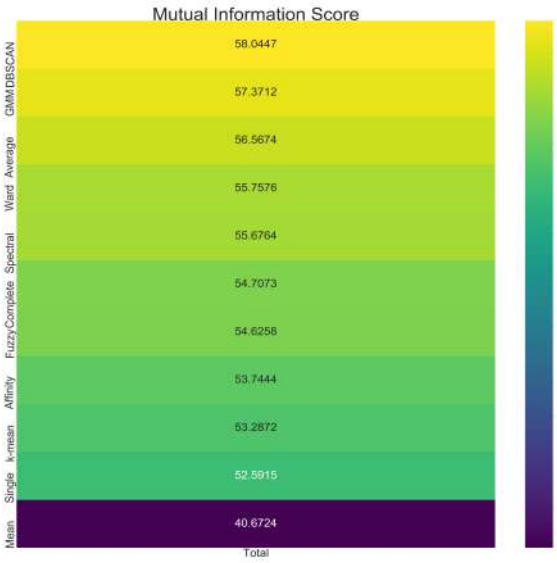
Cluster	No. of pts
-1	7
0	17
1	18
2	6
3	3

Table 6: Affinity Propagation Clusters

Cluster	No. of pts
0	2
1	3
2	10
3	9
4	1
5	19
6	7

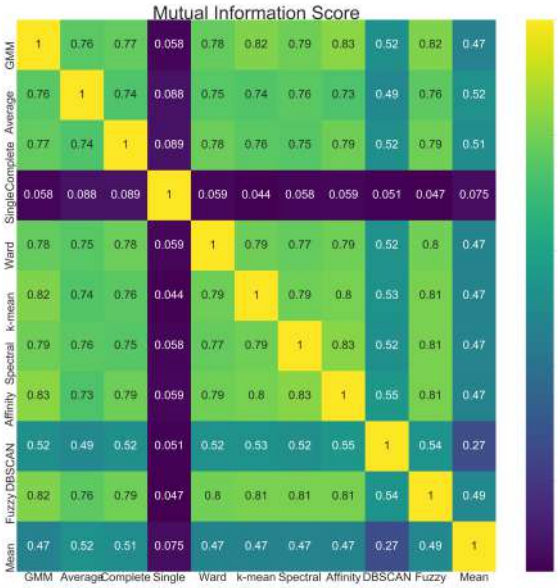


(a) All Methods Quadrant 1

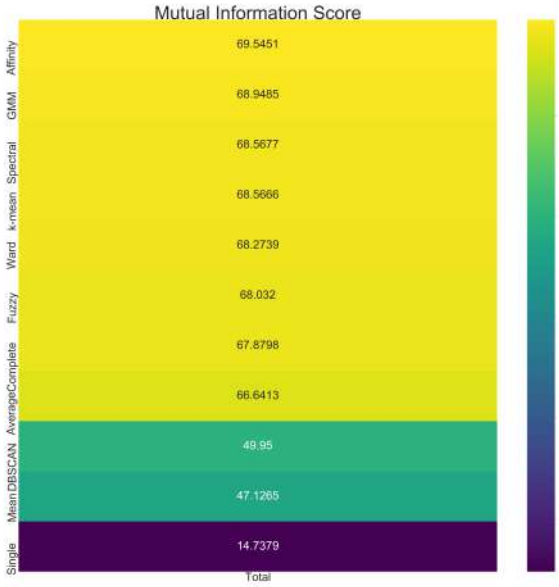


(b) Concensus Totals Quadrant 1

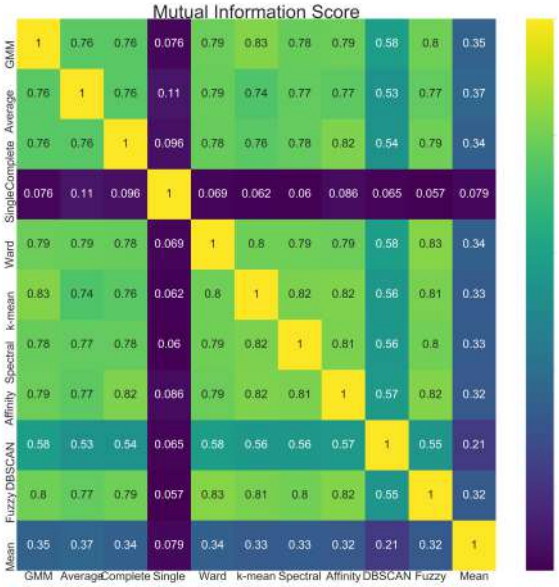
Figure 59: Plots providing the Mutual Information Scores when comparing the clustering methods proposed in Section 3. Figure 59a provides the index values between the clustering methods, while Figure 59b provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 1 of the B12 dataset.



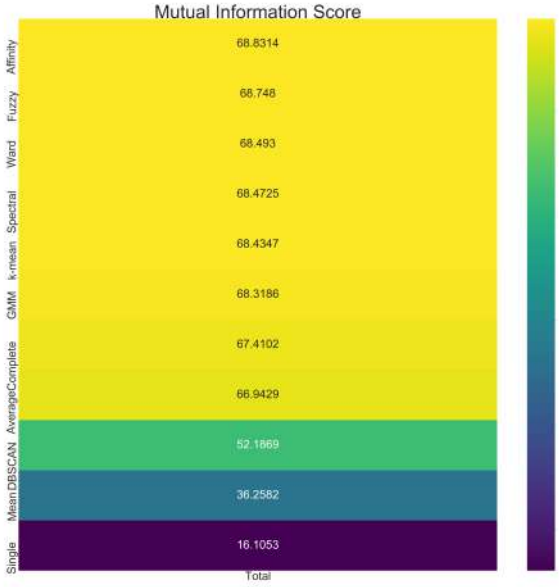
(a) All Methods Quadrant 2



(b) Consensus Totals Quadrant 2

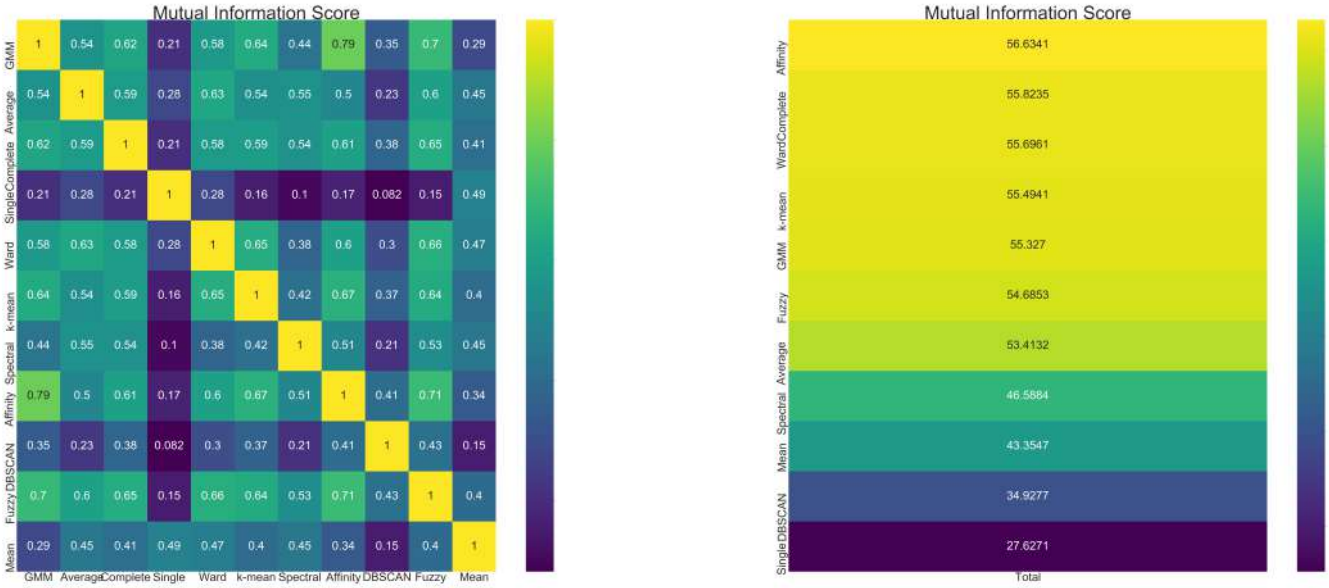


(c) All Methods Quadrant 3



(d) Concensus Totals Quadrant 3

Figure 60: Plots providing the Mutual Information Scores when comparing the clustering methods proposed in Section 3. Figures 60c and 61a provides the index values between the clustering methods, while Figures 60d and 61b provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 2 and 2 of the B12 dataset.



(a) All Methods Quadrant 4

(b) Consensus Totals Quadrant 4

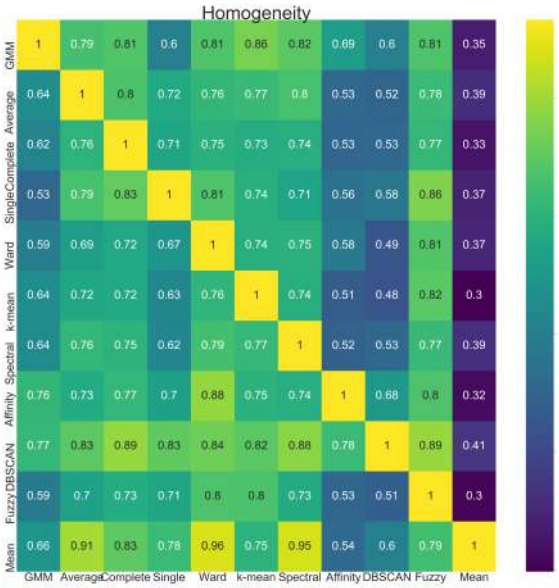
Figure 61: Plots providing the Mutual Information Scores when comparing the clustering methods proposed in Section 3. Figure 61a provides the index values between the clustering methods, while Figure 61b provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 4 of the B12 dataset.

4.14.3 Homogeneity and Completeness - B12

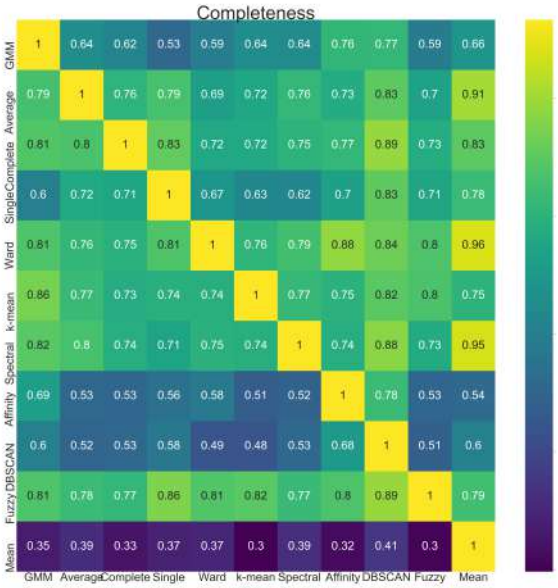
Recall here that a clustering result is homogenous if each cluster only has points that are considered to be in the same class. While completeness exists in clustering when all members of a given class are assigned to the same cluster. These definitions of *class* would come from the *true* label assignments, which in this case is the result from each method which is then compared to the rest of the methods.
Before evaluating the results obtained from the Homogeneity and Completeness scores it is important to note that these scores are not symmetric, instead it is governed by the relation in

$$homogeneity(a, b) == completeness(b, a). \tag{4.14.1}$$

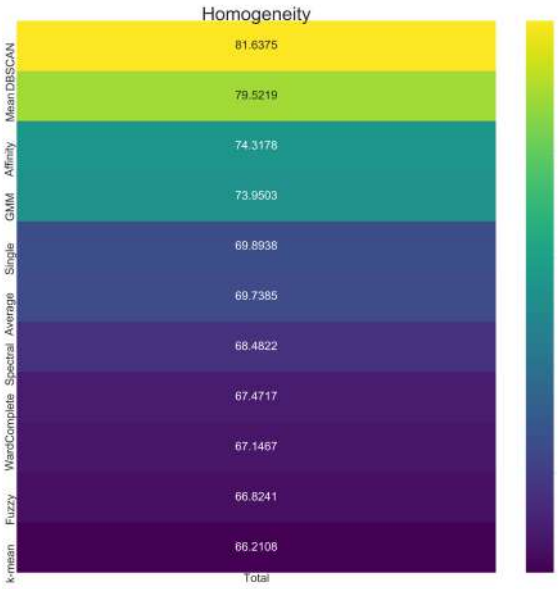
This means that the homogeneity score calculated when a result **a** is assigned as the *true* label while a result **b** is assigned the *predicted* label, would be equal to the completeness score calculated for the inverse of these.
Looking at the graphs which map the completeness scores between the methods, it is expected that when a method which creates a small number of large clusters is assigned to be the true label, it will have low completeness scores with methods producing smaller clusters. This is because the completeness measure would interpret these smaller clusters as dividing points of the same class into multiple clusters, as is the case with Mean Shift clustering and Single linkage in Figure 63b.
Due to the relation in Equation 4.14.1, the inverse will be true in measuring homogeneity for methods that create large clusters. As seen in Figure 63a, homogeneity scores between Mean Shift Clustering and methods that produce smaller clusters will result in high homogeneity scores as it could appear to the index that smaller clusters that are created by the other methods do group together only points that belong to the same class.



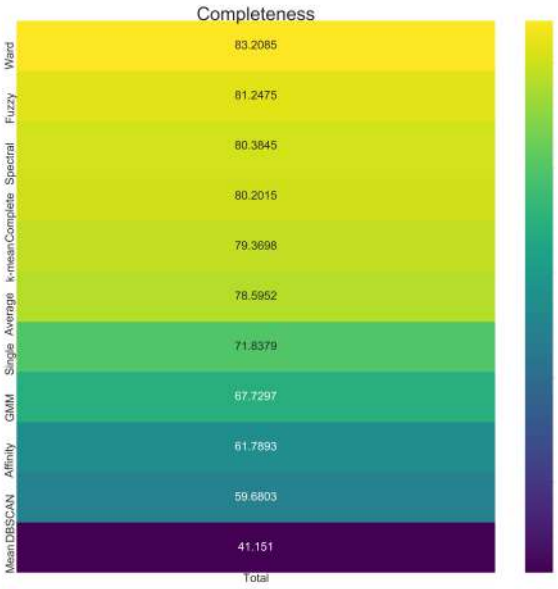
(a) All Methods Quadrant 1



(b) All Methods Quadrant 1

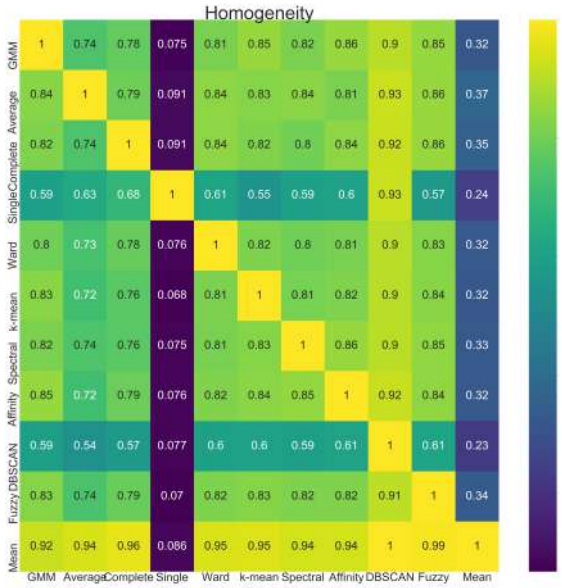


(c) Consensus Totals Quadrant 1

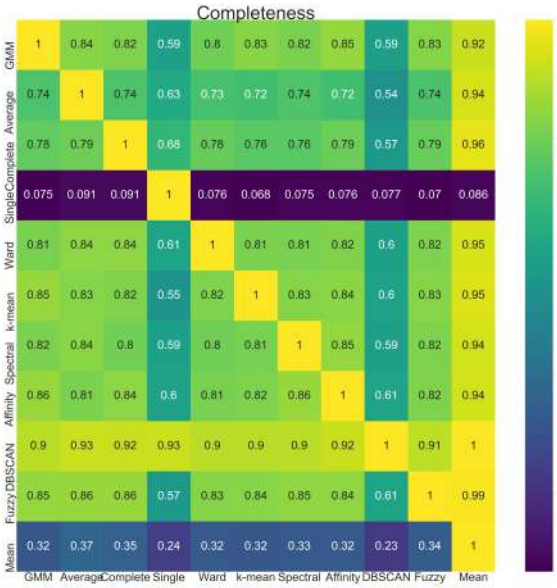


(d) Consensus Totals Quadrant 1

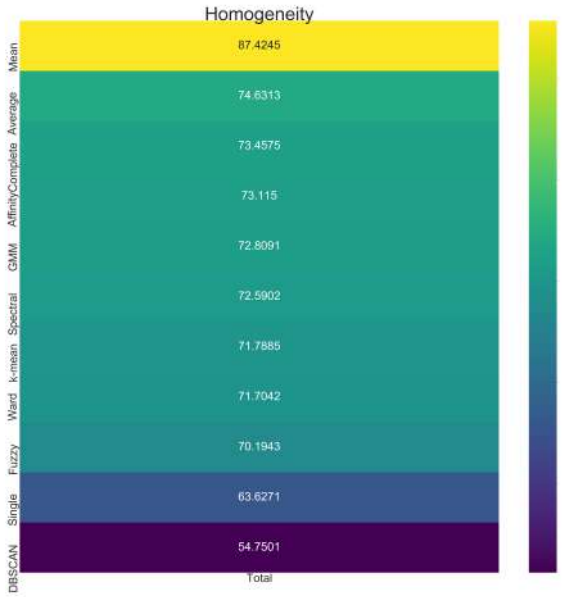
Figure 62: Plots providing the Homogeneity and Completeness scores when comparing the clustering methods proposed in Section 3. Figure 62a provides the Homogeneity scores between the clustering methods, while Figure 62b provides the Completeness scores.Figures 62c and 62d provides the total value(given as a percentage) for the Homogeneity and Completeness scores when compared to the remaining methods. Figures relate to data in Quadrant 1 of the B12 dataset.



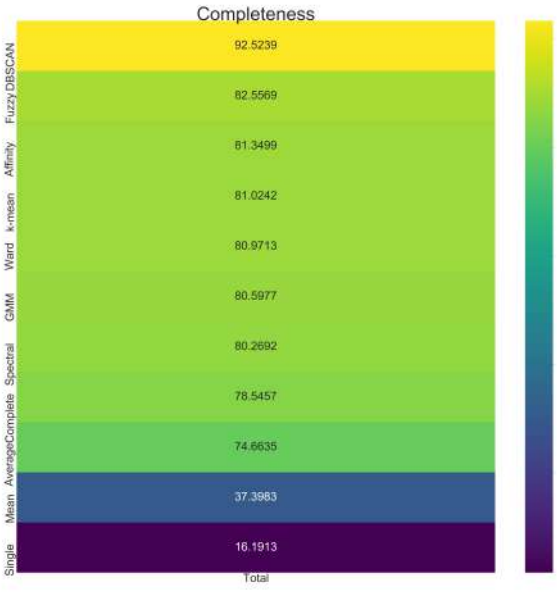
(a) All Methods Quadrant 2



(b) All Methods Quadrant 2



(c) Consensus Totals Quadrant 2

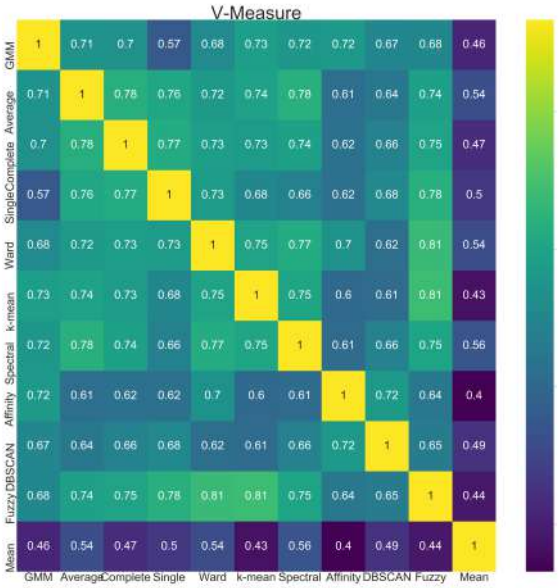


(d) Consensus Totals Quadrant 2

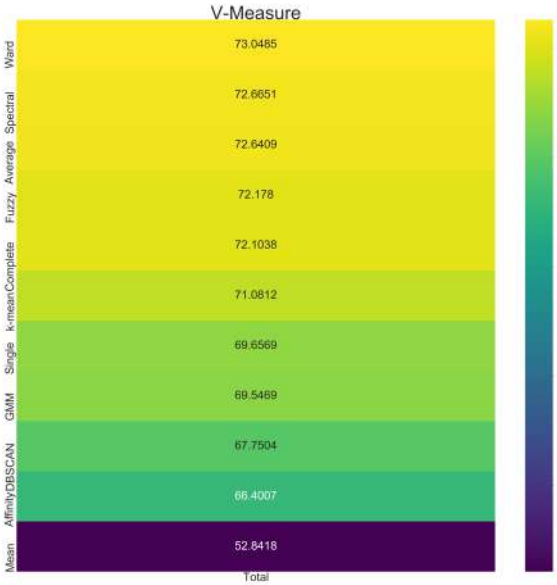
Figure 63: Plots providing the Homogeneity and Completeness scores when comparing the clustering methods proposed in Section 3. Figure 63a provides the Homogeneity scores between the clustering methods, while Figure 63b provides the Completeness scores.Figures 63c and 63d provides the total value(given as a percentage) for the Homogeneity and Completeness scores when compared to the remaining methods. Figures relate to data in Quadrant 2 of the B12 dataset.

4.14.4 V-Measure - B12

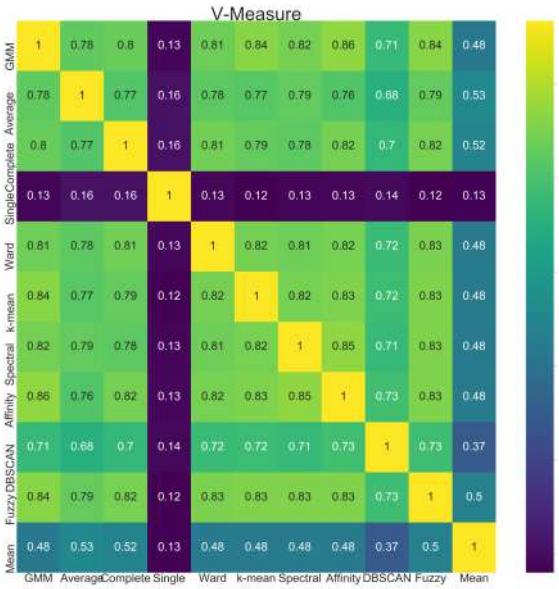
The V-measure is defined as the harmonic mean of homogeneity and completeness. This mathematical relation allows for either homogeneity or completeness to be favoured, but to find a balance between clusters that are homogenous and complete the weighting of these were made equal. The result of testing the agreement between the clustering methods in this way are depicted in Figure 64a-65d.
In reference to Figure 64c, it is clear that methods which have average sized clusters would produce higher V-Measure scores when tested against the other methods. This is depicted by methods such as the Fuzzy method, Affinity Propagation, Gaussian Mixture Model and k-means having the highest V-Measure totals (Figure 64d), while Single linkage and DBSCAN have lower totals (methods which produced few large clusters and many small clusters respectively).



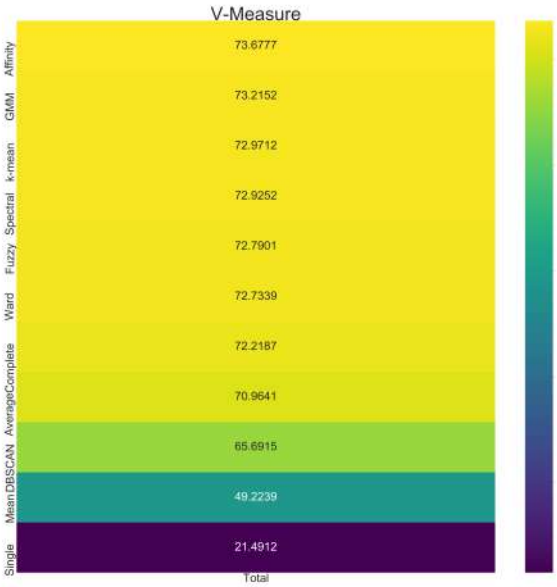
(a) All Methods Quadrant 1



(b) Concensus Totals Quadrant 1

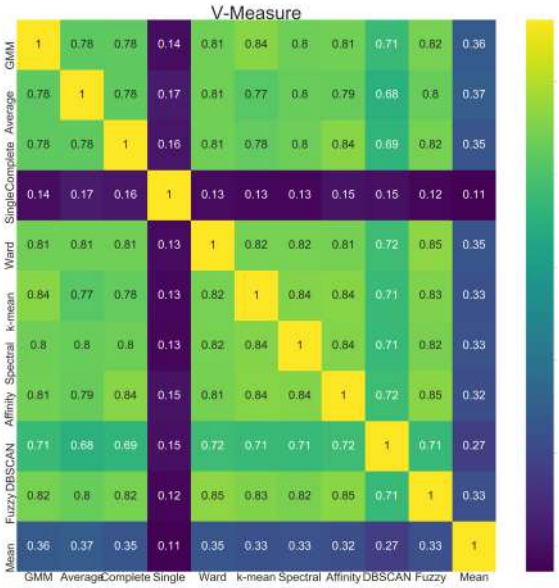


(c) All Methods Quadrant 2

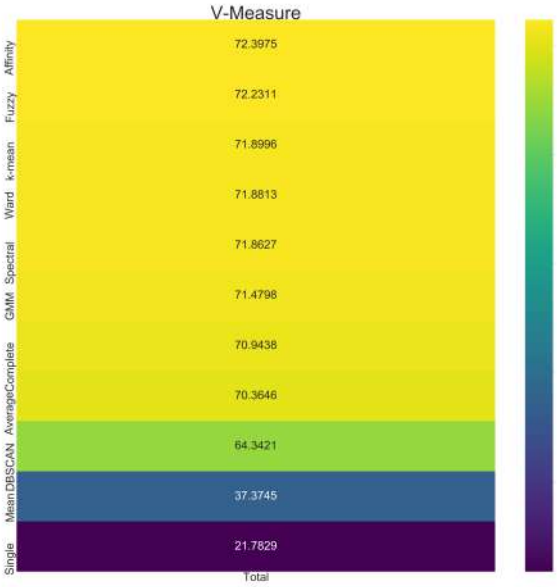


(d) Consensus Totals Quadrant 2

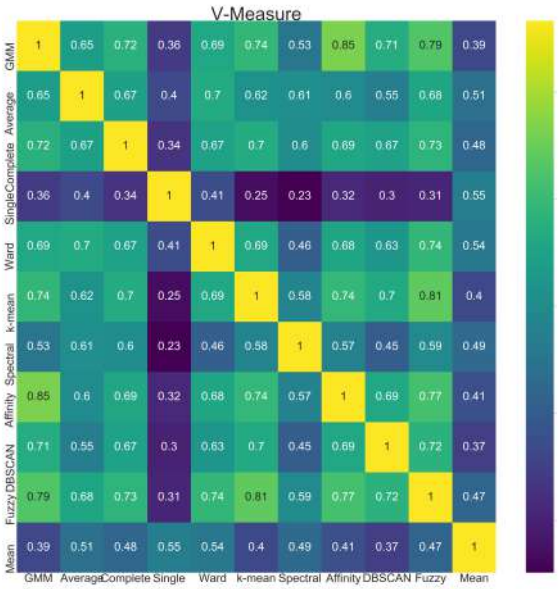
Figure 64: Plots providing the V-Measure values when comparing the clustering methods proposed in Section 3. Figures 64a and 64c provides the index values between the clustering methods, while Figures 64b and 64d provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 1 and 2 of the B12 dataset.



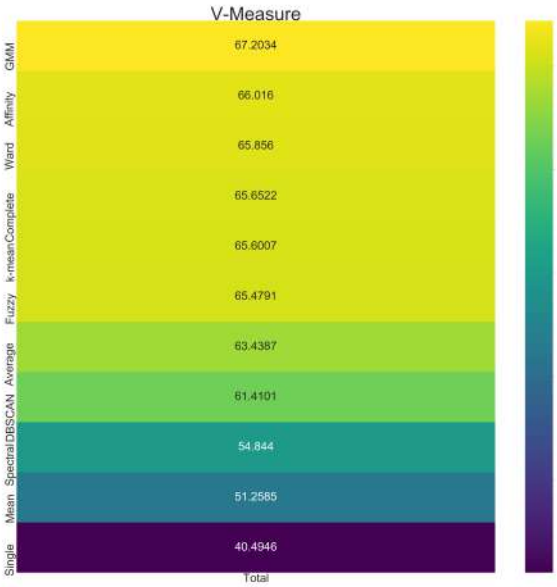
(a) All Methods Quadrant 3



(b) Concensus Totals Quadrant 3



(c) All Methods Quadrant 4

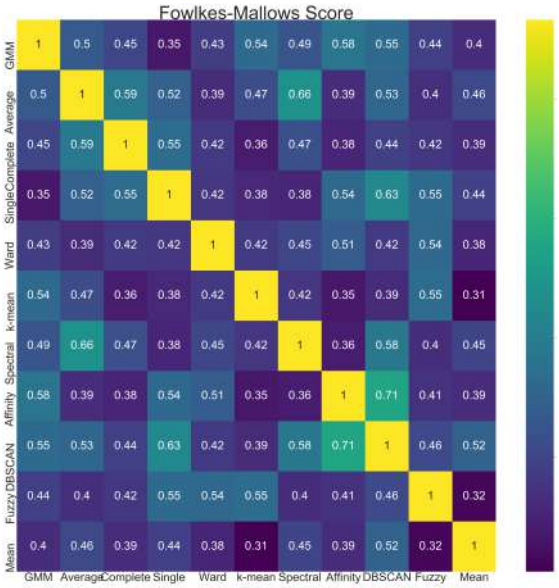


(d) Consensus Totals Quadrant 4

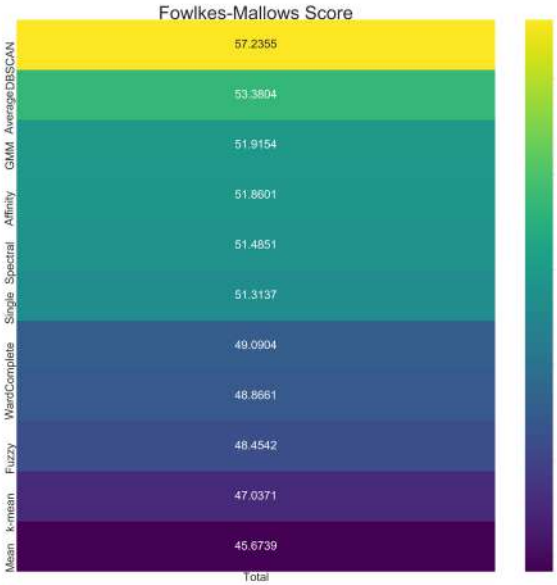
Figure 65: Plots providing the V-Measure values when comparing the clustering methods proposed in Section 3. Figures 65a and 65c provides the index values between the clustering methods, while Figures 65b and 65d provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 3 and 4 of the B12 dataset.

4.14.5 Fowlkes-Mallows Score - B12

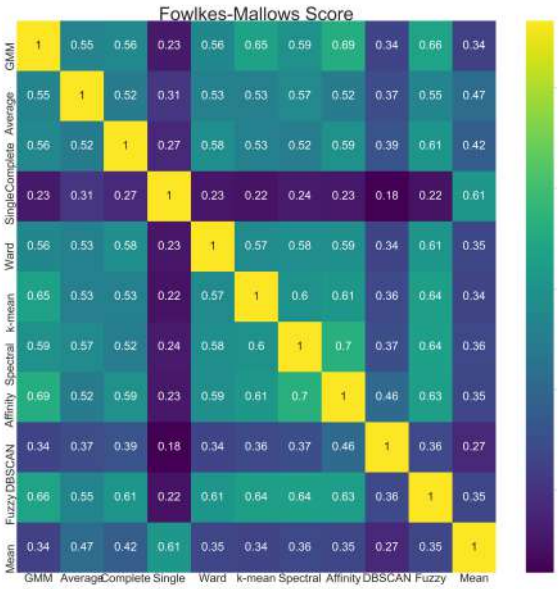
The Fowlkes-Mallows Score measures the similarity between 2 different clustering assignments much in the same way as the Adjusted-Rand Index. One marked difference between the two scores is that the Fowlkes-Mallows score accounts for points that are clustered together in the *predicted* assignment but not in the *true* assignment. This implies that in Fowlkes-Mallows calculations points that are clustered together in the *predicted* assignment but not in the *true* assignment, are not immediately written off as wrong assignments. While the labels produced by the *predicted* assignment are taken into account in Fowlkes-Mallows scoring, with this score the prevailing similarity between the two methods in question is important. The result of this being that methods which produce average sized clusters as is the case with Affinity Propagation and Fuzzy Clustering will produce higher total Fowlkes-Mallows scores as is seen in Figure 66d.



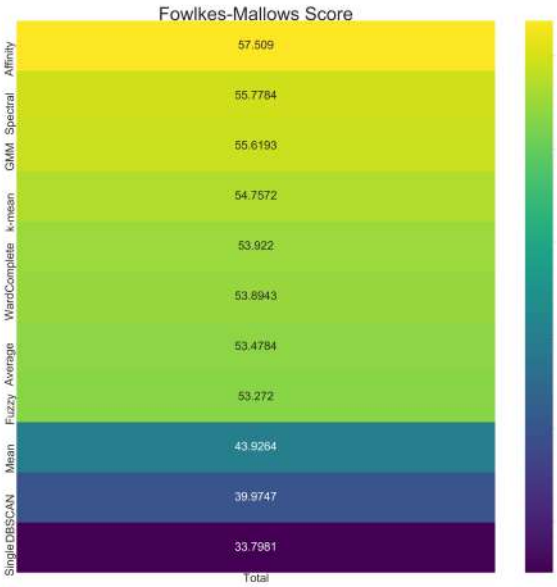
(a) All Methods Quadrant 1



(b) Concensus Totals Quadrant 1

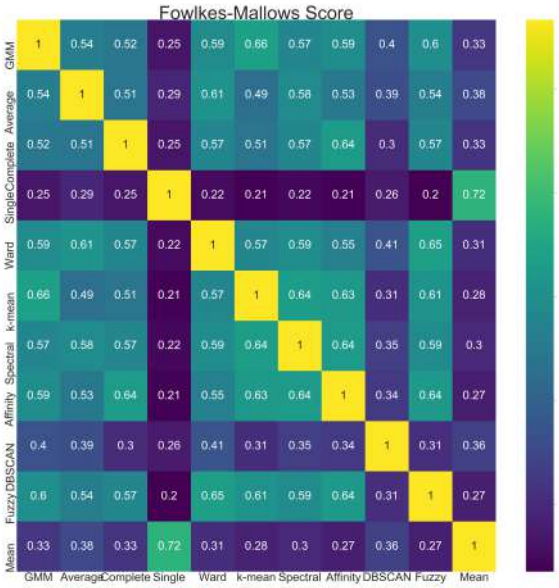


(c) All Methods Quadrant 2

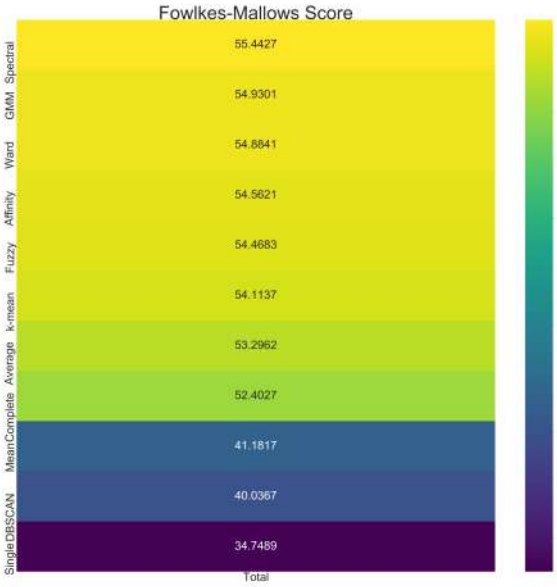


(d) Consensus Totals Quadrant 2

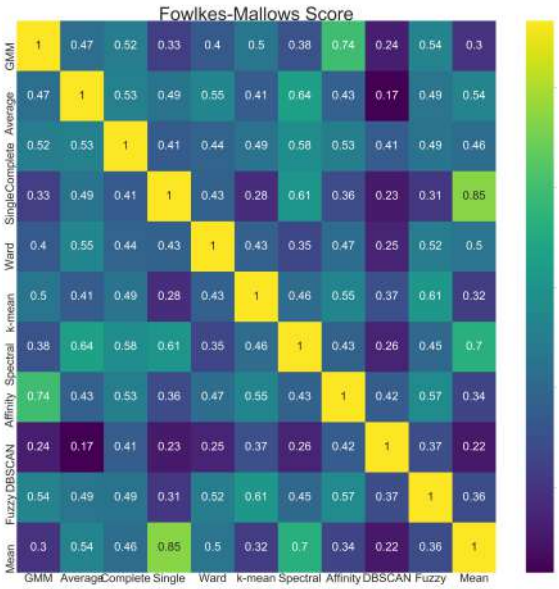
Figure 66: Plots providing the Fowlkes-Mallows score values when comparing the clustering methods proposed in Section 3. Figures 66a and 66c provides the index values between the clustering methods, while Figures 66b and 66d provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 1 and 2 of the B12 dataset.



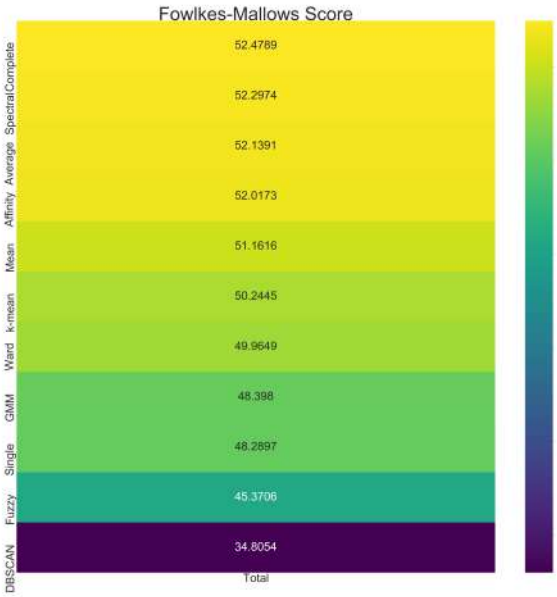
(a) All Methods Quadrant 3



(b) Concensus Totals Quadrant 3



(c) All Methods Quadrant 4



(d) Consensus Totals Quadrant 4

Figure 67: Plots providing the Fowlkes-Mallows score values when comparing the clustering methods proposed in Section 3. Figures 67a and 67c provides the index values between the clustering methods, while Figures 67b and 67d provides the total value(given as a percentage) for each method when compared to the remaining methods. Figures relate to data in Quadrant 3 and 4 of the B12 dataset.

4.15 Summary of Statistical Tests Comparing Clustering Results

Without the knowledge of *true* labelling assignments for a given dataset a *predicted* solution can take this placeholder if its assignments are sufficiently validated. This was done by looking for a consensus between the clustering results that were presented in Section 4.14. A solution which displays a high level of consensus when tested against the results from other clustering algorithms can be presented as a preliminary *true* solution.

Table 7: Consensus Across All Datasets

	ARI	AMI	Completeness	Homogeneity	V-measure	FM	Total
k-mean	42.90677	58.30466	79.17642	66.35457	69.59694	51.04462	61.23066
Fuzzy	42.55434	57.95667	81.31665	65.65482	69.22826	50.55444	61.21086
GMM	43.16628	58.12113	77.38255	67.80347	68.62162	51.79507	61.14835
Ward	43.09346	58.54387	79.90129	66.12411	68.45678	50.7568	61.14605
Affinity	42.75029	57.63457	78.00721	66.88058	68.37388	51.26241	60.81816
Complete	40.7542	56.83056	76.08508	67.78391	67.88978	50.65307	59.99943
Average	40.27902	54.88802	72.43079	69.40099	66.99711	51.6979	59.2823
Spectral	40.18642	54.73755	72.41338	68.19913	66.08714	51.17035	58.799
DBSCAN	28.95995	36.20309	73.64621	58.2307	58.9324	42.97927	49.82527
Mean	20.80974	27.85769	31.5241	86.84257	40.26393	45.7232	42.17021
Single	16.37029	18.8509	27.08887	64.94013	31.45422	41.6912	33.39927

Table 7 above is a summary of all scores(presented as percentages) as calculated by the metrics presented in Section 4.14. The colour scale used in this table colours higher values in green and lower values in red. The values presented here are produced by the summation of the scores for all clusters in the B12, B32 and B67 datasets for each clustering method.

- **ARI** - contains the scores as calculated by the Adjusted-Rand Index. It was expected that methods which produced a few large clusters along with very small clusters(only one of two points) would produce low ARI scores. This has been validated by DBSCAN, Mean Shift and Single Linkage having the lowest total scores across all datasets. Furthermore, the Gaussian Mixture Model(GMM) holds the highest total score with this index which validates the ARI's lack of inherent preference for spherical clusters[42]. High ARI values for k-mean, Fuzzy Clustering, GMM, Wards Minimum Variance and Affinity Propagation imply that average sized clusters are favoured by this index.

- **AMI** - contains the scores as calculated by the Adjusted-Mutual-Information Score. The AMI tests the similarity/consensus between two clustering assignments much in the same way as the ARI. This can be seen by k-mean, Fuzzy Clustering, GMM, Wards Minimum variance and Affinity Propagation having the 5 highest score with both of these indices. This index too favours average sized clusters.

- **Completeness** and **Homogeneity** - These results are interpreted together as they share an inverse relation due to the asymmetry of both. Using the definitions of homogeneity[43] and completeness[44], it was expected that methods producing large clusters would result in low completeness scores and high homogeneity scores when tested against methods which produce smaller clusters. This can be seen Mean Shift clustering having a high homogeneity score and low completeness score. High Completeness scores for k-means, Fuzzy Clustering,

[42]The Gaussian Mixture Model is one of the few tested methods that can specifically identify ellipsoidal clusters.
[43]Only points of the same class are included in each cluster.
[44]All points of the same class are assigned to the same cluster.

GMM, Wards Minimum Variance and Affinity Propagation result from the Completeness index interpreting the smaller clusters produced by these methods as not being split up by methods producing larger clusters, thus satisfying completeness requirements. The inverse of this is true for homogeneity as the Homogeneity Index interprets bigger clusters as the wrongful grouping together of points that belong to different classes when smaller clusters(as produced by the methods in question) are assigned as the *true* labels.

- **V-measure** - The V-measure is the harmonic mean of **completeness** and **homogeneity** (neither homogeneity nor completeness were favoured). The balance between these two indicators results in average sized clusters having the highest total scores.
- **FM** - Recall that the Fowlkes-Mallows score accounts for *predicted* assignments which do not match the *true* assignments. This has resulted in a more even spread of consensus totals, with a range of 1.63 separating the highest 8 totals. The methods with the lowest Fowlkes-Mallows scores are ones which produce one large cluster that dominates the data space. While FM does account for *predicted* assignments, the similarity between *true* and *predicted* assignments still rests at the core of this index and therefore a large cluster cannot be reconciled with many average sized clusters.

According to the **Total** column as presented in Table 7 the k-means algorithm has the highest level of agreement(consensus) with the other clustering methods that have been evaluated. This is to say that the clustering solution provided by the k-means algorithm can be selected as the best representative of all the clustering methods. At this point it should be stressed that similarity is not being used interchangeably with correctness. The central idea here being that if no information exists on the nature of the underlying data[45] then it is not inherently clear which clustering method should be used. This is owing to the fact that the performance[46] of a clustering method is very closely linked to the distribution of the underlying data. When looking at the **Total** column in Table 7, a value of 61.23% for the k-means algorithm does not imply that 61.23% of the assignments made by this algorithm are correct. Rather this value implies that on average 61.23% of the assignments made by the k-means are the same or similar to the assignments made by the remaining 10 algorithms. This would make the k-means algorithm results desirable assignments to test using the *STRING* database as it is not feasible or useful to test all methods. The rationale for which will be further outlined in Section 5.

The k-means algorithm producing results similar to the Gaussian Mixture model and Fuzzy clustering is due to the fact that the k-means algorithm is simply a special case of the latter clustering methods. But by presenting the totals as percentages, valuable information can be gleaned regarding the degree of similarity that exists between the k-means algorithm and Mean Shift clustering, DBSCAN and Hierarchical clustering with single linking.
While the k-means method results was used in the biological validation steps, it needs be noted that there is very little difference between the **total** for the k-means algorithm and the Affinity Propagation **total**(4th highest total). The potential of this being that if the k-means method does not prove to be a suitable solution, there may not be much of an improvement produced by testing the results from Fuzzy Clustering, Gaussian Mixture Model, Wards Minimum Variance and Affinity Propagation. The reason for this being, that the difference in clustering assignments of these methods is marginal according to the similarity indices provided. While even these slight differences in clustering may produce significantly different results when undergoing the process outlined in Section 5 (further outlined in Section 5.2.1). If the results from the k-means method prove to be significantly incorrect or lacking in any use then there may be more value in looking at the clusters produced by Complete Linkage, Average Linkage or Spectral Clusters.

[45] This extends to both the actual distribution of the data as well as the prevailing biological mechanism for immunity or susceptibility which produced the differential expression patterns for both hosts at 12, 32 and 67 days-post-infection.
[46] Performance here being defined as the clustering methods ability to correctly and accurately cluster the given data.

5 Interpreting Biological Validity of Results

To ascertain whether the clusters produced by the k-means algorithm do in fact group together genes that are functionally related[47] or co-expressed, the interactions or relation between these genes need to be examined. Insight into these interactions can be achieved not by looking at how the genes themselves interact with one another, but rather at how their related proteins interact. These protein-protein interactions provide a view into the machinery of the cell.

The STRING database is a collection of the known and predicted Protein-Protein Interactions(PPI) that exist for every known genome [36]. These interactions can either be direct(physical interactions) or they can be functional associations. These interactions come from the aggregation of information from other primary databases[48], the transference of knowledge between organisms and from computational prediction. This information has been gathered and aggregated from automated text-mining, conserved co-expression, high-throughput lab experiments,genomic context predictions and the previous knowledge that exists in databases to provide interactions between 24 584 628 proteins in 5 090 organisms.

The data in this project originated form the Cassava plant. While there has been a considerable amount of work done on the sequencing of Cassava genes[49], only 69% of these genes have known functional annotations. In contrast, the Arabidopsis plant is the most studied as it was the first plant to be sequenced. This plant therefore can act as a suitable reference genome with which the Cassava plant can be compared against.

While *Arabidopsis thaliana* can act as a suitable reference genome for the Cassava plant, these plants along with their genomes are not interchangeable. This distinction is important when attempting to uncover the mechanism for immunity in a plant or organism. There are gene homologs that exist between the plants but they do not necessarily have a similar mechanism for host immunity. This produces a problem in deciding on a clustering method by testing it against one organism and then applying this method to another organism. One reason for this being that the performance of a clustering method is very closely linked to the nature of the underlying data(the specifics of which can be found in Table 2). With no guarantee that the *Arabidopsis thaliana* underlying distribution(produced by the specific host immunity mechanism) matches the Cassava plants', then it is not as simple as comparing the two. Furthermore, if this were to be attempted then one would need data for both organisms originating from an incredibly similar experimental protocol. Meaning that an *Arabidopsis thaliana* host that is susceptible to a virus and a host that is tolerant to the virus would be needed and these hosts would need to undergo the same protocol that has been outlined in Figure 70a.

It has been established that the Cassava plant and *Arabidopsis thaliana* have homologous genes[50] but this relation is not exhaustive. This is to say that there is not a matching gene in the *Arabidopsis thaliana* genome for every gene in the Cassava plant genome. This is detailed in Table 9 where it is shown that for the 6117 genes containing a differential expression in both the tolerant and susceptible host, 5460 genes had an *Arabidopsis thaliana* homolog. This alone produces at least a 10% disparity between the plants.

Furthermore, while functional annotations[51] for the Cassava plant can be partly inferred by the functional annotations of the *Arabidopsis thaliana* genome (due do homolog genes existing between these plants) this alone does not provide a mechanism for immunity. Immunity is an incredibly complex process. In order to discover the exact process which produces immunity in a plant, the protein/gene functions alone are not sufficient. Instead the biological process or pathway that a protein is involved in needs to be uncovered. Once provided with this knowledge, then the differential expression of the gene that encodes for this protein can offer insight into how the virus impacts this process/pathway or how this process/pathway contributed to host immunity[52]. This being said,uncovering the biological process or pathway that a protein is involved in is no simple task. This is because a single gene may encode for a single protein which has one function, but this protein can act in multiple processes/pathways.

[47] Functional relation here is relating to the assumption that genes which have a similar expression have a common underlying mechanism.

[48] Protein sequence databases or plant genome DNA sequence database. The sequences in these databases aid in the determination of the gene or protein that is being mapped as well as the alignment of the experimental/reference sequences.

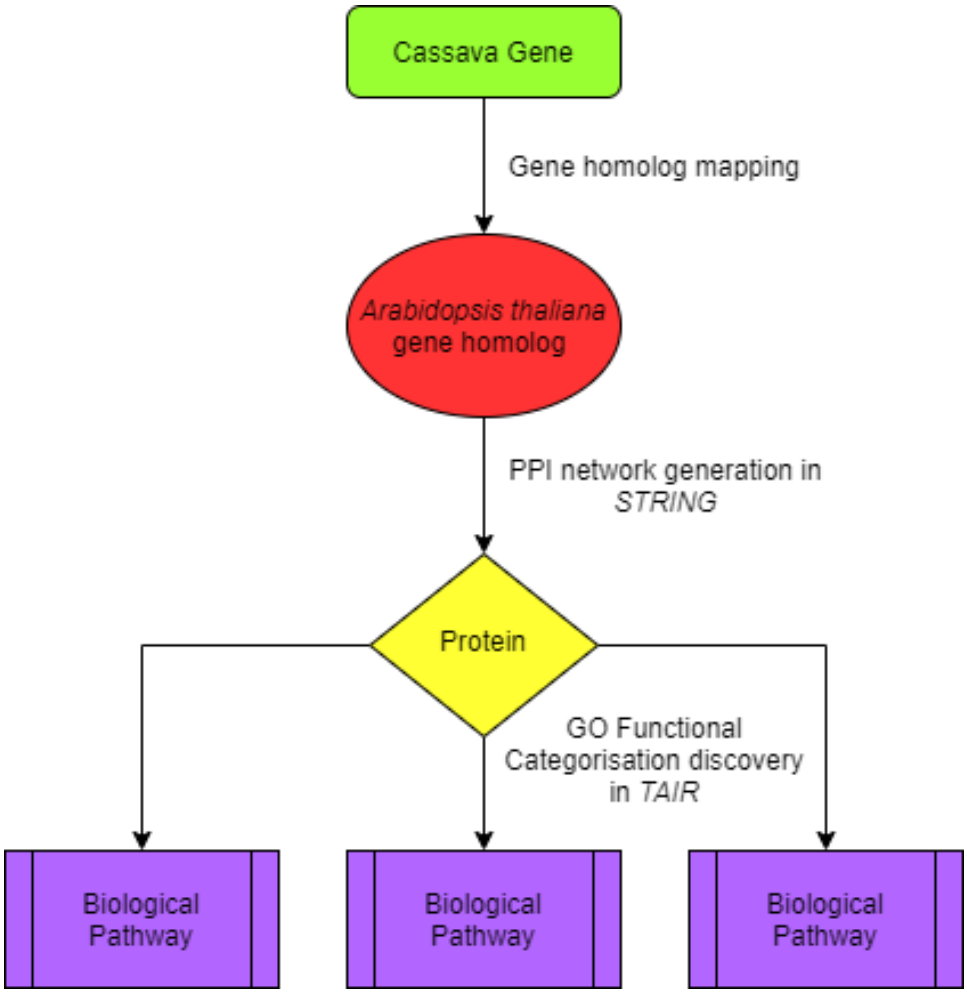
[49] Which can be found in *Phytozome* along with sequencing information of numerous other plants.

[50] Resulting from these plants having a common ancestor.

[51] Provides the function of the protein that is encoded by a gene.

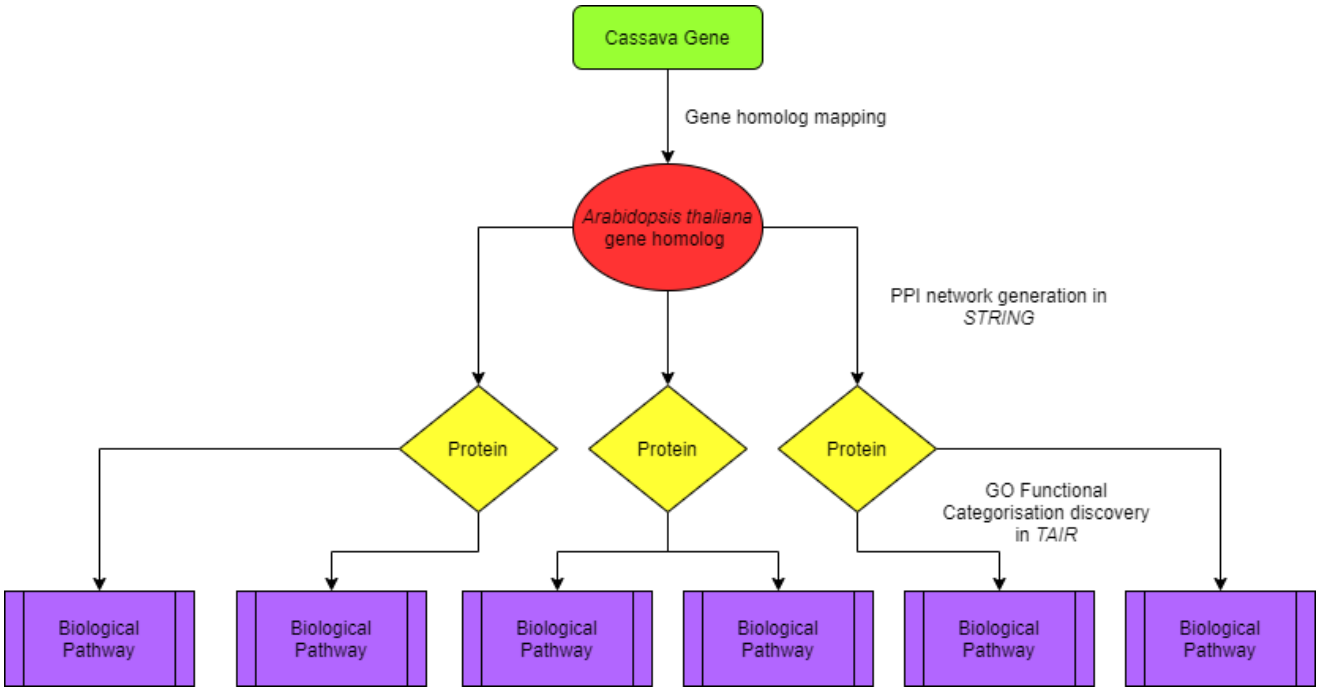
[52] This would be seen by looking at whether the gene is up-regulated or down-regulated in both the tolerant and susceptible host.

Additionally a single gene may encode for multiple proteins which then each have a function that may be present in multiple processes/pathways. These scenarios have been illustrated in Figure 68a and 69a below.



(a)

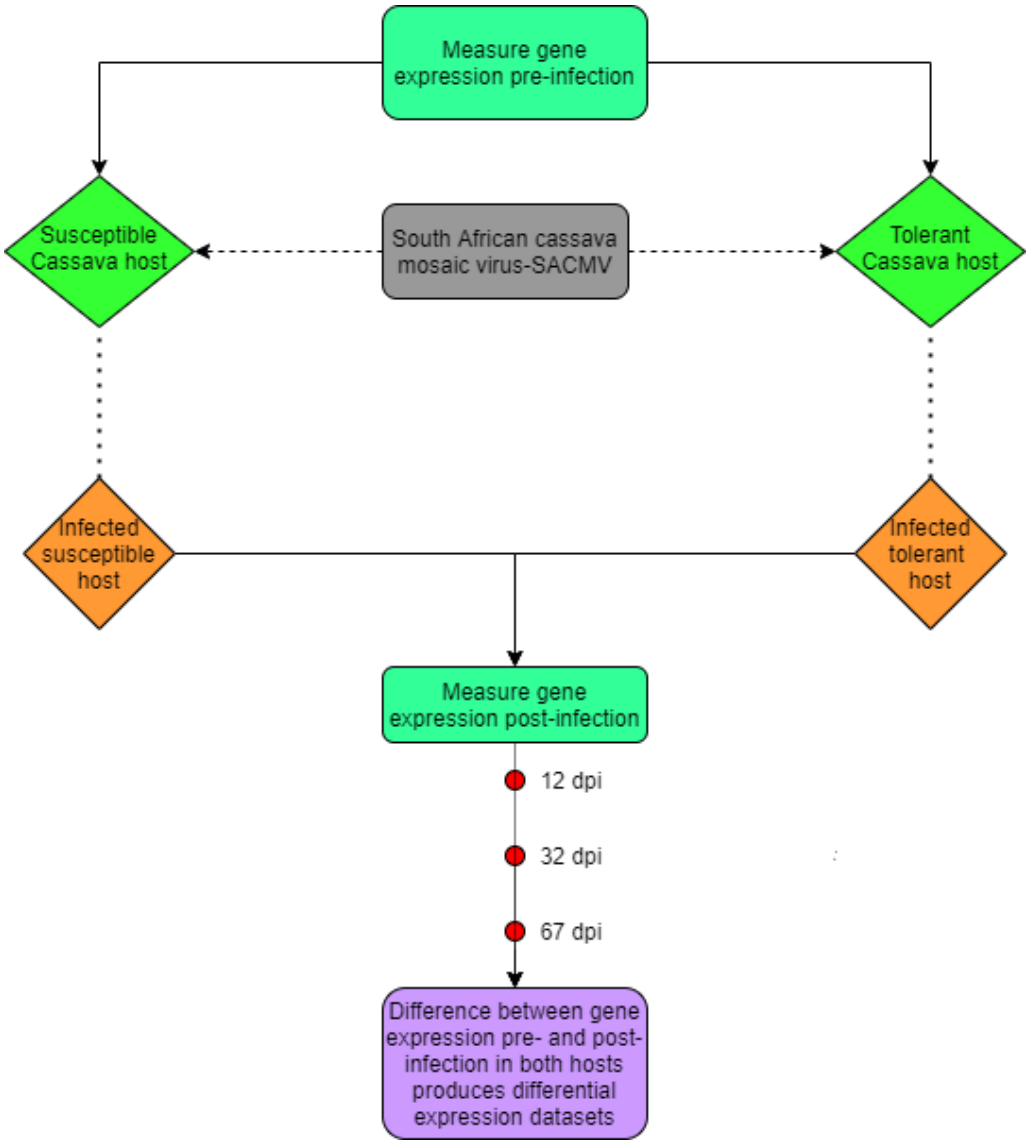
Figure 68: A visual representation of the multiple biological pathways that a protein may be involved in.



(a)

Figure 69: A visual representation of the multiple proteins that can be encoded by one gene and the subsequent biological pathways that these proteins may be involved in.

If the discovery of gene function or the functional annotation of the protein encoded for by the gene were sufficient in understanding host immunity. Then a reasonable starting point would simply be to look at the functional annotations of the gene homologs existing within *Arabidopsis thaliana* and ascribe them to the matching Cassava genes.
However, to begin to understand the mechanism for immunity one would need to investigate the biological pathway in which a protein is involved in. Furthermore a comparative analysis of the protein expression between a susceptible and tolerant host, and therefore the prevailing affect on the related biological pathway, provides insight into the mechanisms that confer immunity. The initial step being to identify the differentially expressed genes in a tolerant and susceptible host. This is done by measuring the gene expression of a tolerant as well as a susceptible host pre-infection and post-infection. The difference between these pre- and post-infection expression levels in each host provides a differential expression profile for each host, based on changes induced by the viral infection. This process is broadly summarised in Figure 70a below.



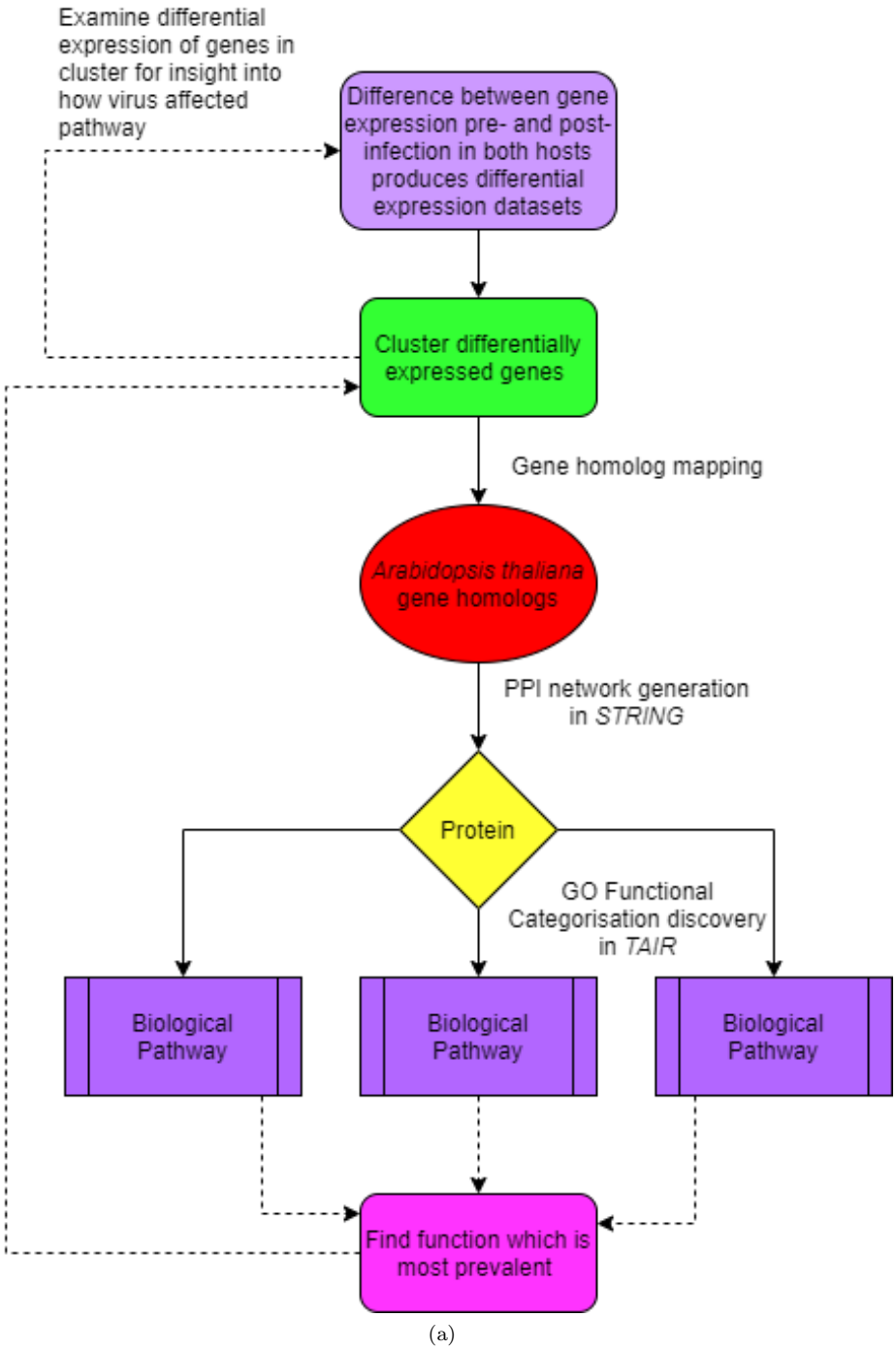
(a)

Figure 70: A summarised view of the process required to produce differential gene expression data.

Once these differentially expressed genes have been identified then it becomes a question of isolating genes which have a similar differential expression. This was done by the use of the clustering algorithms outlined in Section 3, with the results being presented in Section 4.
The next step in the process of identifying the mechanism for host immunity was to make biological sense of the results obtained. This was done only for the results produced by the k-means algorithm. The core reasons for this being:

- It would take a not inconsiderable amount of time to produce PPI networks in STRING for each clustering method. Furthermore this particular part of the process was contributed by a member of the MCB department, which therefore made it more feasible to submit a solution which was of the most interest. This solution being the result produced by the k-means algorithm.
- The relation between gene, protein and biological pathway outlined in Figures 68a, 69a and 71a would produce

seemingly different results for clusters from 2 different methods, even when these clusters differed by only a small percentage of genes. All of which would have to be experimentally validated before a single method could be selected as optimal. This will be further explained following the presentation of the results obtained from The Arabidopsis Information Resource(*TAIR*) in Section 5.2.1.



(a)

Figure 71: Proposed process for discovering the mechanism for immunity.

The genes from the Cassava plant[53] were mapped to *Arabidopsis thaliana* homologous genes. Table 8 depicts a sample of this mapping which was then inserted into the STRING database to discover the protein-protein interactions that exists within each cluster.

The gene homologue mapping between *Arabidopsis thaliana* and the Cassava plant along with the PPI network generation in *STRING* was provided by Warren Freeborough from the Molecular and Cell Biology(MCB) department. One such PPI network produced in STRING is shown in Figure 72a.

Table 8: An example of the gene homologue mapping between Cassava and *Arabidopsis thaliana*

Cassava	***Arabidopsis thaliana***
Manes.18G136900	AT5G58410
Manes.02G113700	AT2G45440
Manes.03G079300	AT1G53490
Manes.03G079300	AT1G53490
Manes.14G114500	AT4G25810
Manes.09G025800	AT3G14470
Manes.02G201000	AT2G45360
Manes.11G051500	AT1G08510

Table 9: The total number of genes per dataset where *Arabidopsis thaliana* gene homologs could be found for the Cassava genes.

Dataset	**Quadrant**	**No.Cassava Genes**	**No.*Arabidopsis thaliana* homologs**	**Difference**
B12	1	51	39	12
	2	591	478	113
	3	912	781	131
	4	133	129	4
B32	1	617	559	58
	2	399	361	38
	3	585	511	74
	4	1000	935	65
B67	1	319	273	46
	2	419	405	14
	3	470	423	47
	4	621	566	55
Totals		**6117**	**5460**	**657**

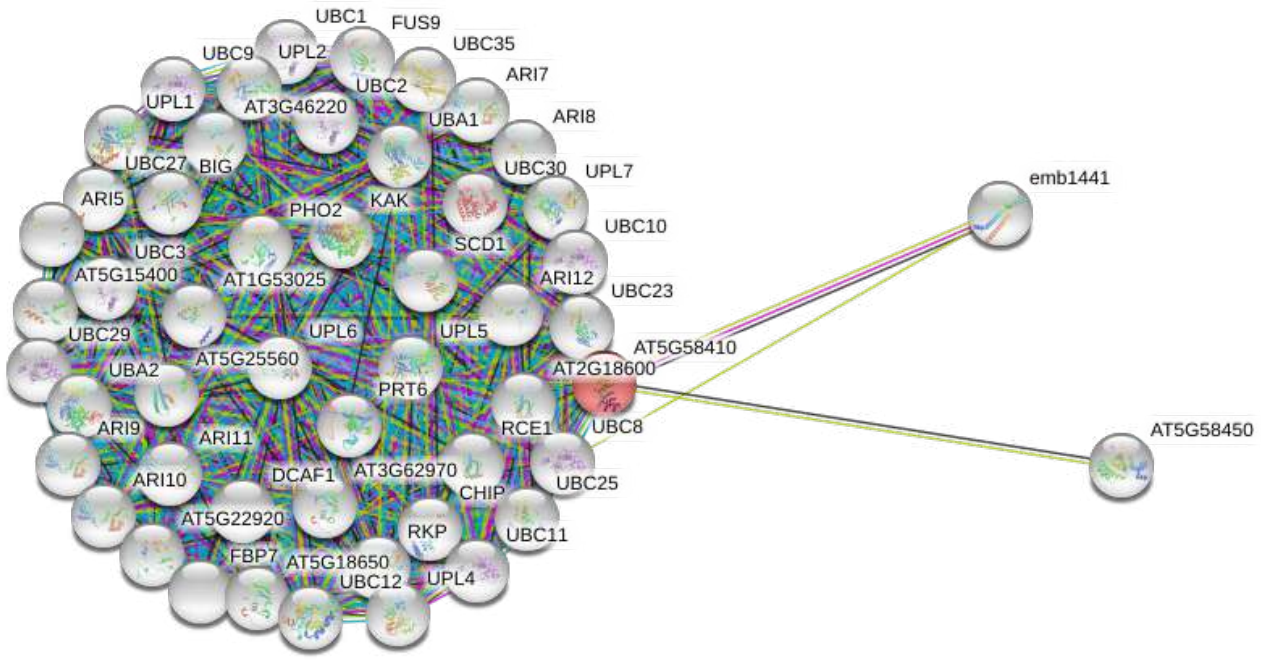
Table 9 above provides some insight into the limitations that exists when mapping the Cassava genes to *Arabidopsis thaliana* gene homologs. The totals presented above show that there is at least a 10% loss of information produced in the mapping process.

[53]There are some experimentally determined gene functions for the Cassava plant.

These values are presented here as they provide some indication as to the uncertainty that is introduced into the system when trying to relate the biological pathway/process of a protein back to the gene that these proteins are encoded by. This uncertainty that exists when mapping Cassava and *Arabidopsis thaliana* gene homologs, relating proteins to these homologs and then trying to ascribe a functional pathway to these proteins means that it can be quite difficult to create a metric to compare clustering methods after the STRING PPI network generation and *TAIR* process(briefly outlined in Figure 71a above) as opposed to before.

5.1 PPI Networks and the STRING Database

The PPI network in Figure 72a below was created by inserting the relevant *Arabidopsis thaliana* gene homologs into STRING with a minimum required interaction of 0.4 (medium confidence),no maximum set on the 1st shell, a max number of 50 for the 2nd shell and the resulting network rendered in evidence mode.



(a) PPI network produced by STRING for Cluster 0 In Quadrant 1 of B12 dataset

Figure 72: PPI network produced in *STRING* illustrating the interactions between the proteins related to the genes that exist in the cluster labelled 0 within Quadrant 1 of the B12 dataset.

A medium confidence threshold on the minimum required interactions means that only protein-protein interactions above a value of 0.4 would be included in the resulting network. The options for this value include low confidence (0.15), high confidence (0.7) and highest confidence (0.9). While low confidence would result in more interactions, it would also increase the likelihood of false positives. High or highest confidence on the other-hand, while providing only the maximum protein-protein interactions, may result in the exclusion of lower-value yet insightful/useful interactions/relations. With this in mind, a medium confidence threshold would provide a suitable middle-ground between accuracy and total information on the interactions that exist in the network.

The 1st shell provides direct interactions that are directly connected with the genes that have been used as inputs. This value was set to 0, which while increasing the resultant visual complexity of the network, provides all possible interactions that exist with the genes that are being studied. The 2nd shell provides the indirect protein interactions that exist with the proteins in the 1st shell. This value could also be set to zero which would produce a network

that only includes the direct connections with proteins in the 1st shell. Setting this value to 50[54] would allow the inclusion of these indirect connections in the network, thus providing more information on the protein interactions that exist within the relevant cluster. While this parameter may provide more enriched information regarding the interactions that exist within the PPI network(by allowing for the inclusion of protein interactions between genes in the cluster and proteins in the second shell), it does increase the variable nature of the PPI networks for a cluster.

Consider again Figure 71a. The loss of information seen between the mapping of Cassava genes(green rectangle) and their *Arabidopsis thaliana* gene homologs(red ellipsoid) has already been explained. But it is here in the PPI generation using the STRING database that another level of complexity is introduced(process represented by movement from red ellipsoid to yellow square in Figure 71a).
Consider two clusters from two different clustering methods which differ by a small number of genes[55]. By allowing for a maximum number of 50 interactions between proteins in the first shell(proteins encoded for by genes in the cluster) and proteins in the second shell(proteins not encoded for by genes in the cluster), the PPI networks for these clusters could be significantly different. These networks alone would then be difficult to compare, making this part of the process a none-viable point for the comparison of results from different clustering methods. Furthermore a PPI network without the relevant biological pathway/process would not provide insight into the mechanism for immunity. It is for this reason that the investigation cannot stop at PPI network generation. It would need to progress to the identification of related biological pathways/processes, as represented by the transition from the yellow square to the purple rectangles in Figure 71a.

A PPI network rendered in evidence mode produces connections between proteins that indicate the type of evidence that exist for the interaction. The different types of lines/connections that can exist include:

- Red line - fusion evidence is present
- Green line - neighbourhood evidence
- Blue line - concurrence evidence
- Purple line - experimental evidence
- Yellow line - text-mining evidence
- Light blue line - database evidence
- Black line - co-expression evidence

For this network, all active interaction sources where selected, therefore utilising all possible information that exists for these protein-protein interactions. The STRING database takes all these inputs to generate a network image using a spring model. In a spring model the nodes(proteins) are modelled as masses while the edges/connections act as springs. The final position of these nodes and edges in the graph is produced by minimising the energy of the entire system. This is done by increasing the strength of the spring of high confidence edges, which will allow these edges to find an optimal position before the edges with lower confidence and therefore lower spring strength.
When interpreting the resulting network it is important to note that the physical distance between any two nodes does not indicate the strength of the connection. While the algorithm does attempt to place nodes that are connected with high confidence closer to each other, the default of this value is only 80% of what would be the edge's normal length. Furthermore, colourful nodes are the input proteins as well as 1st shell interactions while white nodes show the 2nd shell of interactions.
The network in Figure 72a contains far more interactions than would be expected between randomly selected proteins in a genome. But these proteins were not randomly selected, instead they were deemed to be similar in expression profile by the k-means clustering algorithm. While there are at least 50 possibly irrelevant interactions that were added to the network by setting the maximum number of interactions in the 2nd shell to 50, this value is

[54]These parameters were set and selected by the collaborators in the MCB department.
[55]This is to say that the clusters differ by a small percentage of the total genes in each cluster.

markedly smaller than the total number of edges in the network which amounts to just over 1000. It can therefore be concluded from this network that the group of proteins that were used as inputs are partially connected in terms of biological function.

Evidence of a functional biological connection between the proteins that exist in the clusters at this stage of the analysis, provides some validation for the use of k-means clustering in this context. However, more information is needed on the nature of this biological connection if the k-means algorithm is to be considered a suitable analysis tool for these datasets.

5.2 The Arabidopsis Information Resource(*TAIR*)

More robust insight into the evaluation of the suitability of the clustering method chosen, could be to compare functional gene ontology (GO) category enrichment of the genes in the STRING network for each of the clusters. This can be done by using the functional enrichment in the PPI network which provides information on the ontologies of the proteins. This output of STRING shows the more enriched terms in the proteins that were used to create the network. Table 10 presents an example of the Biological Process Gene Ontologies for the network in Figure 72a. The Gene Ontologies in this network are each paired with a description of their term along with the proteins in the network that match these terms.

Table 10: Gene Ontologies for PPI Network in Figure 72a

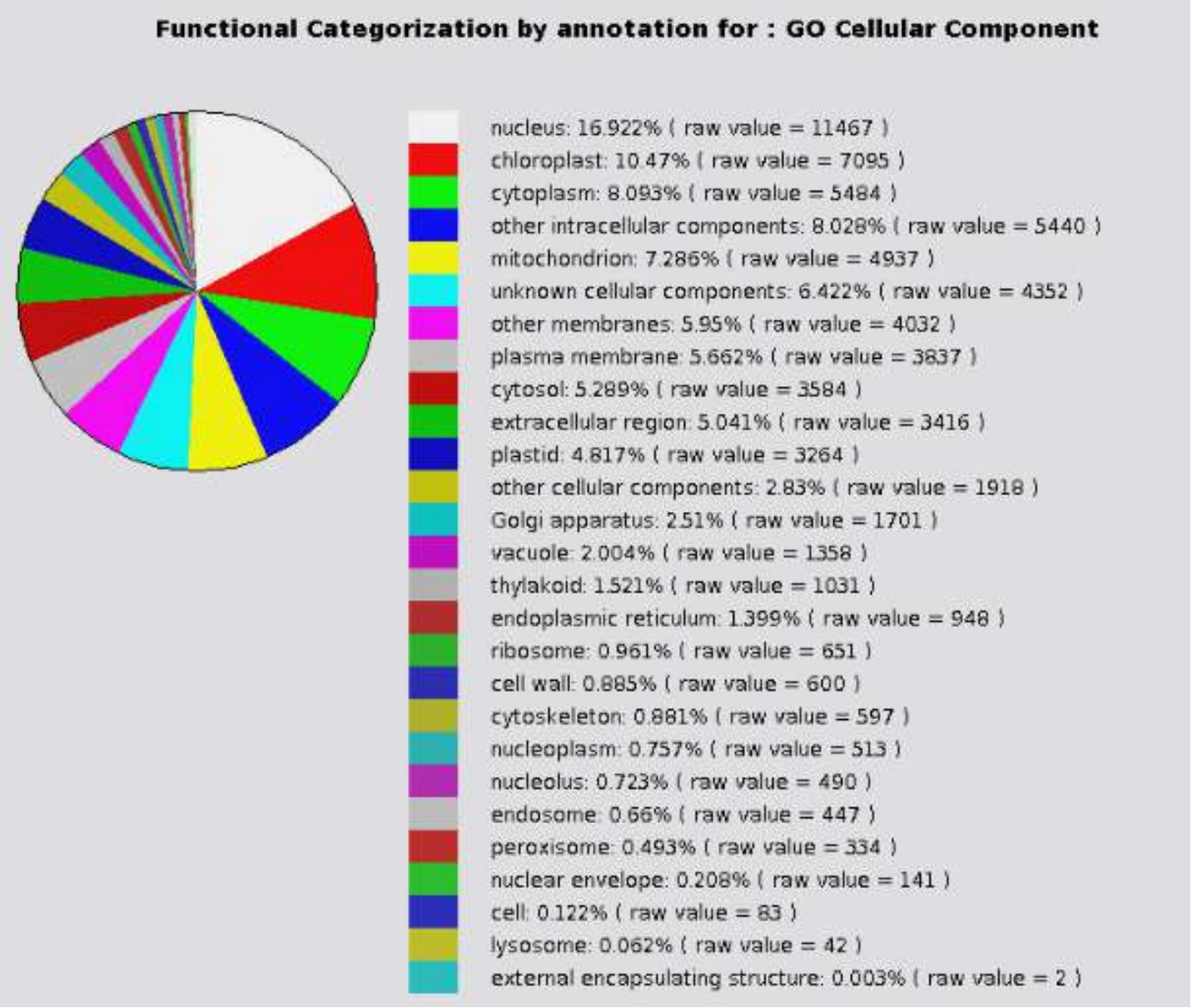
#term ID	term description	observed gene count	background gene count	false discovery rate	matching proteins in your network (IDs)
GO:0032446	protein modification by small protein conjugation	44	723	4.18E-59	AT1G05880.2, AT1G05890.2, AT1G14400.1, AT1G21760.1, AT1G53025.1, AT1G55860.1, AT1G65430.1, AT1G70320.1, AT1G78870.1, AT2G02760.1, AT2G16740.1, AT2G16920.1, AT2G18600.1, AT2G22010.2, AT2G30110.1, AT2G31510.1, AT2G31760.1, AT2G31770.1, AT2G31780.1
GO:0016567	protein ubiquitination	41	696	1.45E-53	AT1G05880.2, AT1G05890.2, AT1G14400.1, AT1G21760.1, AT1G53025.1, AT1G55860.1, AT1G65430.1, AT1G70320.1, AT1G78870.1, AT2G02760.1, AT2G16740.1, AT2G16920.1, AT2G22010.2, AT2G30110.1, AT2G31510.1, AT2G31760.1, AT2G31770.1, AT2G31780.1, AT2G33770.1, AT3G07370.1

This list of proteins matching the GO terms in the network were inserted into the TAIR(The Arabidopsis Information Resource) GO annotation search and functional categorization as locus identifiers. This tool groups the proteins in the STRING network into three broad functional categories that are based on GO hierarchy and the high level terms therein. These broad categories are:

- GO Cellular Component

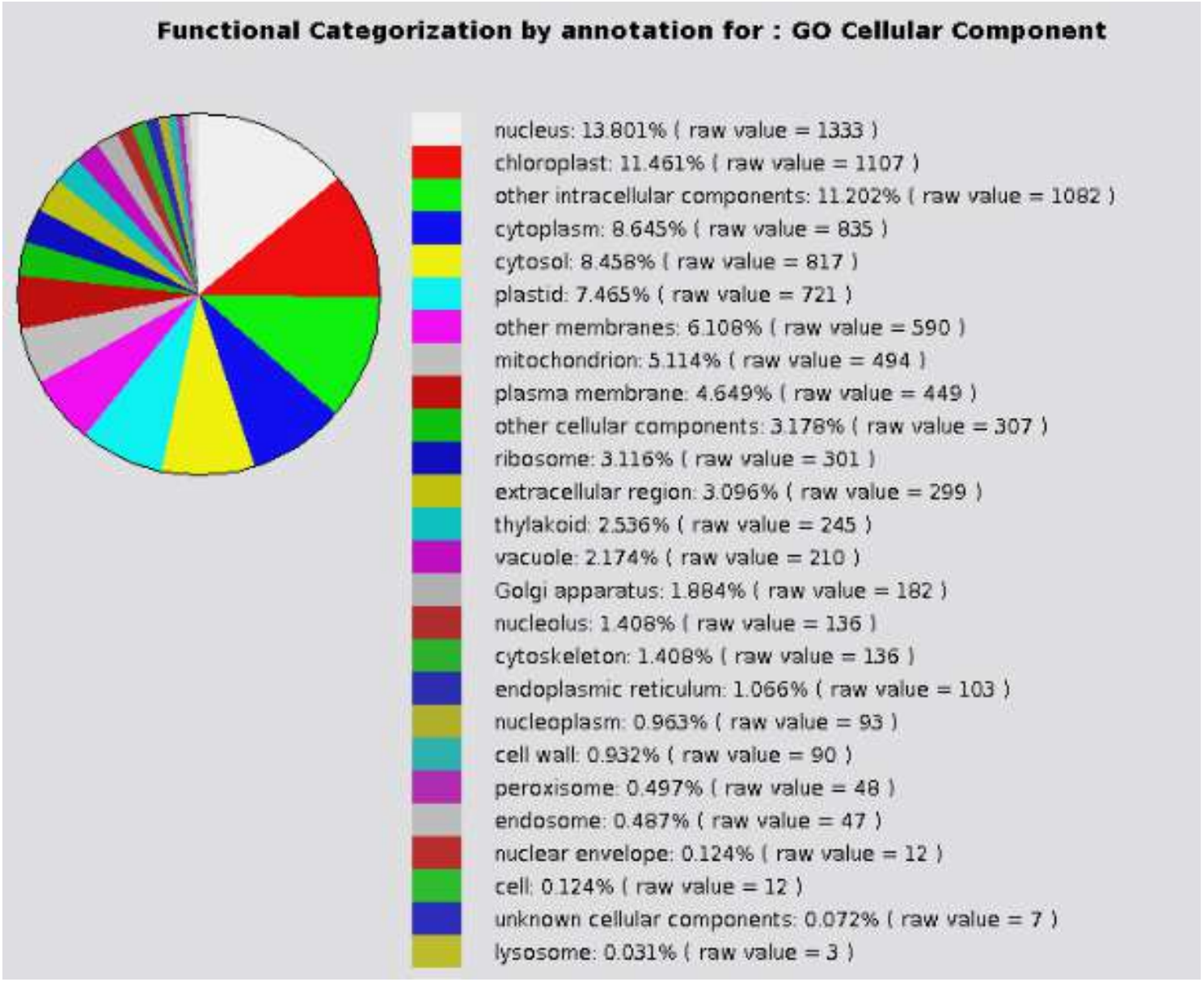
- GO Biological Process
- GO Molecular Function

Each of these broad categories contain further divisions, the relative proportions of which can be visualised using pie-charts as depicted in Figure 73a - 78a.



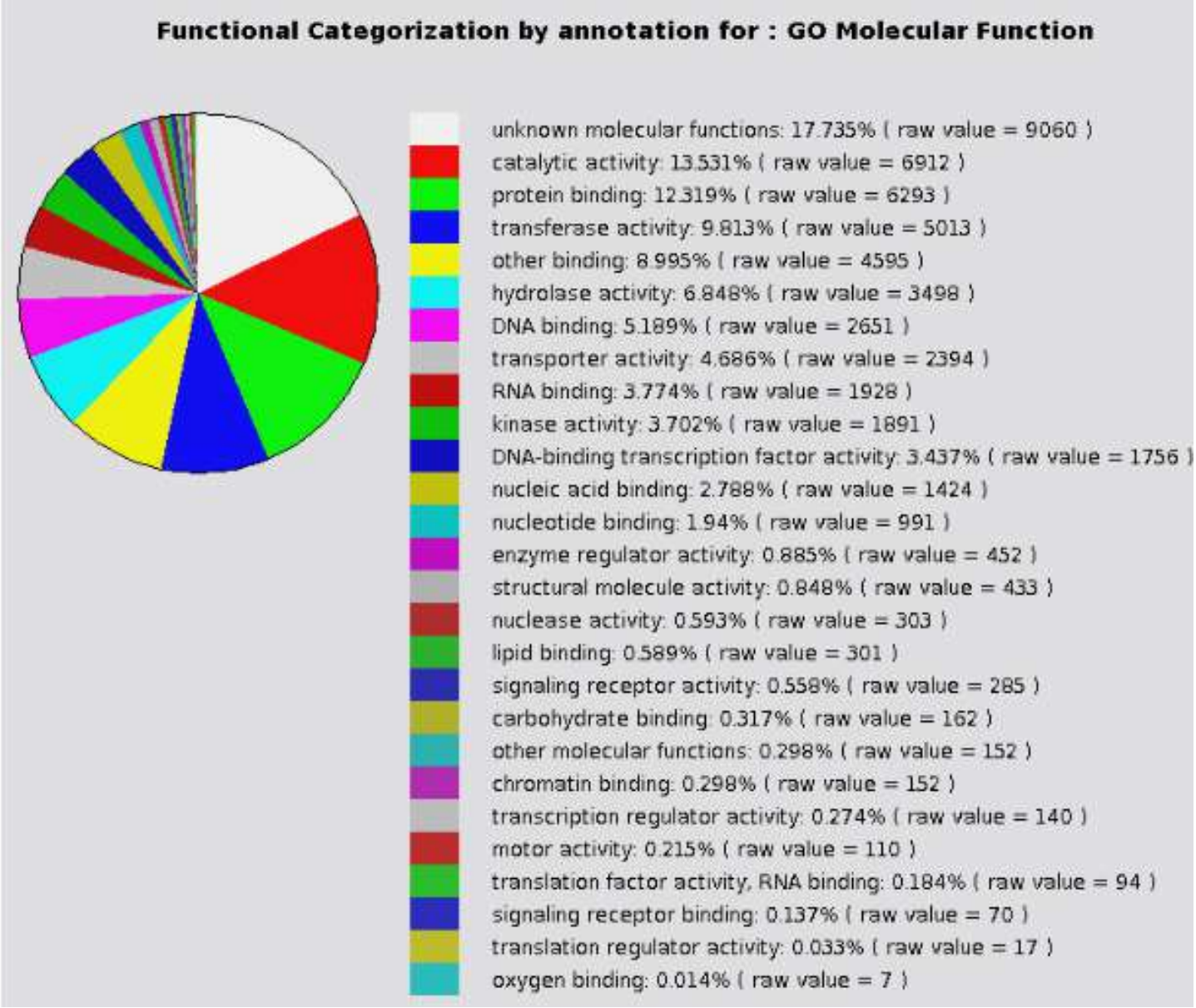
(a) GO Cellular Component Functional Catgorisation for Whole Genome of *Arabidopsis thaliana*

Figure 73: Pie chart depicting the functional categorisations (given as percentages) of GO Cellular Components when using the GO terms for all genes in the *Arabidopsis thaliana* genome.



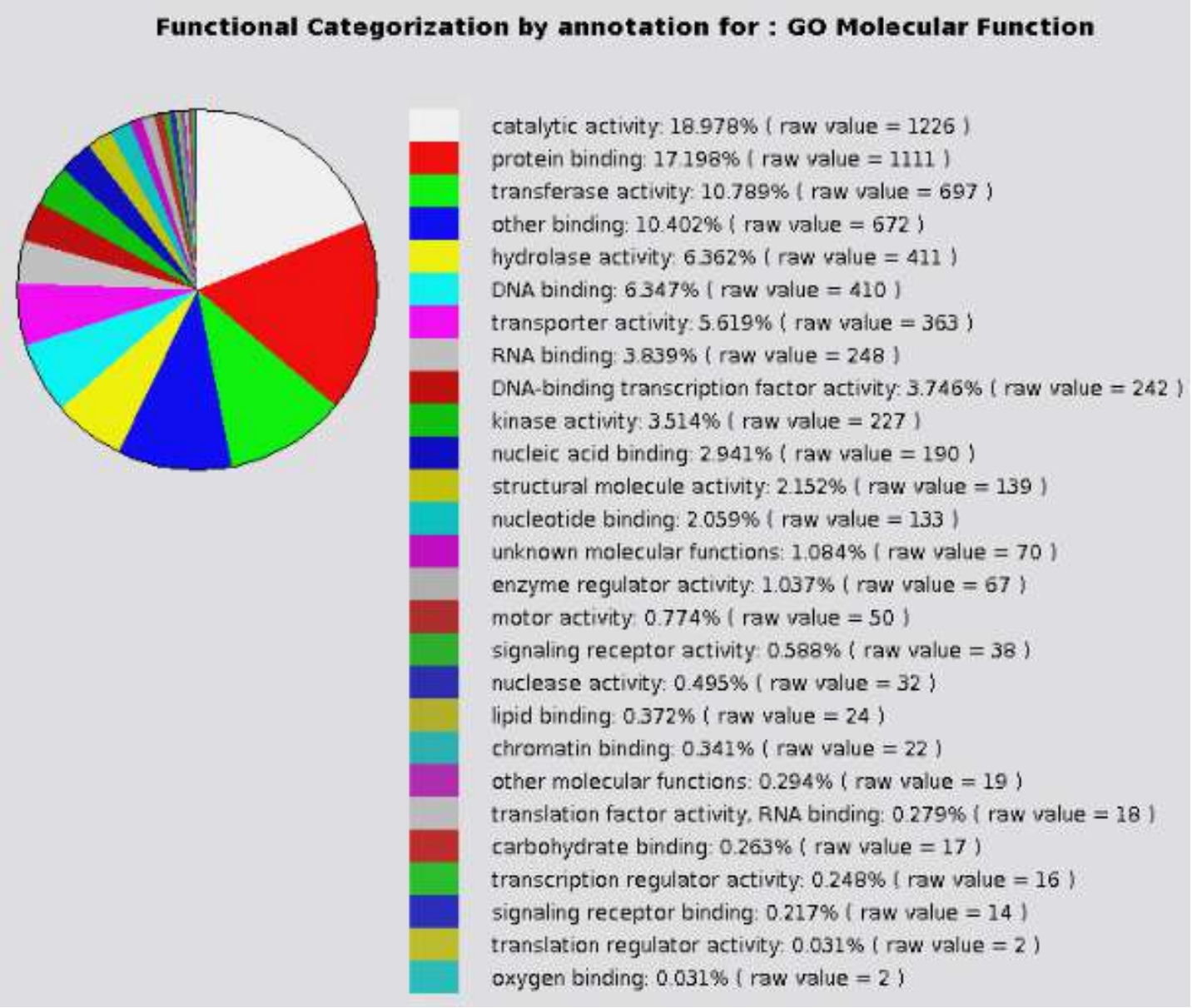
(a) GO Cellular Component Functional Catgorisation for B12 Dataset

Figure 74: Pie chart depicting the functional categorisations (given as percentages) of GO Cellular Components when using the GO terms for all genes in the B12 dataset that had a *Arabidopsis thaliana* gene homolog.



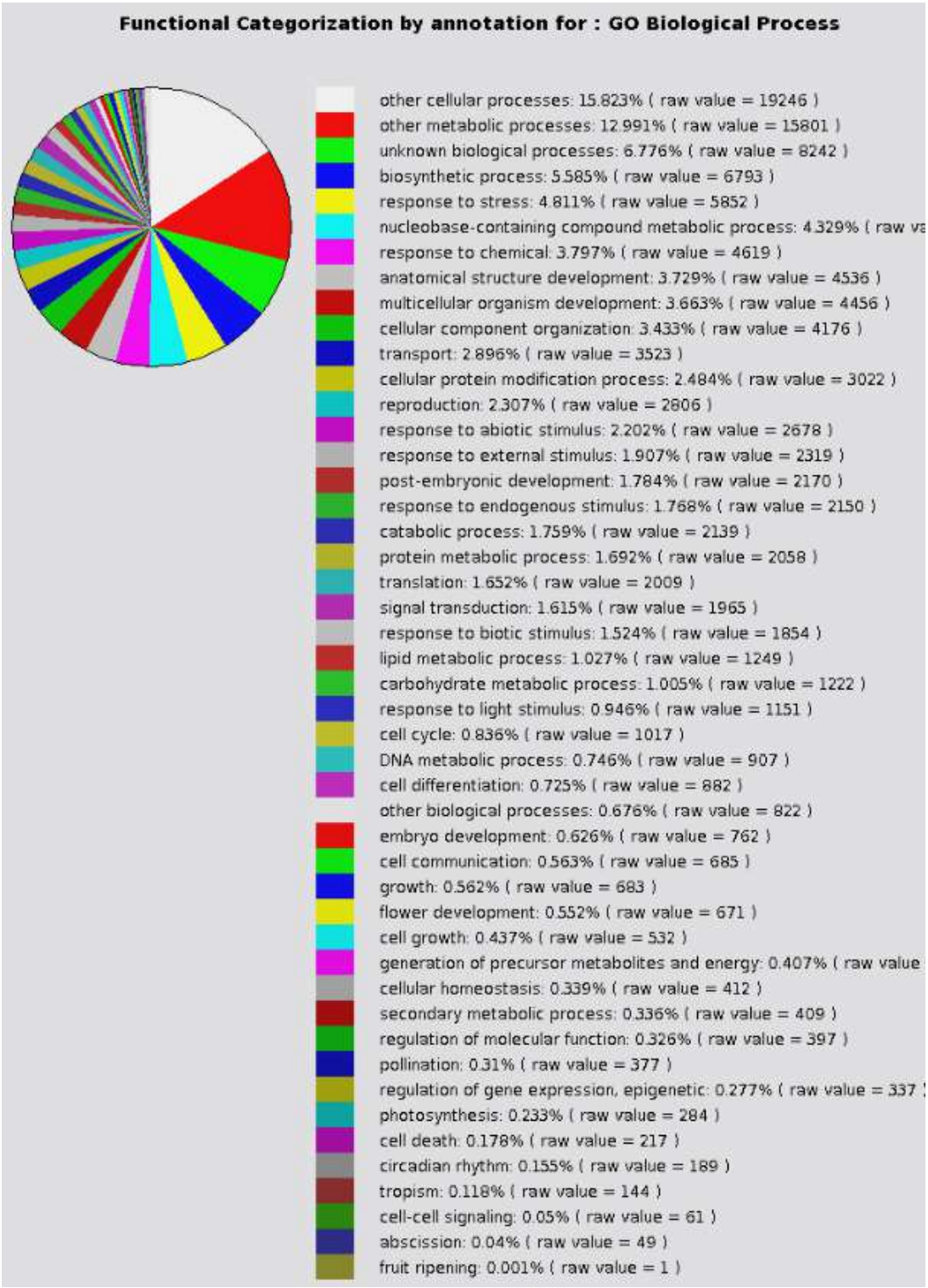
(a) GO Molecular Function Functional Catgorisation for Whole Genome of *Arabidopsis thaliana*

Figure 75: Pie chart depicting the functional categorisations (given as percentages) of GO Molecular Functions when using the GO terms for all genes in the *Arabidopsis thaliana* genome.



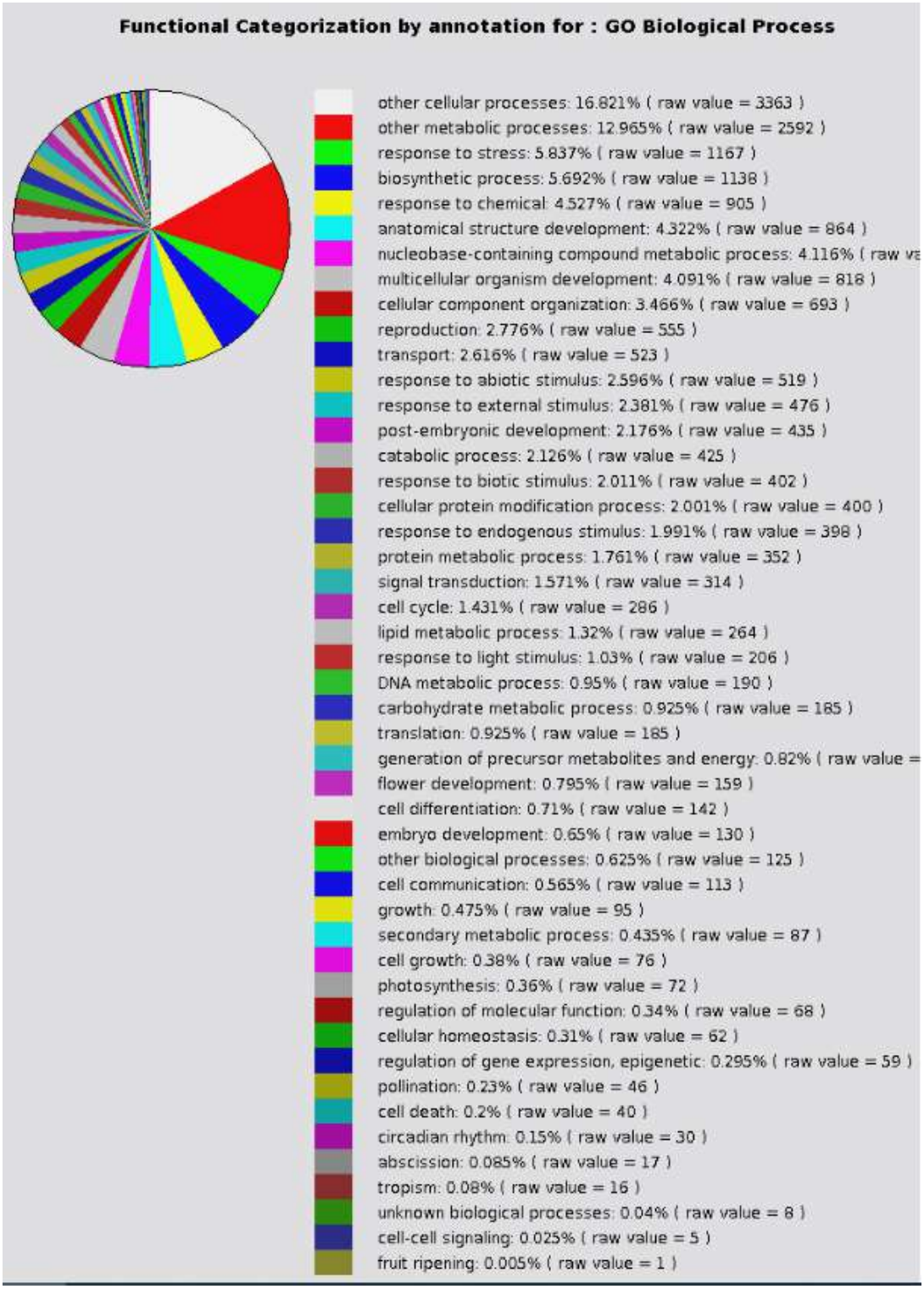
(a) GO Molecular Function Functional Catgorisation for B12 Dataset

Figure 76: Pie chart depicting the functional categorisations (given as percentages) of GO Molecular Functions when using the GO terms for all genes in the B12 dataset that had a *Arabidopsis thaliana* gene homolog.



(a) GO Biological Process Functional Catgorisation for Whole Genome of *Arabidopsis thaliana*

Figure 77: Pie chart depicting the functional categorisations (given as percentages) of GO Biological Processes when using the GO terms for all genes in the *Arabidopsis thaliana* genome.



(a) GO Biological Process Functional Catgorisation for B12 Dataset

Figure 78: Pie chart depicting the functional categorisations (given as percentages) of GO Biological Processes when using the GO terms for all genes in the B12 dataset that had a *Arabidopsis thaliana* gene homolog.

The pie-charts in Figure 73a, 75a and 77a are for the whole genome of *Arabidopsis thaliana* and show the proportional split for the GO Cellular Component, GO Molecular Function and GO Biological Process functional categorisations respectively. Figure 74a, 76a and 78a show the functional categorisations of the GO annotations for the whole B12 dataset. While the actual values for each functional categorisation in the B12 dataset do not correlate exactly with the whole genome of *Arabidopsis thaliana*, the general divisions and proportions are sufficiently similar. A possible explanation of the slight discrepancy being attributed to the exclusion of genes with insignificant differential expressions in Section 3.1 along with the lack of *Arabidopsis thaliana* gene homologs for each Cassava gene in the dataset.
The values in Figures 76a and 78a[56] were compared with the functional categorisations and proportion split for each cluster in the B12, B32 and B67 datasets. Table 11 shows the results for GO Molecular Function functional categorisations and Table 12 shows the categorisations for GO Biological Process for the clusters in Quadrant 1 of the B12 dataset[57].

Table 11: GO Molecular Function categorisations for clusters in the B12 Dataset - Quadrant 1.

Molecular Function	B12	C0	C2	C3	C4	C5	C8	C9	C11	C12	C13
carbohydrate binding	0.3	0.5									0.1
catalytic activity	17	33	21	42	13	29	16	10	4	20	33
chromatin binding	0.3		1								
DNA - binding transcription factor activity	3.1			1		1			1		
DNA binding	6.3		5	1		12			2		
enzyme regulator activity	1				1	0.1	3	1			
hydrolase activity	5.7		10	1	7	4	6	2	17	2	
kinase activity	3.3		5		5	2	5			22	1
lipid binding	0.4				1						
motor activity	0.6								14		
nuclease activity	0.4		3	1							
nucleic acid binding	2.4		1			3					
nucleotide binding	5.9	9.3	13	6	8	2	9	3	10	9	14
other binding	17	15	25	16	32	7	26	8	13	11	15
other molecular functions	0.3										
oxygen binding	0.1						0.1				
protein binding	14	10	6	6	7	12	10	2	19	15	5
RNA binding	3.7		2	1		1	0.1		7		
signalling receptor activity	0.5				1		1			4	
signalling receptor binding	0.2						1				
structural molecule activity	1.8				3	0.1					
transcription regulator activity	0.2					0.1					
transferase activity	10	32	7	26	5	25	10		14	18	32
translation factor activity, RNA binding	0.3		1				0.1				
translation regulator activity	0.1										
transporter activity	4.7				17	0.1	12	75			
unknown molecular functions	0.9		1								

Column titled B12 contains the percentages of the genes that are involved in each Molecular Function subcategory - as denoted by the Gene Ontology - for the B12 dataset. The columns titled C0 to C13 contain the percentages of genes that are involved in each Molecular Function subcategory - as denoted by the Gene Ontology - in each cluster of Quadrant 1. The colour scale used ranges from red to blue, with blue cells containing higher values and red cells containing lower values.

[56] GO Cellular Component Analysis was excluded as the resulting categories are far too broad for this stage of the analysis. The time expenditure in doing this analysis would far outweigh the information produced. If the results outlined in the following sections do in fact prove useful in the discovery of host immunity, then the GO Cellular Component analysis can be done at a future stage to augment these results.

[57] Cluster 10(C10) is not included as there were no *Arabidopsis thaliana* gene homologs found for the Cassava genes in this cluster.

Table 12: GO Biological Process categorisations for clusters in the B12 Dataset - Quadrant 1.

Biological Process	B12	C0	C2	C3	C4	C5	C8	C9	C11	C12	C13
abscission	0.1				0.1						
anatomical structure development	4	4	1	2	4	1	4	5	6	4	2
biosynthetic process	5	1	11	12	7	14	3	0.1	4		0.1
carbohydrate metabolic process	0.9		0.1	1			1				
catabolic process	2.2	12	1	2	4	1	2	0.1			14
cell communication	0.5	0.1		0.1			0.1		1		1
cell cycle	1.8	1	8				0.1		1		1
cell death	0.3				0.1		1	0.1	0.1		
cell differentiation	0.8	0.1				0.1	1	1	0.1	2	0.1
cell growth	0.4	0.1				0.1	1	1	0.1	2	
cell-cell signalling	0.1										
cellular component organization	3.6	2	8	1	5	2	5	4	4	2	2
cellular homeostasis	0.3				2			2			
cellular protein modification process	2.1	12	0.1	1	2	1	1		2	14	14
circadian rhythm	0.2										
DNA metabolic process	1.1	1	9			4			1		1
embryo development	0.5	0.1	0.1	1	0.1	0.1	1	0.1	1		
flower development	0.9	0.1			0.1		1		0.1		1
fruit ripening	0.1										
generation of precursor metabolites and energy	0.6		0.1	0.1	1	0.1	0.1				
growth	0.6	0.1				0.1	1	2	2	2	
lipid metabolic process	1.4	0.1		14	4	1	4				0.1
multicellular organism development	3.9	4	2	3	1	1	4	4	6	2	3
nucleobase-containing compound metabolic process	3.7		1	3		17			12		
other biological processes	0.7	0.1		1			1	3	1		0.1
Other cellular processes	17	15	26	18	14	22	13	9	22	16	17
other metabolic processes	13	14	16	24	19	20	9	7	16	14	15
photosynthesis	0.1		0.1	0.1	2	0.1					
pollination	0.2				2	0.1	0.1			2	
post-embryonic development	1.9	3	1	2	1	1	2	2	1	2	2
protein metabolic process	1.6	12	1	1	1	1	2	0.1	1	2	14
regulation of gene expression,epigenetic	0.4					0.1					
regulation of molecular function	0.4		0.1			0.1	0.1	0.1		2	
reproduction	2.7	2	9	2	3	1	2	1	1	4	1
response to abiotic stimulus	2.5	1	1	1	1	1	3	5	0.1		1
response to biotic stimulus	2	2	0.1	1	2		5	2		6	1
response to chemical	4.5	3	0.1	2	5	1	6	10	5		2
response to endogenous stimulus	2.1	1		1	1	1	3	4	4		1
response to external stimulus	2.4	2	0.1	1	4		6	2		6	2
response to light stimulus	1	1		0.1	2	1	2	1	1		1
response to stress	5.9	4	4	3	5	5	10	7	2	6	4
secondary metabolic process	0.5										
signal transduction	1.9	0.1		1	1	0.1	2	2	3	2	0.1
translation	0.8	0.1	0.1			1	0.1		0.1		0.1
transport	2.8	1			6	1	6	25		10	1
tropism	0.1										
unknown biological processes	0.1							0.1			

Column titled B12 contains the percentages of the genes that are involved in each Biological Process subcategory - as denoted by the Gene Ontology - for the B12 dataset. The columns titled C0 to C13 contain the percentages of genes that are involved in each Biological Process subcategory - as denoted by the Gene Ontology - in each cluster of Quadrant 1. The colour scale used ranges from red to blue, with blue cells containing higher values and red cells containing lower values.

Tables 11 and 12 show the actual percentages of each subdivision in the respective broad category for each cluster in Quadrant 1 for the B12 dataset. However it is more informative to see how these percentages for the GO function in the Quadrant 1 of the B12 dataset differ from the percentages for each GO function in the whole dataset. An effective clustering method would produce a group of genes significantly related to a single biological or molecular function as opposed to a cluster with several unrelated functions amongst the whole dataset. Tables 13 and 14 show the relative increase or decrease in percentage for each function within the broad functional category when compared to the percentage for each function in the whole dataset. This relative change was found using the following:

$$\frac{\text{pct for fucntional categorisation in cluster} - \text{pct for fucntional categorisation in whole dataset}}{\text{pct for fucntional categorisation in whole dataset}} \tag{5.2.1}$$

These tables are for GO Molecular Function and GO Biological Process respectively.

Table 13: Relative change in percentage of each Molecular Function subcategory between the clusters in Quadrant 1 of the B12 dataset and the whole B12 dataset.

Molecular Function	B12	C0	C2	C3	C4	C5	C8	C9	C11	C12	C13
carbohydrate binding	0.3	0.5									0.5
catalytic activity	17.0	0.9	0.3	1.5	-0.2	0.7	-0.1	-0.4	-0.8	0.2	0.9
chromatin binding	0.3		1.8								
DNA - binding transcription factor activity	3.1			-0.7		-0.6			-0.7		
DNA binding	6.3		-0.2	-0.9		0.9			-0.7		
enzyme regulator activity	1.0				-0.3	-0.6	1.9	-0.2			
hydrolase activity	5.7		0.8	-0.9	0.2	-0.3	0.1	-0.7	2	-0.7	
kinase activity	3.3		0.5		0.4	-0.4	0.5			5.7	-0.7
lipid binding	0.4				0.9						
motor activity	0.6								22		
nuclease activity	0.4		6.8	1							
nucleic acid binding	2.4		-0.7			0.4					
nucleotide binding	5.9	0.6	1.2	-0.02	0.4	-0.6	0.6	-0.5	0.7	0.5	1.3
other binding	16.5	-0.1	0.5	-0.05	0.9	-0.5	0.6	-0.5	-0.2	-0.3	-0.1
other molecular functions	0.3										
oxygen binding	0.1						16				
protein binding	14.2	-0.3	-0.6	-0.6	-0.5	-0.2	-0.3	-0.9	0.3	-0.1	-0.6
RNA binding	3.7		-0.4	-0.8		-0.8	-0.9		0.8		
signalling receptor activity	0.5				1.7		1.5			6.3	
signalling receptor binding	0.2						3.4				
structural molecule activity	1.8				0.5	-0.8					
transcription regulator activity	0.2					0.8					
transferase activity	10.0	2.2	-0.3	1.6	-0.5	1.5	0		0.4	0.8	2.2
translation factor activity, RNA binding	0.3		1.8				0.5				
translation regulator activity	0.1										
transporter activity	4.7				2.6	-0.9	1.5	15			
unknown molecular functions	0.9		-0.1								

Column titled B12 contains the percentages of the genes that are involved in each Molecular Function subcategory - as denoted by the Gene Ontology - for the B12 dataset. The columns titled C0 to C13 contain the relative change in the percentage of genes that are involved in each Molecular Function subcategory - as denoted by the Gene Ontology - between the clusters in Quadrant 1 and the whole B12 dataset . The colour scale used ranges from red to blue, with blue cells containing higher values and red cells containing lower values.

Table 14: Relative change in percentage of each Biological Process Function subcategory between the clusters in Quadrant 1 of the B12 dataset and the whole B12 dataset.

Biological Process	B12	C0	C2	C3	C4	C5	C8	C9	C11	C12	C13
abscission	0.1				3.4						
anatomical structure development	4.0	0.1	-0.8	-0.5	-0.1	-0.8	-0.1	0.2	0.6	0.1	-0.4
biosynthetic process	5.0	-0.9	1.1	1.3	0.3	1.9	-0.4	-0.9	-0.3		-1.0
carbohydrate metabolic process	0.9		-0.8	0.6			-0.2				
catabolic process	2.2	4.4	-0.6	0.1	0.9	-0.3	-0.2	-0.8			5.3
cell communication	0.5	-0.6		-0.1			-0.4		1.9		0.1
cell cycle	1.8	-0.5	3.7				-0.9		-0.2		-0.7
cell death	0.3				0.4		3.0	0.3	0.3		
cell differentiation	0.9	-0.5				-0.6	0.3	0.1	-0.5	1.5	-0.7
cell growth	0.4	0.1				-0.3	0.7	1.9	-0.1	3.9	
cell-cell signalling	0.1										
cellular component organization	3.6	-0.6	1.2	-0.8	0.5	-0.4	0.3	0.1	0.1	-0.5	-0.5
cellular homeostasis	0.3				4.0			5.0			
cellular protein modification process	2.1	4.8	-0.9	-0.7	-0.1	-0.4	-0.4		0.1	5.6	5.5
circadian rhythm	0.2										
DNA metabolic process	1.1	-0.2	6.6			3.0			-0.1		-0.1
embryo development	0.5	-0.2	-0.3	0.8	-0.2	-0.4	-0.1	-0.3	0.4		
flower development	0.9	-0.5			-0.6		0.3		-0.6		-0.4
fruit ripening	0.1										
generation of precursor metabolites and energy	0.6		-0.7	-0.2	1.1	-0.5	-0.4				
growth	0.6	-0.3				-0.5	0.4	1.6	2.1	2.3	
lipid metabolic process	1.4	-0.8		9.5	1.7	-0.6	2.0				-0.8
multicellular organism development	3.9	0.1	-0.5	-0.1	-0.7	-0.8	0.1	0.1	0.5	-0.5	-0.2
nucleobase-containing compound metabolic process	3.7		-0.7	-0.1		3.6			2.1		
other biological processes	0.7	-0.7		0.3			-0.1	3.5	0.1		-0.6
Other cellular processes	17.3	-0.1	0.5	0.1	-0.2	0.3	-0.3	-0.5	0.3	-0.1	-0.1
other metabolic processes	13.3	0.1	0.2	0.8	0.5	0.5	-0.3	-0.5	0.2	0.1	0.2
photosynthesis	0.1		0.7	1.0	13.1	1.6					
pollination	0.2				9.6	0.3	0.4			7.6	
post-embryonic development	1.9	0.4	-0.7	0.2	-0.6	-0.7	-0.2	-0.2	-0.4	0.1	-0.1
protein metabolic process	1.6	6.6	-0.6	-0.6	-0.2	-0.3	0.1	-0.8	-0.5	0.3	7.5
regulation of gene expression,epigenetic	0.4					-0.2					
regulation of molecular function	0.4		-0.5			-0.2	-0.1	0.1		4.5	
reproduction	2.8	-0.4	2.2	-0.4	0.2	-0.8	-0.1	-0.6	-0.5	0.5	-0.5
response to abiotic stimulus	2.5	-0.5	-0.6	-0.4	-0.7	-0.6	0.4	0.9	-0.9		-0.8
response to biotic stimulus	2.0	-0.2	-0.8	-0.5	0.1		1.6	-0.1		2.0	-0.6
response to chemical	4.5	-0.4	-0.9	-0.5	0.1	-0.8	0.3	1.2	0.1		-0.5
response to endogenous stimulus	2.1	-0.4		-0.6	-0.6	-0.7	0.37	0.9	1.1		-0.6
response to external stimulus	2.4	-0.2	-0.8	-0.6	0.7		1.4	-0.2		1.5	-0.3
response to light stimulus	1.0	0.12		-0.7	0.7	0.5	0.6	-0.2	-0.2		-0.2
response to stress	5.9	-0.3	-0.3	-0.5	-0.3	-0.2	0.7	0.1	-0.7	0.1	-0.4
secondary metabolic process	0.5										
signal transduction	1.9	-0.8		-0.6	-0.3	-0.9	0.3	-0.2	0.8	0.1	-0.9
translation	0.8	-0.5	-0.5			0.1	-0.8		-0.6		-0.7
transport	2.8	-0.8			1.2	-0.7	1.1	8.0		2.6	-0.8
tropism	0.1										
unknown biological processes	0.1							8.0			

Column titled B12 contains the percentages of the genes that are involved in each Biological Process subcategory - as denoted by the Gene Ontology - for the B12 dataset. The columns titled C0 to C13 contain the relative change in the percentage of genes that are involved in each Biological Process subcategory - as denoted by the Gene Ontology - between the clusters in Quadrant 1 and the whole B12 dataset . The colour scale used ranges from red to blue, with blue cells containing higher values and red cells containing lower values.

Furthermore, these values indicating the relative increase or decrease of percentage for each function were normalised by using min/max scaling as well as taking the logarithm with base 10 of the value. Scaling of the relative change of percentage was done in order to give a clearer picture of the GO Biological Process or GO Molecular Function that were more represented in each cluster.
The colour scale that was used in both Table 13 - 14 colours higher values in a blue shade and lower values in red. These figures provide a broad overview of the functional split that exists within the B12 dataset for GO Biological Process and GO Molecular Function.
To produce a summary of the results that have been obtained here the maximum values in each cluster(showing the function that was most represented) were extracted along with the cluster which contained the highest proportion for each functional categorisation. To clarify this point consider Cluster 3 in Table 14 above. In this cluster *lipid metabolic function* had the highest relative change in percentage when compared to the whole B12 dataset out of all of the functional categorisations. This is to say that the percentage of proteins in cluster 3 that related to *lipid metabolic function* was considerably higher than the percentage of proteins relating to this function in the whole B12 dataset(this is a column-wise operation). From this point onwards this relation will be referred to as the function which was the most relatively represented in the cluster. A similar row-wise operation was done for each functional categorisation. For *lipid metabolic process* this was also found to be Cluster 3 of the B12 dataset. It was in this cluster where the greatest relative change in percentage between cluster and B12 dataset was found.
This was done for the actual percentage of each functional categorisation, the relative change of that percentage when compared to the whole dataset, along with the min/max normalisation and logarithm of this value indicating relative change. These values have been combined in Table 15 and Table 16.

Table 15: Summary of GO Molecular Function for the B12 dataset

Molecular Function	Actual	Change	Normalised	Log	Pts	Max Function	
carbohydrate binding	Quad 3 C8	Quad 3 C8	Quad 3 C8	Quad 3 C8	23	carbohydrate binding	Match
catalytic activity	Quad 1 C3	Quad 1 C3	Quad 1 C3	Quad 1 C3	3	transferase activity	No Match
chromatin binding	Quad 3 C9	Quad 3 C9	Quad 3 C9	Quad 3 C9	30	chromatin binding	Match
DNA - binding transcription factor activity	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	translation regulator activity	No Match
DNA binding	Quad 4 C0	Quad 4 C0	Quad 4 C0	Quad 4 C0	10	transcription regulator activity	No Match
enzyme regulator activity	Quad 2 C7	Quad 2 C7	Quad 2 C7	Quad 2 C7	33	enzyme regulator activity	Match
hydrolase activity	Quad 4 C8	Quad 4 C8	Quad 4 C8	Quad 4 C8	7	transcription regulator activity	No Match
kinase activity	Quad 1 C12	Quad 1 C12	Quad 1 C12	Quad 1 C12	1	signalling receptor activity	No Match
lipid binding	Quad 4 C13	Quad 4 C13	Quad 4 C13	Quad 4 C13	1	lipid binding	Match
motor activity	Quad 3 C13	Quad 3 C13	Quad 3 C13	Quad 3 C13	47	motor activity	Match
nuclease activity	Quad 3 C2	Quad 3 C2	Quad 3 C2	Quad 3 C2	24	nuclease activity	Match
nucleic acid binding	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	translation regulator activity	No Match
nucleotide binding	Quad 2 C3	Quad 2 C3	Quad 2 C3	Quad 2 C3	37	nucleotide binding	Match
other binding	Quad 2 C2	Quad 2 C2	Quad 2 C2	Quad 2 C2	33	other binding	Match
other molecular functions	Quad 3 C11	Quad 3 C11	Quad 3 C11	Quad 3 C11	33	other molecular functions	Match
oxygen binding	Quad 2 C21	Quad 2 C21	Quad 2 C21	Quad 2 C21	24	oxygen binding	Match
protein binding	Quad 3 C2	Quad 3 C2	Quad 3 C2	Quad 3 C2	24	nuclease activity	No Match
RNA binding	Quad 2 C4	Quad 2 C4	Quad 2 C4	Quad 2 C4	22	structural molecule activity	No Match

Molecular Function	Actual	Change	Normalised	Log	Pts	Max Function	
signalling receptor activity	Quad 2 C1	Quad 2 C1	Quad 2 C1	Quad 2 C1	17	signalling receptor activity	Match
signalling receptor binding	Quad 3 C14	Quad 3 C14	Quad 3 C14	Quad 3 C14	38	signalling receptor binding	Match
structural molecule activity	Quad 3 C19	Quad 3 C19	Quad 3 C19	Quad 3 C19	41	structural molecule activity	Match
transcription regulator activity	Quad 4 C8	Quad 4 C8	Quad 4 C8	Quad 4 C8	7	transcription regulator activity	Match
transferase activity	Quad 1 C13	Quad 1 C13	Quad 1 C13	Quad 1 C13	3	transferase activity	Match
translation factor activity, RNA binding	Quad 2 C6	Quad 2 C6	Quad 2 C6	Quad 2 C6	45	translation regulator activity	No Match
translation regulator activity	Quad 3 C15	Quad 3 C15	Quad 3 C15	Quad 3 C15	44	translation regulator activity	Match
transporter activity	Quad 1 C9	Quad 1 C9	Quad 1 C9	Quad 1 C9	2	transporter activity	Match
unknown molecular functions	Quad 3 C11	Quad 3 C11	Quad 3 C11	Quad 3 C11	33	other molecular functions	No Match

The first column holds the different categories that fall under GO Molecular Function. The four columns Actual, Change, Normalisation and Log hold the clusters in which there was the greatest representation of each functional category. The notation used in these columns first include the quadrant where the cluster resides (Quad 1,2,3 or 4) followed by the cluster number itself(C0, C1,..., etc.). The 6th column indicates the amount of points(genes) that were assigned to the cluster mentioned in each row. The 7th column named Max Function matches the cluster in columns 2 - 4 to the functional categorisation that was most represented(highest value) in the mentioned cluster. The final column indicates whether there is a match between the cluster that has the highest representation of each functional categorisation with the most represented functional categorisation in the same cluster. Cells are coloured in red if they are duplicates in their respective columns. Duplicate cells in columns 2-4 imply that there is more than one functional categorisation that has a strong presence in a single cluster.

Table 16: Summary of GO Biological Process for the B12 dataset

Biological Process	Actual	Change	Normalised	Log	Pts	Max Function	
abscission	Quad 2 C9	Quad 2 C9	Quad 2 C9	Quad 2 C9	27	abscission	Match
anatomical structure development	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	circadian rhythm	No Match
biosynthetic process	Quad 4 C11	Quad 4 C11	Quad 4 C11	Quad 4 C11	24	lipid metabolic process	No Match
carbohydrate metabolic process	Quad 3 C8	Quad 3 C8	Quad 3 C8	Quad 3 C8	23	carbohydrate metabolic process	Match
catabolic process	Quad 1 C13	Quad 1 C13	Quad 1 C13	Quad 1 C13	3	protein metabolic process	No Match
cell communication	Quad 4 C0	Quad 4 C0	Quad 4 C0	Quad 4 C0	10	cell communication	Match
cell cycle	Quad 2 C7	Quad 2 C7	Quad 2 C7	Quad 2 C7	33	cell cycle	Match
cell death	Quad 3 C4	Quad 3 C4	Quad 3 C4	Quad 3 C4	46	cell death	Match
cell differentiation	Quad 3 C6	Quad 3 C6	Quad 3 C6	Quad 3 C6	47	generation of precursor metabolites and energy	No Match
cell growth	Quad 2 C19	Quad 2 C19	Quad 2 C19	Quad 2 C19	21	cell growth	Match
cell-cell signalling	Quad 2 C3	Quad 2 C3	Quad 2 C3	Quad 2 C3	37	tropism	No Match
cellular component organization	Quad 2 C4	Quad 2 C4	Quad 2 C4	Quad 2 C4	22	translation	No Match

Biological Process	Actual	Change	Normalised	Log	Pts	Max Function	
cellular homeostasis	Quad 2 C21	Quad 2 C21	Quad 2 C21	Quad 2 C21	24	cellular homeostasis	Match
cellular protein modification process	Quad 1 C12	Quad 1 C12	Quad 1 C12	Quad 1 C12	1	pollination	No Match
circadian rhythm	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	circadian rhythm	Match
DNA metabolic process	Quad 1 C2	Quad 1 C2	Quad 1 C2	Quad 1 C2	4	DNA metabolic process	Match
embryo development	Quad 3 C22	Quad 3 C22	Quad 3 C22	Quad 3 C22	15	embryo development	Match
flower development	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	circadian rhythm	No Match
fruit ripening	Quad 2 C8	Quad 2 C8	Quad 2 C8	Quad 2 C8	33	fruit ripening	Match
generation of precursor metabolites and energy	Quad 4 C9	Quad 4 C9	Quad 4 C9	Quad 4 C9	17	photosynthesis	No Match
growth	Quad 2 C17	Quad 2 C17	Quad 2 C17	Quad 2 C17	34	unknown biological processes	No Match
lipid metabolic process	Quad 1 C3	Quad 1 C3	Quad 1 C3	Quad 1 C3	3	lipid metabolic process	Match
multicellular organism development	Quad 3 C22	Quad 3 C22	Quad 3 C22	Quad 3 C22	15	embryo development	No Match
nucleobase-containing compound metabolic process	Quad 1 C5	Quad 1 C5	Quad 1 C5	Quad 1 C5	5	nucleobase-containing compound metabolic process	Match
other biological processes	Quad 3 C19	Quad 3 C19	Quad 3 C19	Quad 3 C19	41	translation	No Match
Other cellular processes	Quad 3 C13	Quad 3 C13	Quad 3 C13	Quad 3 C13	47	circadian rhythm	No Match
other metabolic processes	Quad 2 C2	Quad 2 C2	Quad 2 C2	Quad 2 C2	33	generation of precursor metabolites and energy	No Match
photosynthesis	Quad 4 C9	Quad 4 C9	Quad 4 C9	Quad 4 C9	17	photosynthesis	Match
pollination	Quad 1 C4	Quad 1 C4	Quad 1 C4	Quad 1 C4	6	photosynthesis	No Match
post-embryonic development	Quad 2 C5	Quad 2 C5	Quad 2 C5	Quad 2 C5	28	circadian rhythm	No Match
protein metabolic process	Quad 1 C13	Quad 1 C13	Quad 1 C13	Quad 1 C13	3	protein metabolic process	Match
regulation of gene expression,epigenetic	Quad 3 C2	Quad 3 C2	Quad 3 C2	Quad 3 C2	24	regulation of gene expression,epigenetic	Match
regulation of molecular function	Quad 2 C7	Quad 2 C7	Quad 2 C7	Quad 2 C7	33	cell cycle	No Match
reproduction	Quad 1 C2	Quad 1 C2	Quad 1 C2	Quad 1 C2	4	DNA metabolic process	No Match
response to abiotic stimulus	Quad 2 C13	Quad 2 C13	Quad 2 C13	Quad 2 C13	20	response to abiotic stimulus	Match
response to biotic stimulus	Quad 4 C7	Quad 4 C7	Quad 4 C7	Quad 4 C7	10	response to biotic stimulus	Match
response to chemical	Quad 4 C1	Quad 4 C1	Quad 4 C1	Quad 4 C1	13	pollination	No Match
response to endogenous stimulus	Quad 4 C0	Quad 4 C0	Quad 4 C0	Quad 4 C0	10	cell communication	No Match

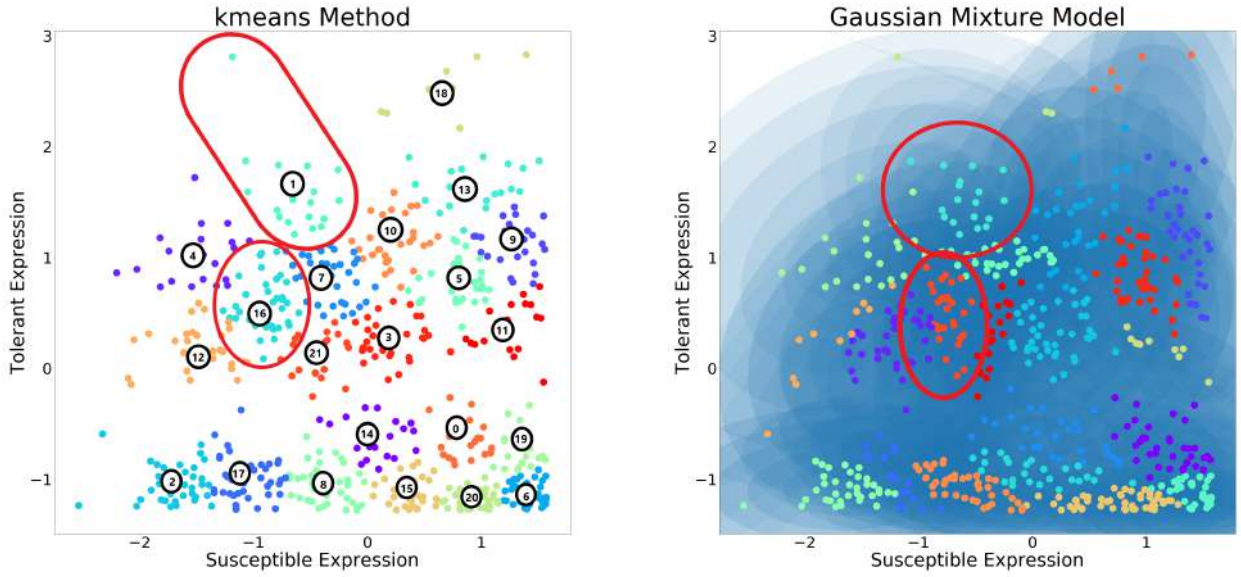
Biological Process	Actual	Change	Normalised	Log	Pts	Max Function	
response to external stimulus	Quad 4 C7	Quad 4 C7	Quad 4 C7	Quad 4 C7	10	response to biotic stimulus	No Match
response to light stimulus	Quad 4 C9	Quad 4 C9	Quad 4 C9	Quad 4 C9	17	photosynthesis	No Match
response to stress	Quad 2 C13	Quad 2 C13	Quad 2 C13	Quad 2 C13	20	response to abiotic stimulus	No Match
secondary metabolic process	Quad 3 C0	Quad 3 C0	Quad 3 C0	Quad 3 C0	32	secondary metabolic process	Match
signal transduction	Quad 4 C13	Quad 4 C13	Quad 4 C13	Quad 4 C13	1	cell communication	No Match
translation	Quad 3 C19	Quad 3 C19	Quad 3 C19	Quad 3 C19	41	translation	Match
transport	Quad 1 C9	Quad 1 C9	Quad 1 C9	Quad 1 C9	2	transport	Match
tropism	Quad 2 C3	Quad 2 C3	Quad 2 C3	Quad 2 C3	37	tropism	Match
unknown biological processes	Quad 2 C17	Quad 2 C17	Quad 2 C17	Quad 2 C17	34	unknown biological processes	Match

The first column holds the different categories that fall under GO Biological Process. The four columns Actual, Change, Normalisation and Log hold the clusters in which there was the greatest representation of each functional category. The notation used in these columns first include the quadrant where the cluster resides (Quad 1,2,3 or 4) followed by the cluster number itself(C0, C1,..., etc.). The 6th column indicates the amount of points(genes) that were assigned to the cluster mentioned in each row. The 7th column named Max Function matches the cluster in columns 2 - 4 to the functional categorisation that was most represented(highest value) in the mentioned cluster. The final column indicates whether there is a match between the cluster that has the highest representation of each functional categorisation with the most represented functional categorisation in the same cluster. Cells are coloured in red if they are duplicates in their respective columns. Duplicate cells in columns 2-4 imply that there is more than one functional categorisation that has a strong presence in a single cluster.

To clarify the results presented in Table 15 and 16 consider Cluster 0 in Quadrant 4 in Table 16. In this cluster both cell communication and response to endogenous stimulus have a strong presence(i.e large percentage of the proteins in this cluster fall under these functional categories). A duplicate value in column 7 is only relevant if the matching values in column 2-4 are different. To clarify, if there are existing duplicates in columns 2-4 then there will naturally be duplicate values in column 7 (this is the case with cell cycle in Table 16). The duplicates in column 7 that are of interest are the ones which have the same functional categorisation but do not have a matching cluster in columns 2-4. For these cases the implication is that a single functional categorisation is split between more than one cluster, meaning that there was a strong presence of this functional categorisation in more than one cluster. An example of this is cell communication. According to Table 16 this functional categorisation is the most represented category in both Cluster 0 and Cluster 13 in Quadrant 4 of the B12 dataset. Evaluating the duplicates that exist in Table 15 and Table 16 provide valuable insight into the efficacy of k-means clustering on this dataset.

5.2.1 k-means and the Gaussian Mixture Model

Before proceeding into the biological relevance of the duplicates in Tables 15 and 16, a small example of the comparison of the results obtained from the k-means method and the Gaussian Mixture model will be presented.
To clarify the loss of information resulting from the mapping of Cassava genes to *Arabidopsis thaliana* gene homologs as well as the complexity introduced into the system by relating genes to proteins and then proteins to biological pathways/processes(process outline in Figure 71a), random clusters were chosen from the B12 dataset in Quadrant 2. For the k-means method these were clusters 1 and 16, with the relevant clusters in the Gaussian Mixture Model being clusters 16 and 7. These are presented in Figure 79a and 79b below with clusters outlined in red being the ones of interest.



(a) k-means Clustering Results in Quadrant 2 of the B12 Dataset (b) GMM Clustering Results in Quadrant 2 of the B12 Dataset

Figure 79: Plots depicting the results obtained from the k-means method as well as the Gaussian Mixture Model in Quadrant 2 of the B12 dataset. Clusters outlined in red are the clusters that will be examined for comparison.

Tables 17 and 18 below provide an analysis of the similarity between the clusters outlined in Figures 79a and 79b, before and after the gene homolog mapping process. Table 17 provides an example between 2 clusters that were highly similar both before and after the mapping. While Table 18 provides and example of a larger disparity between clusters 16 and 7 both before and after the gene homolog mapping.

Table 17: Comparison of the genes common to cluster 1 from the k-means method and cluster 16 from the Gaussian Mixture model before and after gene homolog mapping.

	Cluster	No. Genes	Genes in Common	No. Genes after Mapping	Common After Mapping
k-means	1	17	16	13	12
GMM	16	20	16	14	12

Table 18: Comparison of the genes common to cluster 16 from the k-means method and cluster 7 from the Gaussian Mixture model before and after gene homolog mapping.

	Cluster	No. Genes	Genes in Common	No. Genes after Mapping	Common After Mapping
k-means	16	35	28	24	18
GMM	7	37	28	23	18

Table 19: A comparison of the GO Molecular Function categorisations for cluster 1 from the k-means method and cluster 16 from the Gaussian Mixture Model.

Molecular Function	B12	k-mean	GMM	KM Change	GMM Change
carbohydrate binding	0.32				
catalytic activity	17.01	13.98	15.08	-0.18	-0.11
chromatin binding	0.28				
DNA - binding transcription factor activity	3.08	5.38	1.01	0.74	-0.67
DNA binding	6.30	8.60	1.01	0.36	-0.84
enzyme regulator activity	1.01	4.30	6.03	3.28	4.99
hydrolase activity	5.68	3.23	25.63	-0.43	3.51
kinase activity	3.27	5.38	3.02	0.64	-0.08
lipid binding	0.36		1.51		3.22
motor activity	0.62				
nuclease activity	0.41		0.50		0.24
nucleic acid binding	2.42	4.30		0.78	
nucleotide binding	5.88	3.23	4.02	-0.45	-0.32
other binding	16.51	11.83	7.54	-0.28	-0.54
other molecular functions	0.26	1.61		5.33	
oxygen binding	0.03				
protein binding	14.24	20.97	15.58	0.47	0.09
RNA binding	3.71		4.02		0.09
signalling receptor activity	0.50	3.76	0.50	6.57	0.01
signalling receptor binding	0.19				
structural molecule activity	1.77		3.02		0.70
transcription regulator activity	0.20				
transferase activity	10.03	5.91	5.53	-0.41	-0.45
translation factor activity, RNA binding	0.28				
translation regulator activity	0.03				
transporter activity	4.75	6.99	4.02	0.47	-0.15
unknown molecular functions	0.88	0.54	2.01	-0.39	1.29

The first column provides the GO Molecular Functions, with the percentage for each function provided for the whole B12 dataset in column 2. Columns 3 and 4 holds the actual percentages relating to each function for cluster 1 of the k-means method and cluster 16 of the Gaussian Mixture Model respectively. Columns 5 and 6 provide the relative change of percentage for each function in the cluster when compared to the whole B12 dataset in column 2. These values are given for the k-means method and the Gaussian Mixture Model respectively.

Table 19 provides a comparison of two clusters that differ by a total of 3 points(one of which is unique to the k-means result and two which are exclusively found in the Gaussian Mixture Model result). Due to the loss of information in the gene homolog mapping, combined with the allowance for a 2nd shell of interactions in the PPI network generation, there is a disagreement on which is the most prevalent functional categorisations. The k-means cluster has found the prevalent function to be *signalling receptor activity*, while the Gaussian Mixture Model has found the prevalent function to be *enzyme regulator activity*.
In contrast, Table 20 provides a comparison between 2 clusters that are less similar than the cluster presented in Table 19. Here out of a possible 24 genes that are present after the homolog mapping process in the k-means cluster, only 18 of these are common with the Gaussian Mixture Model cluster. However despite this seemingly

larger difference in common genes, both these clusters have found *translation regulator activity* to be the most prevalent function.
These results alone provide insight into the large disparity in possible solutions that can be produced by the differing of only a few points between different clustering method results. To further add to this point, the same process was applied for GO Biological Process functional categorisations.

Table 20: A comparison of the GO Molecular Function categorisations for cluster 16 from the k-means method and cluster 7 from the Gaussian Mixture Model.

Molecular Function	B12	k-mean	GMM	KM Change	GMM Change
carbohydrate binding	0.32				
catalytic activity	17.01	7.35	34.66	-0.57	1.04
chromatin binding	0.28		0.28		0.01
DNA - binding transcription factor activity	3.08		4.26		0.38
DNA binding	6.30	2.21	4.26	-0.65	-0.32
enzyme regulator activity	1.01		0.28		-0.72
hydrolase activity	5.68	2.94	0.85	-0.48	-0.85
kinase activity	3.27	5.88	0.85	0.80	-0.74
lipid binding	0.36		0.57		0.59
motor activity	0.62				
nuclease activity	0.41		1.14		1.79
nucleic acid binding	2.42		2.84		0.17
nucleotide binding	5.88	3.68	0.85	-0.38	-0.86
other binding	16.51	6.62	6.53	-0.60	-0.60
other molecular functions	0.26		0.28		0.11
oxygen binding	0.03				
protein binding	14.24	12.50	16.19	-0.12	0.14
RNA binding	3.71	29.41	1.71	6.94	-0.54
signalling receptor activity	0.50	0.74	0.28	0.48	-0.43
signalling receptor binding	0.19				
structural molecule activity	1.77	13.97	0.57	6.89	-0.68
transcription regulator activity	0.20				
transferase activity	10.03	4.41	22.73	-0.56	1.27
translation factor activity, RNA binding	0.28	7.35		25.26	
translation regulator activity	0.03	0.74	0.28	28.40	10.36
transporter activity	4.75	2.21	0.28	-0.54	-0.94
unknown molecular functions	0.88		0.28		-0.68

The first column provides the GO Molecular Functions, with the percentage for each function provided for the whole B12 dataset in column 2. Columns 3 and 4 holds the actual percentages relating to each function for cluster 16 of the k-means method and cluster 7 of the Gaussian Mixture Model respectively. Columns 5 and 6 provide the relative change of percentage for each function in the cluster when compared to the whole B12 dataset in column 2. These values are given for the k-means method and the Gaussian Mixture Model respectively.

Table 21: A comparison of the GO Biological Process categorisations for cluster 1 from the k-means method and cluster 16 from the Gaussian Mixture Model.

Biological Process	B12	k-mean	GMM	KM Change	GMM Change
abscission	0.10		0.41		3.28
anatomical structure development	4.01	6.72	1.90	0.68	-0.53
biosynthetic process	5.00	4.54	3.25	-0.09	-0.35
carbohydrate metabolic process	0.88		5.42		5.13
catabolic process	2.17	0.67	4.61	-0.69	1.12
cell communication	0.52	0.67	1.08	0.29	1.09
cell cycle	1.79		1.63		-0.09
cell death	0.30	1.01	0.68	2.37	1.27
cell differentiation	0.81	1.68		1.07	
cell growth	0.41		0.14		-0.67
cell-cell signalling	0.02				
cellular component organization	3.61	1.01	5.56	-0.72	0.54
cellular homeostasis	0.33				
cellular protein modification process	2.13	0.17	0.41	-0.92	-0.81
circadian rhythm	0.17				
DNA metabolic process	1.13	0.34	0.14	-0.70	-0.88
embryo development	0.53	0.34	0.41	-0.37	-0.24
flower development	0.95	2.19	0.14	1.30	-0.86
fruit ripening	0.01				
generation of precursor metabolites and energy	0.58		0.41		-0.29
growth	0.61	0.67		0.11	
lipid metabolic process	1.37	0.84	0.54	-0.39	-0.61
multicellular organism development	3.89	6.56	1.36	0.68	-0.65
nucleobase-containing compound metabolic process	3.73	2.52	2.30	-0.32	-0.38
other biological processes	0.70	0.67	0.14	-0.04	-0.81
Other cellular processes	17.33	13.95	16.13	-0.20	-0.07
other metabolic processes	13.28	9.58	9.62	-0.28	-0.28
photosynthesis	0.12		0.27		1.32
pollination	0.23				
post-embryonic development	1.91	2.52	0.81	0.32	-0.57
protein metabolic process	1.59	1.68	0.14	0.06	-0.91
regulation of gene expression,epigenetic	0.35				
regulation of molecular function	0.37	1.51	0.14	3.15	-0.63
reproduction	2.71	3.19	0.95	0.18	-0.65
response to abiotic stimulus	2.50	4.03	2.58	0.61	0.03
response to biotic stimulus	1.98	2.52	8.54	0.27	3.31
response to chemical	4.51	7.90	4.74	0.75	0.05
response to endogenous stimulus	2.10	4.37	1.63	1.08	-0.23
response to external stimulus	2.36	3.19	8.67	0.35	2.68
response to light stimulus	0.98	0.67	0.81	-0.31	-0.17
response to stress	5.94	8.74	10.98	0.47	0.85
secondary metabolic process	0.45	0.50	0.14	0.11	-0.70
signal transduction	1.87	3.53	2.30	0.89	0.23
translation	0.83				
transport	2.76	1.51	0.81	-0.45	-0.71
tropism	0.07		0.14		1.06
unknown biological processes	0.04		0.14		2.09

The first column provides the GO Biological Process, with the percentage for each function provided for the whole B12 dataset in column 2. Columns 3 and 4 holds the actual percentages relating to each function for cluster 1 of the k-means method and cluster 16 of the Gaussian Mixture Model respectively. Columns 5 and 6 provide the relative change of percentage for each function in the cluster when compared to the whole B12 dataset in column 2. These values are given for the k-means method and the Gaussian Mixture Model respectively. The colour scale used identifies lower value in red and higher values in blue.

Table 22: A comparison of the GO Biological Process categorisations for cluster 16 from the k-means method and cluster 7 from the Gaussian Mixture Model.

Biological Process	B12	k-mean	GMM	KM Change	GMM Change
abscission	0.10				
anatomical structure development	4.01	1.49	5.02	-0.63	0.25
biosynthetic process	5.00	2.97	3.99	-0.41	-0.20
carbohydrate metabolic process	0.88	0.50		-0.44	
catabolic process	2.17	0.50	6.86	-0.77	2.16
cell communication	0.52		0.41		-0.21
cell cycle	1.79		0.21		-0.89
cell death	0.30				
cell differentiation	0.81		0.10		-0.87
cell growth	0.41	0.50	0.10	0.21	-0.75
cell-cell signalling	0.02				
cellular component organization	3.61	4.46	1.33	0.23	-0.63
cellular homeostasis	0.33				
cellular protein modification process	2.13	2.48	8.29	0.16	2.90
circadian rhythm	0.17		0.61		2.65
DNA metabolic process	1.13	1.49	0.72	0.32	-0.36
embryo development	0.53	0.99	0.10	0.86	-0.81
flower development	0.95	0.50	2.35	-0.48	1.48
fruit ripening	0.01				
generation of precursor metabolites and energy	0.58				
growth	0.61	0.50	0.21	-0.18	-0.66
lipid metabolic process	1.37		2.66		0.94
multicellular organism development	3.89	1.98	3.99	-0.49	0.03
nucleobase-containing compound metabolic process	3.73	4.95	2.15	0.33	-0.42
other biological processes	0.70	2.48	0.51	2.53	-0.27
Other cellular processes	17.33	16.83	15.25	-0.03	-0.12
other metabolic processes	13.28	16.34	13.51	0.23	0.02
photosynthesis	0.12				
pollination	0.23	0.50	0.10	1.12	-0.56
post-embryonic development	1.91	0.99	2.87	-0.48	0.50
protein metabolic process	1.59	2.48	6.24	0.55	2.92
regulation of gene expression,epigenetic	0.35				
regulation of molecular function	0.37				
reproduction	2.71	0.99	3.28	-0.63	0.21
response to abiotic stimulus	2.50	0.99	3.28	-0.60	0.31
response to biotic stimulus	1.98	0.50	0.51	-0.75	-0.74
response to chemical	4.51	5.94	3.89	0.32	-0.14
response to endogenous stimulus	2.10	3.47	1.13	0.65	-0.46
response to external stimulus	2.36	0.50	0.82	-0.79	-0.65
response to light stimulus	0.98	0.50	1.95	-0.49	0.99
response to stress	5.94	2.97	5.73	-0.50	-0.03
secondary metabolic process	0.45		0.10		-0.77
signal transduction	1.87	1.98	0.82	0.06	-0.56
translation	0.83	19.31	0.51	22.40	-0.38
transport	2.76		0.41		-0.85
tropism	0.07				
unknown biological processes	0.04				

The first column provides the GO Biological Process, with the percentage for each function provided for the whole B12 dataset in column 2. Columns 3 and 4 holds the actual percentages relating to each function for cluster 16 of the k-means method and cluster 7 of the Gaussian Mixture Model respectively. Columns 5 and 6 provide the relative change of percentage for each function in the cluster when compared to the whole B12 dataset in column 2. These values are given for the k-means method and the Gaussian Mixture Model respectively. The colour scale used identifies lower value in red and higher values in blue.

Tables 21 and 21, show further disagreement between the clusters considered. In Table 21 in cluster 1 from the k-means solution, regulation of molecular function was found to be the most represented functional categorisation. This is in contrast to cluster 16 from the Gaussian Mixture Model solution which found this function to be carbohydrate metabolic process. The clear disparity between these solutions further highlights the complexity in attempting to compare clustering methods in this way.
The k-means solution that has been selected for testing does not claim to have the best solution. But prior to the increase in variables and complexity introduced by gene homolog mapping and PPI network generation, it was the solution that was highly similar to more than half of the methods tested. It therefore provides a good *"jumping-off point"* for the discovery of host immunity through differential gene expression analysis.

5.3 Unique Matches

Table 23 and Table 24 show the GO Biological Process and GO Molecular Functional categorisations in the B12 dataset respectively, where there is a unique match between the cluster that has the highest proportion of the function and this function has the strongest presence in said cluster. Functional categorisations that are unique along with their corresponding cluster, quadrant, the number of points that were assigned to that cluster and the percentage for that functional categorisation for the whole B12 dataset are depicted here.

Table 23: GO Biological Process Unique Functional Categorisation

Unique Functional Categorisations	Cluster	Points	Pct in B12 Dataset
abscission	Quad 2 C9	27	0.1
carbohydrate metabolic process	Quad 3 C8	23	0.88
cell death	Quad 3 C4	46	0.3
cell growth	Quad 2 C19	21	0.41
cellular homeostasis	Quad 2 C21	24	0.33
fruit ripening	Quad 2 C8	33	0.01
nucleobase-containing compound metabolic process	Quad 1 C5	5	3.37
regulation of gene expression,epigenetic	Quad 3 C2	24	0.35
secondary metabolic process	Quad 3 C0	32	0.45
transport	Quad 1 C9	2	2.76
lipid metabolic process	Quad 1 C3	3	1.37

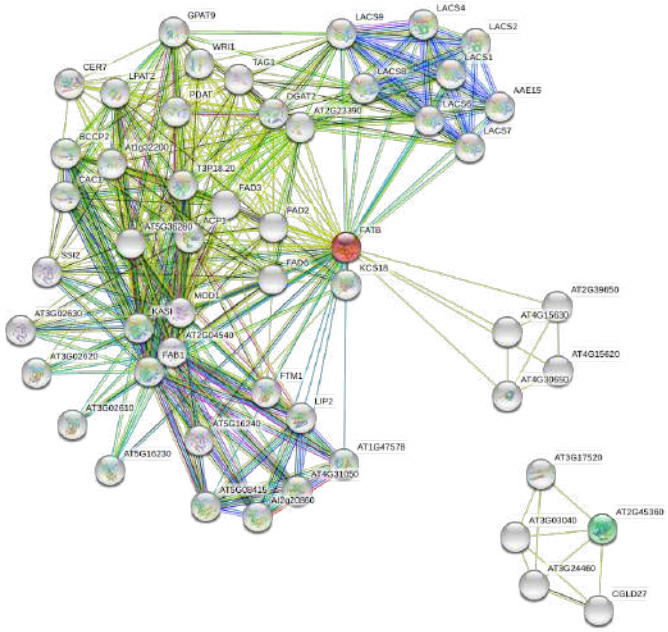
The first column provides the GO Biological Process that is in question. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the percentage of proteins in the B12 dataset which relate to the function in column one.

Table 24: GO Molecular Function Unique Functional Categorisation

Unique Functional Categorisations	Cluster	Points	Pct in B12 Dataset
carbohydrate binding	Quad 3 C8	23	0.3
chromatin binding	Quad 3 C9	30	0.3
enzyme regulator activity	Quad 2 C7	33	1
lipid binding	Quad 4 C13	1	0.4
motor activity	Quad 3 C13	47	0.6
nucleotide binding	Quad 2 C3	37	5.9
other binding	Quad 2 C2	33	17
oxygen binding	Quad 2 C21	24	0.03
signalling receptor binding	Quad 3 C14	38	0.2
transporter activity	Quad 1 C9	2	4.7

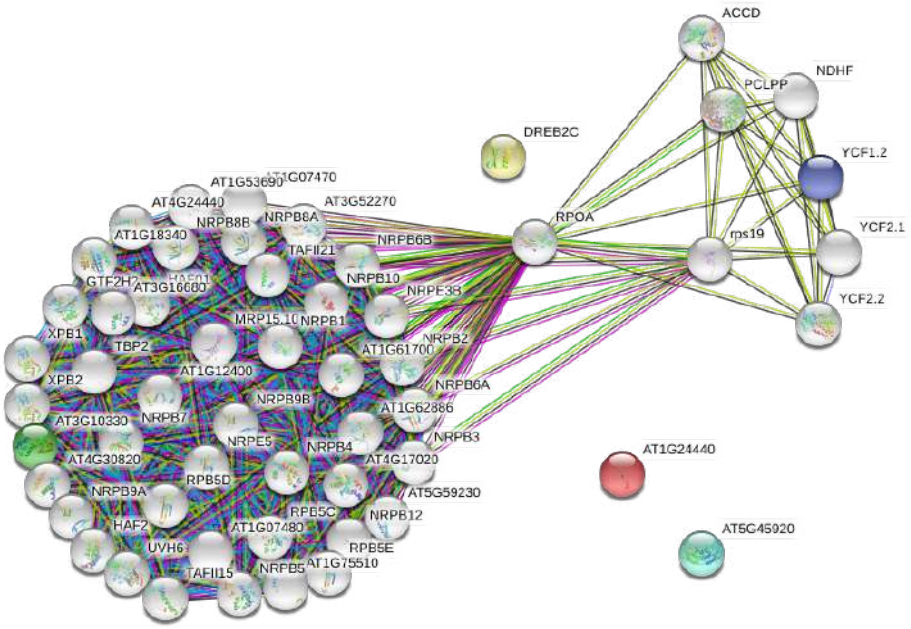
The first column provides the GO Molecular Function that is in question. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the percentage of proteins in the B12 dataset which relate to the function in column one.

The corresponding STRING networks for the 11 clusters in Table 23 are shown in Figures 80a - 90a.



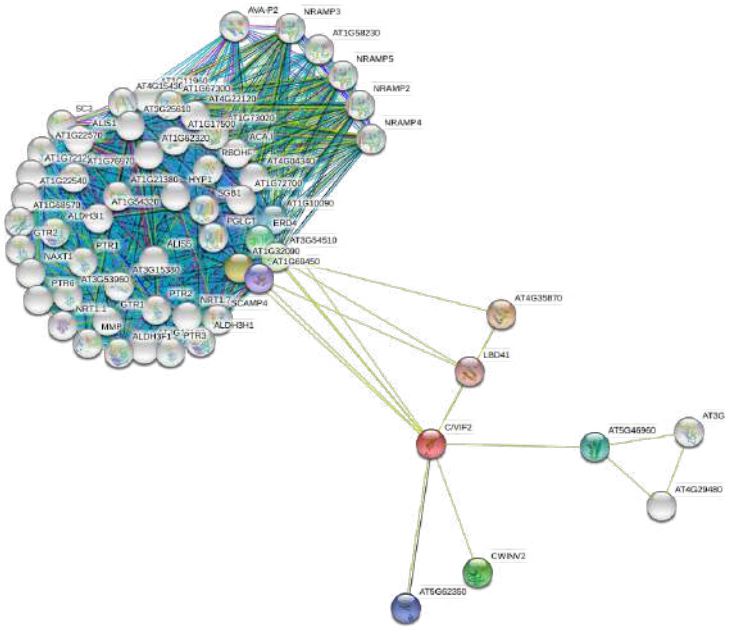
(a) PPI network for Cluster 3 In Qudrant 1 of B12 dataset

Figure 80: PPI network showing the interactions between the proteins for the genes in Cluster 3 of Quadrant 1 for the B12 dataset.



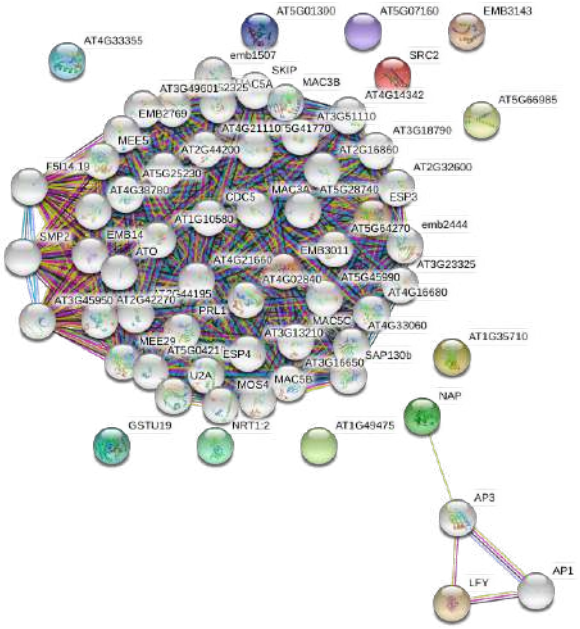
(a) PPI network for Cluster 5 In Qudrant 1 of B12 dataset

Figure 81: PPI network showing the interactions between the proteins for the genes in Cluster 5 of Quadrant 1 for the B12 dataset.



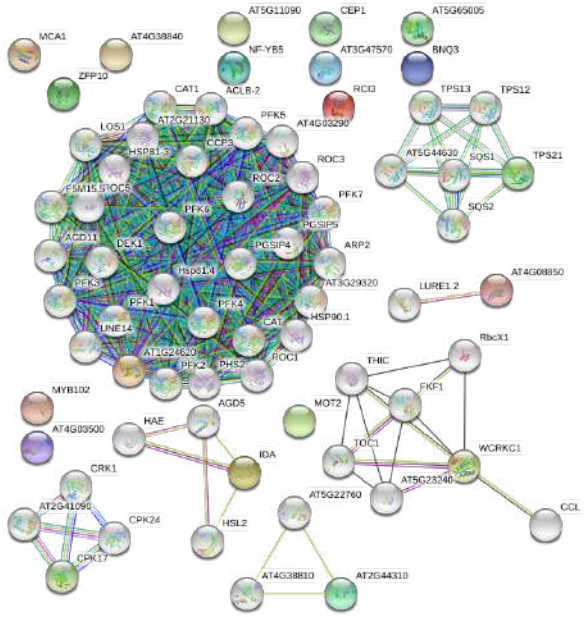
(a) PPI network for Cluster 9 In Qudrant 1 of B12 dataset

Figure 82: PPI network showing the interactions between the proteins for the genes in Cluster 9 of Quadrant 1 for the B12 dataset.



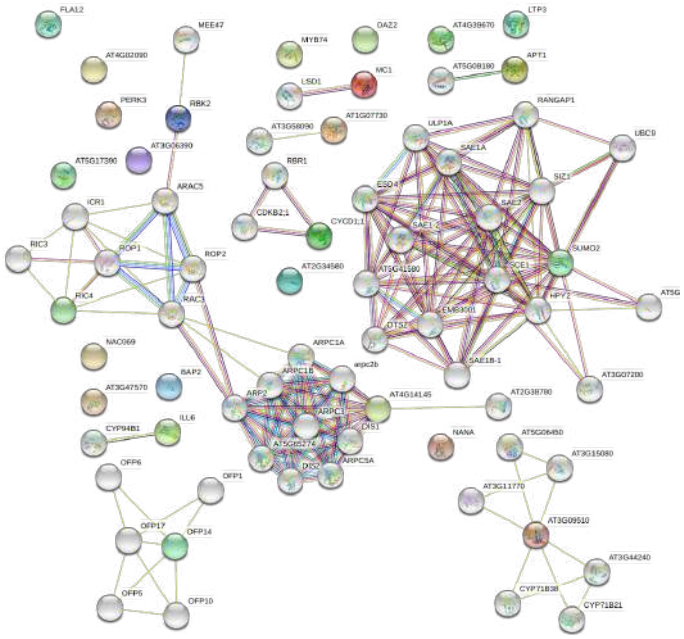
(a) PPI network for Cluster 8 In Qudrant 2 of B12 dataset

Figure 83: PPI network showing the interactions between the proteins for the genes in Cluster 8 of Quadrant 2 for the B12 dataset.



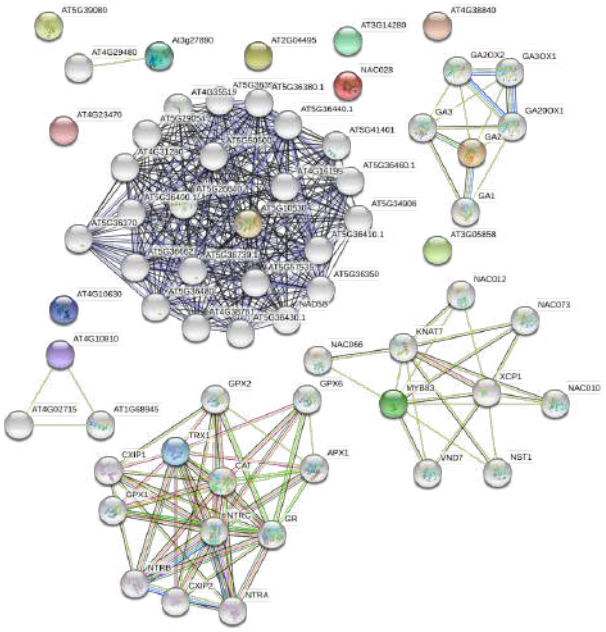
(a) PPI network for Cluster 9 In Qudrant 2 of B12 dataset

Figure 84: PPI network showing the interactions between the proteins for the genes in Cluster 9 of Quadrant 2 for the B12 dataset.



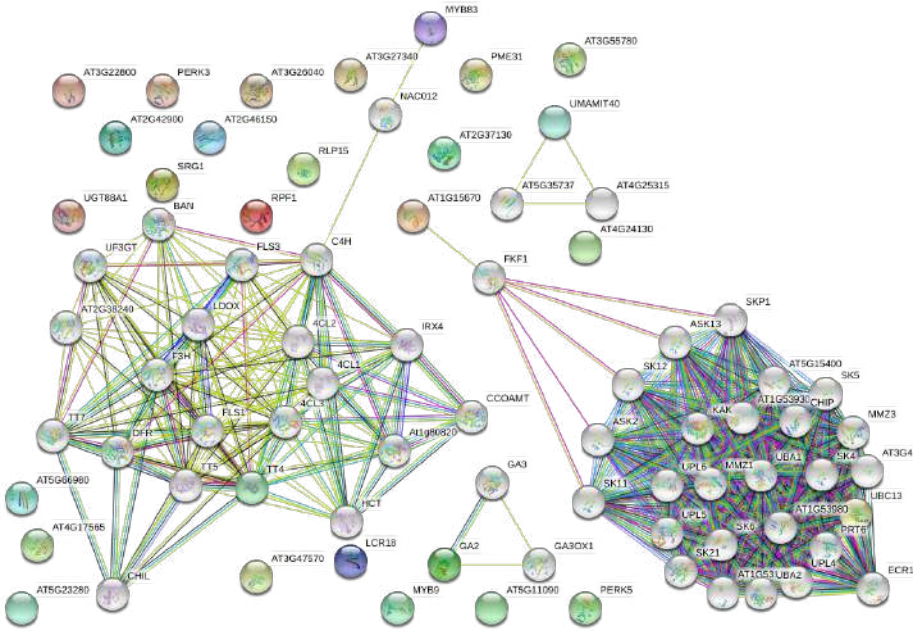
(a) PPI network for Cluster 19 In Qudrant 2 of B12 dataset

Figure 85: PPI network showing the interactions between the proteins for the genes in Cluster 19 of Quadrant 2 for the B12 dataset.



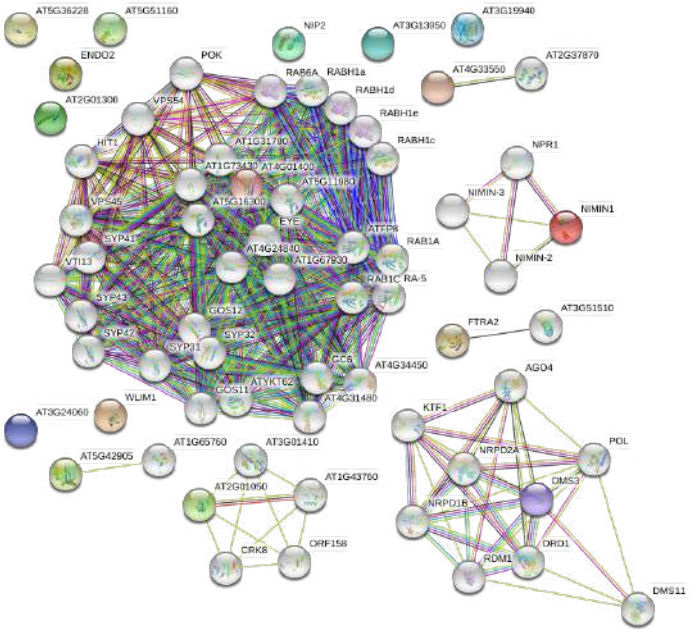
(a) PPI network for Cluster 21 In Qudrant 2 of B12 dataset

Figure 86: PPI network showing the interactions between the proteins for the genes in Cluster 21 of Quadrant 2 for the B12 dataset.



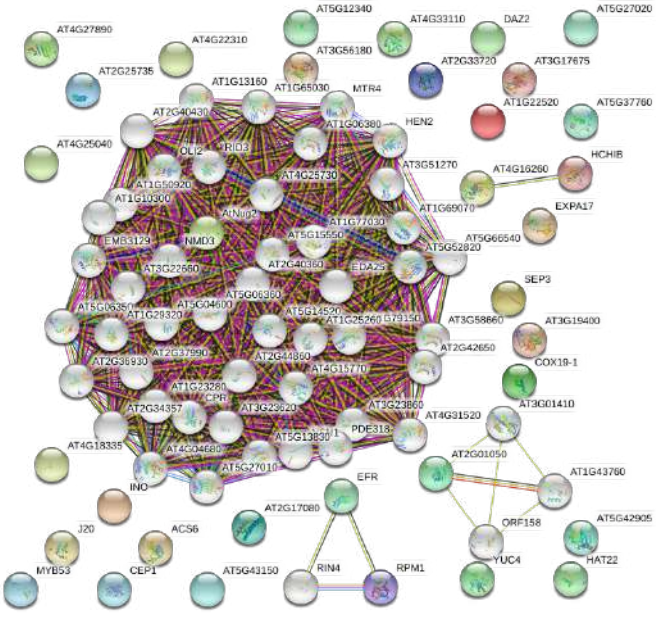
(a) PPI network for Cluster 0 In Qudrant 3 of B12 dataset

Figure 87: PPI network showing the interactions between the proteins for the genes in Cluster 0 of Quadrant 3 for the B12 dataset.



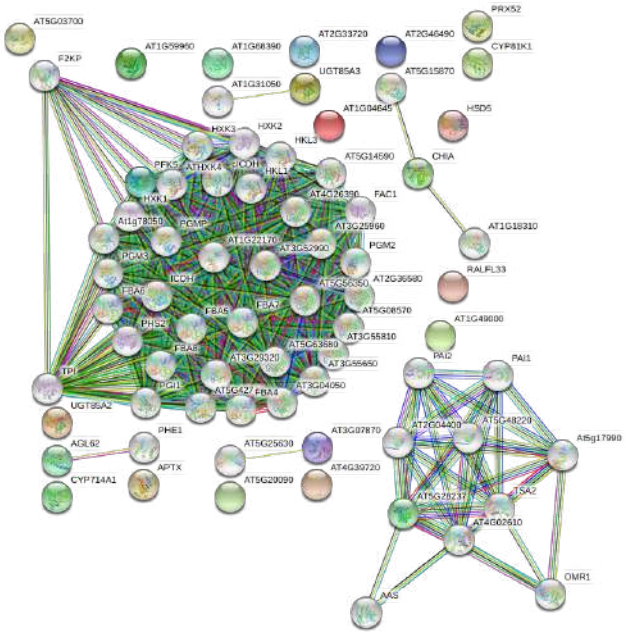
(a) PPI network for Cluster 2 In Qudrant 3 of B12 dataset

Figure 88: PPI network showing the interactions between the proteins for the genes in Cluster 2 of Quadrant 3 for the B12 dataset.



(a) PPI network for Cluster 4 In Qudrant 3 of B12 dataset

Figure 89: PPI network showing the interactions between the proteins for the genes in Cluster 4 of Quadrant 3 for the B12 dataset.



(a) PPI network for Cluster 8 In Qudrant 3 of B12 dataset

Figure 90: PPI network showing the interactions between the proteins for the genes in Cluster 8 of Quadrant 3 for the B12 dataset.

When looking at the STRING networks in conjunction with the total number of points(genes) that were assigned to that cluster it could be said that in these cases the k-means algorithm did succeed in grouping together genes of a related biological function.
Consider Figure 83a, while there appear to be more indirect connections(edges between white nodes) than direct connections(edges between coloured nodes), there is a tight interconnected network that has been formed at the centre. This implies that there is some relation in biological function, which could be the GO Biological Process of fruit ripening. This is not to say that fruit ripening does not feature at all in any other clusters, but in this cluster specifically there are a substantial amount of proteins that point to fruit ripening. The last point of interest from Table 23 is the column that shows the percentage of each functional categorisation for the B12 dataset. Each of these functional categorisation are relative low (at most 3.37%). Recall that the relative change in percentage was taken for each functional categorisation in each cluster and it is these that had a maximum relative change that are being discussed here. Looking at the percentages in the whole B12 dataset for these functional categorisations, it is clear that any value that is markedly above these small percentages would produce a substantially large relative change value. While this could imply that these functions have been erroneously marked as the most prevalent and most represented in the clusters, it is important to note that the relative change values were normalised to account for a large range of values produced. Furthermore if these particular functional categorisations are highly prevalent in one cluster when compared to the whole B12 dataset then a consequence of this could be that the cluster has correctly identified a set of genes that are specifically related to this functional categorisation.

5.4 Mismatched Functional Categorisations

Table 25 and Table 26 show the GO Biological Processes and GO Molecular Functions respectively, which have a high prevalence in only one cluster but which are not the most represented in that cluster. Using Quad 4 C13 (Cluster 13 in Quadrant 4 of the B12 dataset)as an example - the largest relative change in percentage for signal transduction was found in Cluster 13 of Quadrant 4 but this is not the functional categorisation that had the highest percentage in the cluster.

Table 25: GO Biological Process Unique Functional Categorisations

Unique Functional Categorisation	Cluster	Pts	Most Represented in Cluster
signal transduction	Quad 4 C13	1	cell communication
Other cellular processes	Quad 3 C13	47	circadian rhythm
cell differentiation	Quad 3 C6	47	generation of precursor metabolites and energy
other metabolic processes	Quad 2 C2	33	generation of precursor metabolites and energy
biosynthetic process	Quad 4 C11	24	lipid metabolic process
lipid metabolic process	Quad 1 C3	3	lipid metabolic process
pollination	Quad 1 C4	6	photosynthesis
cellular protein modification process	Quad 1 C12	1	pollination
response to chemical	Quad 4 C1	13	pollination
cellular component organization	Quad 2 C4	22	translation

The first column provides the GO Biological Process which has a high prevalence in only one cluster. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the the function which is most represented in the cluster given in column two.

Table 26: GO Molecular Function Unique Functional Categorisations

Unique Functional Categorisation	Cluster	Pts	Most Represented in Cluster
catalytic activity	Quad 1 C3	3	transferase activity
DNA binding	Quad 4 C0	10	transcription regulator activity
kinase activity	Quad 1 C12	1	signalling receptor activity
RNA binding	Quad 2 C4	22	structural molecule activity
translation factor activity, RNA binding	Quad 2 C6	45	translation regulator activity

The first column provides the GO Molecular Function which has a high prevalence in only one cluster. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the the function which is most represented in the cluster given in column two.

The results in these tables do not automatically imply that the clusters were created incorrectly. There are a number of different implications that could be drawn here. Two examples that are of interest here are the GO Biological Processes of signal transduction and cellular protein modification. These functional categorisations have been singled out as they are prevalent in clusters that only have one point in them each and yet these functions are not the most represented functional categorisation in that cluster. Column 4 contains the functional categorisation that is the most represented in these clusters, for signal transduction this function is cell communication while for cellular protein modification process it is pollination. Looking at the first case, this result makes sense as signal transduction is a method of cell communication and therefore it would make sense that a gene is related to both processes. For the latter case pollination may require cellular protein modification processes and therefore these functional categorisations could involve the same gene. The same is true for the other results in Table 25 and Table 26. This implies that while these functional categorisations in the first column are not the most represented, they are highly prevalent in these clusters and therefore these clusters could be related to the relevant functional categorisation.

5.5 Duplicate Clusters of Interest

Table 27 and Table 28 show the GO Biological Process and GO Molecular Function functional categorisations respectively that share a cluster in which they have a high prevalence.

Table 27: GO Biological Process Functional Categorisations

Functional Categorisation	Cluster	Pts	Most Represented
cell communication	Quad 4 C0	10	cell communication
response to endogenous stimulus	Quad 4 C0	10	cell communication
cell cycle	Quad 2 C7	33	cell cycle
regulation of molecular function	Quad 2 C7	33	cell cycle
anatomical structure development	Quad 2 C5	28	circadian rhythm
circadian rhythm	Quad 2 C5	28	circadian rhythm
flower development	Quad 2 C5	28	circadian rhythm
post-embryonic development	Quad 2 C5	28	circadian rhythm
DNA metabolic process	Quad 1 C2	4	DNA metabolic process
reproduction	Quad 1 C2	4	DNA metabolic process
embryo development	Quad 3 C22	15	embryo development
multicellular organism development	Quad 3 C22	15	embryo development
generation of precursor metabolites and energy	Quad 4 C9	17	photosynthesis
photosynthesis	Quad 4 C9	17	photosynthesis
response to light stimulus	Quad 4 C9	17	photosynthesis
catabolic process	Quad 1 C13	13	protein metabolic process
protein metabolic process	Quad 1 C13	3	protein metabolic process
response to abiotic stimulus	Quad 2 C13	20	response to abiotic stimulus
response to stress	Quad 2 C13	20	response to abiotic stimulus
response to biotic stimulus	Quad 4 C7	10	response to biotic stimulus
response to external stimulus	Quad 4 C7	10	response to biotic stimulus
other biological processes	Quad 3 C19	41	translation
translation	Quad 3 C19	41	translation
cell-cell signalling	Quad 2 C3	37	tropism
tropism	Quad 2 C3	37	tropism
growth	Quad 2 C17	34	unknown biological processes
unknown biological processes	Quad 2 C17	34	unknown biological processes

The first column provides the GO Biological Process which shares a cluster in which they have a high prevalence. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the the function which is most represented in the cluster given in column two.

Table 28: GO Molecular Function Functional Categorisations

Functional Categorisation	Cluster	Pts	Most Represented
nuclease activity	Quad 3 C2	24	nuclease activity
protein binding	Quad 3 C2	24	nuclease activity
other molecular functions	Quad 3 C11	33	other molecular functions
unknown molecular functions	Quad 3 C11	33	other molecular functions
hydrolase activity	Quad 4 C8	7	transcription regulator activity
transcription regulator activity	Quad 4 C8	7	transcription regulator activity
DNA - binding transcription factor activity	Quad 2 C5	28	translation regulator activity
nucleic acid binding	Quad 2 C5	28	translation regulator activity

The first column provides the GO Molecular Function which shares a cluster in which they have a high prevalence. Column two contains the Quadrant and cluster of interest, with the number of points in each cluster in column three. The final column provides the the function which is most represented in the cluster given in column two.

A high-level analysis of these results would imply that the clustering was ineffective in isolating functional categorisations in these clusters as there is not one functional catheterisation that is significantly prevalent here. But a closer examination of the nature of the categorisation suggests otherwise. Using embryo development and multicellular organism development as an example, both involve cell division and it can be argued that embryo development is a component of multicellular organism development. Furthermore, the functional categorisation that is most represented in this cluster is embryo development. The conclusion of this being that it is probable and possible that the genes in this cluster relate to these functions and these functions relate to one another. The same is true for the other pairs/ groups of functional categorisations that are depicted here. This leads to the conclusion that while the clusters did not isolate just one functional categorisation, they grouped together genes that are significantly involved in more than one related GO Biological Process or GO Molecular Function.

5.6 Duplicate Functional Categorisations of Interest

Table 29 and Table 30 show the functional categorisations that have a strong presence in more than one cluster for GO Biological Process and GO Molecular Function respectively.

Table 29: GO Biological Process Functional Categorisations

Cluster	**Pts**	**Most Represented**	**Pct B12 Dataset**
Quad 1 C12	1	signalling receptor activity	0.5
Quad 2 C1	17	signalling receptor activity	0.5
Quad 2 C4	22	structural molecule activity	1.8
Quad 3 C19	41	structural molecule activity	1.8
Quad 4 C0	10	transcription regulator activity	0.2
Quad 4 C8	7	transcription regulator activity	0.2
Quad 1 C3	3	transferase activity	10
Quad 1 C13	3	transferase activity	10
Quad 2 C5	28	translation regulator activity	0.03
Quad 2 C6	45	translation regulator activity	0.03
Quad 3 C15	44	translation regulator activity	0.03

The first column provides the Quadrant and cluster of interest, with the number of points in each cluster in column two. The GO Biological Process which is prevalent in more than one cluster in provided in column 3. Column 4 provides the percentage of proteins that are related to the process in column 3 for the whole dataset.

Table 30: GO Molecular Function Functional Categorisations

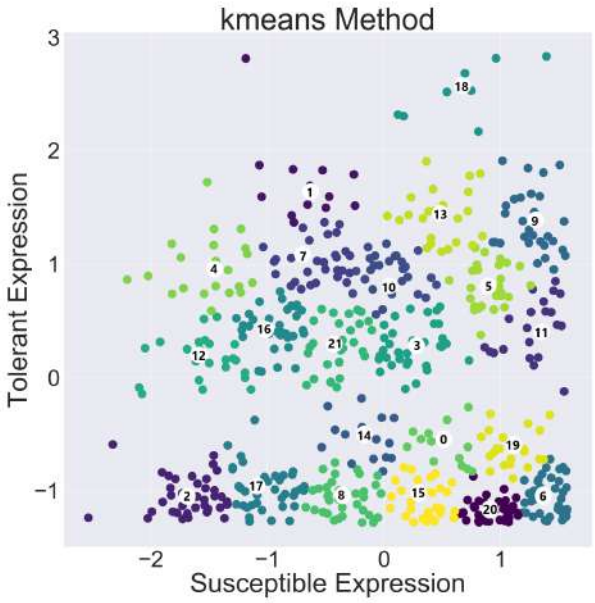
Cluster	Pts	Most Represented	Pct B12 Dataset
Quad 4 C0	10	cell communication	0.5
Quad 4 C13	1	cell communication	0.5
Quad 2 C5	28	circadian rhythm	0.2
Quad 3 C13	47	circadian rhythm	0.2
Quad 3 C6	47	generation of precursor metabolites and energy	0.6
Quad 2 C2	33	generation of precursor metabolites and energy	0.6
Quad 4 C11	24	lipid metabolic process	1.4
Quad 1 C3	3	lipid metabolic process	1.4
Quad 4 C9	17	photosynthesis	0.1
Quad 1 C4	6	photosynthesis	0.1
Quad 1 C12	1	pollination	0.2
Quad 4 C1	13	pollination	0.2
Quad 2 C4	22	translation	0.8
Quad 3 C19	41	translation	0.8

The first column provides the Quadrant and cluster of interest, with the number of points in each cluster in column two. The GO Molecular Function which is prevalent in more than one cluster in provided in column 3. Column 4 provides the percentage of proteins that are related to the process in column 3 for the whole dataset.

There are two different cases that are presented by the tables above. These being:

1. Duplicates in different quadrants.
2. Duplicates in the same quadrants but with respective clusters that do not overlap.

The first case presents a scenario where the same functional categorisation has a strong presence in more than two clusters but in different quadrants. This specifically does not imply an issue in the underlying clustering technique, but it does provide some interesting information on the respective dataset. Signalling receptor activity and structural molecule activity are two such examples in Table 29. For signalling receptor activity there are potentially genes that are involved in this process in both Quad 1 Cluster 12 and Quad 2 Cluster 1. An interpretation of this could be that there are two sets of genes that are involved in this process. The first set which were up-regulated in the susceptible host and down-regulated in the tolerant host(Quad 1 C12) and the second set of genes which was down-regulated in both the susceptible and tolerant host.
The second case can be explained with the use of the clustering results in Figure 91a. Using Cluster 5 and Cluster 6 in Quadrant 2 as an example. While translation regulator activity may be prevalent in two different clusters, these clusters are not physically close together and therefore this is not a case of a larger grouping being split incorrectly. Instead, it is possible that the genes in each cluster are involved in this process and yet do not display the same differential expression profiles.



(a) PPI network produced by STRING for Cluster 0 In Qudrant 1 of B12 dataset

5.7 GOs and Gene Expression Profiles

Table 32 and Table 33 below present a summary of the differential expression profiles for the genes that - according the k-means clustering - relate to the relevant GO Biological Process and GO Molecular Function. This was done by selecting the cluster in which each GO functional categorisation displayed the greatest positive change in percentage when compared to the whole dataset (this same process was applied to produce Table 16 and Table 15). Filtering the results in this way provided insight into the nature of the genes that were clustered by the proposed algorithm as they related to GO Biological Processes and GO Molecular Functions. The columns named *Susceptible* and *Tolerant* indicate whether the differential expression of the genes in the resultant cluster were up-regulated or down-regulated in the Susceptible host (x-axis variable) and Tolerant Host(y-axis variable) respectively. Table 31 below provides a key which aids in the interpretation of Table 32 and Table 33.

Table 31: The Differential expression profiles that exist in the four quadrants

	Susceptible Host	Tolerant Host
Quadrant 1	Up-regulated	Up-regulated
Quadrant 2	Down-regulated	Up-regulated
Quadrant 3	Down-regulated	Down-regulated
Quadrant 4	Up-regulated	Down-regulated

Using *abscission* as an example - according to the combined results produced by k-means clustering, STRING [36] and the information provided by TAIR [37] a plausible analysis could include:

- The 27 possible genes[58] relating to *abscission* were down-regulated in the susceptible host and up-regulated in the tolerant host at 12 days post-infection (B12 dataset = 12dpi).

[58]These genes are considered *possible* genes relating to *abscission* because it is not guaranteed that all the genes in the relevant cluster relate to this function. Rather it was proposed that the relevant cluster displayed a propensity to this function, as it was here that *abscission* displayed the greatest positive change in percentage when compared to the whole dataset.

- The 40 possible genes relating to *abscission* at 32 days post-infection were conversely up-regulated in the susceptible host and down-regulated in the tolerant host.
- The 25 possible genes relating to *abscission* were down-regulated in both the susceptible and tolerant host.

Tracking the GO Functional Categorisations across the datasets could provide insight into the progression of the virus whose affects on the Cassava plant are being examined. Using *abscission* again, the results in Table 32 imply that:

- At 12 days post-infection the susceptible hosts' response to the virus produced a down-regulation of the genes that are involved in *abscission* **or** the tolerant hosts' response to the virus resulted in up-regulation of the genes involved in abscission.
- At 32 days post-infection as the virus' effects on the genes involved in abscission lessen so does the down-regulation of the genes' differential expression.
- At 67 days post-infection differential expression of the genes involved in abscission are down-regulated in both the susceptible and tolerant host suggesting a possible decline in the affects of the virus thus eliminating the need for up-regulation of these genes to support tolerance.

Table 32: Expression profile summary for genes relating to GO Biological Process in Each Dataset. Up or down relates to the regulation of the genes relating to the relevant function . The number of points in the underlying cluster has also been provided for each dataset.

Function	**B12**	**Susceptible**	**Tolerant**	**B32**	**Susceptible**	**Tolerant**	**B67**	**Susceptible**	**Tolerant**
abscission	27	Down	Up	40	Up	Down	25	Down	Down
anatomical structure development	28	Down	Up	33	Down	Down	12	Up	Up
biosynthetic process	24	Up	Down	10	Up	Up	7	Up	Up
carbohydrate metabolic process	23	Down	Down	6	Down	Down	3	Up	Up
catabolic process	3	Up	Up	12	Down	Up	3	Up	Up
cell communication	10	Up	Down	15	Up	Up	3	Up	Up
cell cycle	33	Down	Up	25	Up	Down	18	Down	Up
cell death	46	Down	Down	27	Up	Up	7	Down	Down
cell differentiation	47	Down	Down	20	Down	Up	4	Up	Up
cell growth	21	Down	Up	13	Down	Down	3	Up	Up
cell-cell signalling	37	Down	Up	10	Down	Up	4	Up	Up
cellular component organization	22	Down	Up	10	Down	Up	4	Down	Up
cellular homeostasis	24	Down	Up	10	Down	Up	3	Down	Up
cellular protein modification process	1	Up	Up	12	Up	Up	31	Down	Down
circadian rhythm	28	Down	Up	29	Down	Up	31	Down	Up
DNA metabolic process	4	Up	Up	9	Down	Up	46	Up	Down
embryo development	15	Down	Down	12	Up	Down	12	Up	Up
flower development	28	Down	Up	33	Down	Down	41	Up	Down
fruit ripening	33	Down	Up	26	Up	Up	23	Up	Up

Table 32: Expression profile summary for genes relating to GO Biological Process in Each Dataset. Up or down relates to the regulation of the genes relating to the relevant function . The number of points in the underlying cluster has also been provided for each dataset.

Function	**B12**	**Susceptible**	**Tolerant**	**B32**	**Susceptible**	**Tolerant**	**B67**	**Susceptible**	**Tolerant**
generation of precursor metabolites and energy	17	Up	Down	20	Down	Up	18	Down	Down
growth	34	Down	Up	37	Down	Down	52	Up	Down
lipid metabolic process	3	Up	Up	10	Up	Up	5	Up	Down
multicellular organism development	15	Down	Down	12	Up	Down	4	Up	Up
nucleobase-containing compound metabolic process	5	Up	Up	38	Up	Down	9	Up	Down
other biological processes	41	Down	Down	16	Down	Up	4	Up	Up
Other cellular processes	47	Down	Down	10	Down	Up	7	Up	Up
other metabolic processes	33	Down	Up	43	Down	Up	5	Up	Down
photosynthesis	17	Up	Down	66	Up	Down	12	Up	Down
pollination	6	Up	Up	10	Down	Up	3	Up	Up
post-embryonic development	28	Down	Up	33	Down	Down	3	Down	Up
protein metabolic process	3	Up	Up	12	Down	Up	9	Up	Up
regulation of gene expression,epigenetic	24	Down	Down	29	Down	Up	26	Down	Down
regulation of molecular function	33	Down	Up	17	Up	Down	11	Up	Up
reproduction	4	Up	Up	33	Down	Down	7	Up	Up
response to abiotic stimulus	20	Down	Up	9	Down	Down	9	Up	Up
response to biotic stimulus	10	Up	Down	12	Down	Down	7	Down	Down
response to chemical	13	Up	Down	9	Down	Down	3	Down	Up
response to endogenous stimulus	10	Up	Down	12	Down	Up	16	Up	Up
response to external stimulus	10	Up	Down	24	Down	Down	7	Down	Down
response to light stimulus	17	Up	Down	66	Up	Down	9	Down	Up
response to stress	20	Down	Up	9	Down	Down	9	Up	Up
secondary metabolic process	32	Down	Down	40	Up	Down	5	Down	Down
signal transduction	1	Up	Down	10	Down	Up	20	Down	Up

Table 32: Expression profile summary for genes relating to GO Biological Process in Each Dataset. Up or down relates to the regulation of the genes relating to the relevant function . The number of points in the underlying cluster has also been provided for each dataset.

Function	**B12**	**Susceptible**	**Tolerant**	**B32**	**Susceptible**	**Tolerant**	**B67**	**Susceptible**	**Tolerant**
translation	41	Down	Down	6	Down	Down	1	Down	Up
transport	2	Up	Up	4	Down	Up	25	Up	Up
tropism	37	Down	Up	38	Up	Down	7	Up	Up
unknown biological processes	34	Down	Up	12	Down	Down	12	Down	Up

Table 33: Expression Profile summary for genes relating to GO Molecular Function in each dataset. Up or down relates to the regulation of the genes relating to the relevant function . The number of points in the underlying cluster has also been provided for each dataset.

Function	**B12**	**Susceptible**	**Tolerant**	**B32**	**Susceptible**	**Tolerant**	**B67**	**Susceptible**	**Tolerant**
carbohydrate binding	23	Down	Down	48	Up	Up	18	Down	Down
catalytic activity	3	Up	Up	10	Down	Up	3	Up	Up
chromatin binding	30	Down	Down	100	Up	Down	13	Up	Down
DNA - binding transcription factor activity	28	Down	Up	33	Down	Down	9	Up	Up
DNA binding	10	Up	Down	33	Down	Down	9	Up	Up
enzyme regulator activity	33	Down	Up	6	Down	Down	7	Up	Up
hydrolase activity	7	Up	Down	6	Down	Down	5	Up	Down
kinase activity	1	Up	Up	25	Up	Down	31	Down	Down
lipid binding	1	Up	Down	4	Down	Up	19	Up	Down
motor activity	47	Down	Down	31	Up	Down	20	Down	Up
nuclease activity	24	Down	Down	9	Down	Up	10	Down	Down
nucleic acid binding	28	Down	Up	33	Down	Down	9	Up	Up
nucleotide binding	37	Down	Up	17	Up	Down	20	Down	Up
other binding	33	Down	Up	7	Down	Up	7	Down	Down
other molecular functions	33	Down	Down	9	Down	Down	8	Up	Down
oxygen binding	24	Down	Up	48	Up	Up	5	Down	Up
protein binding	24	Down	Down	35	Down	Up	4	Down	Up
RNA binding	22	Down	Up	6	Down	Down	9	Up	Down
signalling receptor activity	17	Down	Up	33	Down	Down	7	Down	Down
signalling receptor binding	38	Down	Down	24	Down	Down	4	Up	Up
structural molecule activity	41	Down	Down	41	Down	Down	9	Down	Up
transcription regulator activity	7	Up	Down	2	Up	Up	11	Down	Up
transferase activity	3	Up	Up	17	Down	Up	20	Down	Up
translation factor activity, RNA binding	45	Down	Up	25	Down	Up	1	Down	Up
translation regulator activity	44	Down	Down	22	Down	Down	24	Up	Up

Table 33: Expression Profile summary for genes relating to GO Molecular Function in each dataset. Up or down relates to the regulation of the genes relating to the relevant function . The number of points in the underlying cluster has also been provided for each dataset.

Function	**B12**	**Susceptible**	**Tolerant**	**B32**	**Susceptible**	**Tolerant**	**B67**	**Susceptible**	**Tolerant**
transporter activity	2	Up	Up	21	Down	Up	4	Up	Up
unknown molecular functions	33	Down	Down	24	Down	Down	3	Down	Up

6 Conclusion

The methodology that was employed in Section 3 sought to use only differential expression values for the genes in the Cassava plant in both a susceptible host and tolerant host to infer a biological function and host immunity of susceptibility. The analysis presented in Section 5 suggests that the k-means method does provide a valid algorithm for the clustering of gene expression data when Susceptible expression profiles are plotted against Tolerant expression profiles. There is however further testing that needs to be done on the genes that were clustered by the algorithm to ascertain whether the assigned GO functional categorisations are accurate.

One significant task in this project arose from the lack of *a priori* knowledge of the value k, which determines the amount of clusters that are sought out by the majority of the clustering algorithms presented. While there are statistical tests that were employed in an attempt to discover this value for k, the multiple values proposed for an optimal k (resulting from the underlying data not having distinct *natural* groupings) inserted a degree of uncertainty in the results. A value for k was chosen for each quadrant in each of the three datasets provided, by seeking a consensus between the relevant statistical tests. While this is robust logic to implement, it does not guarantee optimal results. Without an external *golden standard* provided by either *true* labelling assignments or a *true* value for k, the various possible values for k as provided by the statistical tests each hold equal degrees of confidence. It is therefore proposed that if the predicted groupings as presented in this project do not yield sufficiently accurate results once the biological inferences are thoroughly tested, then the first improvement of the presented methodology be to select a value for k from the GO functional categorisations. This would either be a value for k of 27 if GO Molecular Functions are desired or setting k to 47 if the k-means method is to group by GO Biological Process[59]. This value of k would be implemented separately in each quadrant of each dataset.

The second recommendation to be made is that if the k-means method does not provide sufficient results once biological validity has been experimentally determined, then the clustering results from Complete linkage, Average Linkage or Spectral Clustering should be tested. According to the results presented in Table 7, these methods will provide a clustering solution that differs from the k-means solution and therefore will provide alternate results.

If however, experimental testing does validate the results obtained by the k-means algorithm then this does not in-turn invalidate the studies done on the application of other clustering algorithms on gene expression data. While studies done by G. Getz *et al.*, K. Yeung *et al.*, L. Peterson etc. found Coupled two-way clustering, model-based clustering, and hierarchical clustering combined with principal component analysis to be favourable methods when tested on the data provided for the relevant experiments [10, 11, 55]. These authors along with the others mentioned were not able to define the methods they tested as the *golden standard*. Therefore the biological validation of the k-means clustering results would add to the motivation for the use of clustering in gene expression analysis, as was first proposed by G. Michael *et al.* [8].

Furthermore if the the k-means algorithm is found to be capable of clustering data by related biological function (through the experimental validation of the results found here), then this would provide grounds for further research on increasing the accuracy of these results.

[59] These values were obtained by counting the total number of functional categorisations in the GO Molecular Function and GO Biological Process broad categories respectively.

References

[1] "Cassava's drought tolerance aids warming world.," *Available in http://croplife.org*, 2014.

[2] "The state of food security and nutrition in the world 2019." http://www.fao.org/state-of-food-security-nutrition. Accessed: 2020-04-10.

[3] C. Rey, "Geminivirus-host interactions: Connecting viral with cellular interactomes."

[4] D. J. Allocco, I. S. Kohane, and A. J. Butte, "Quantifying the relationship between co-expression, co-regulation and gene function," *BMC bioinformatics*, vol. 5, no. 1, p. 18, 2004.

[5] A. Sturn, J. Quackenbush, and Z. Trajanoski, "Genesis: cluster analysis of microarray data," *Bioinformatics*, vol. 18, no. 1, pp. 207–208, 2002.

[6] B. P. Lazzaro and D. S. Schneider, "The genetics of immunity," *Genetics*, vol. 197, no. 2, pp. 467–470, 2014.

[7] A. Anjum, S. Jaggi, E. Varghese, S. Lall, A. Bhowmik, and A. Rai, "Identification of differentially expressed genes in rna-seq data of arabidopsis thaliana: A compound distribution approach," *Journal of Computational Biology*, vol. 23, no. 4, pp. 239–247, 2016.

[8] G. S. Michaels, D. B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi, "Cluster analysis and data visualization of large-scale gene expression data," in *Pacific symposium on biocomputing*, vol. 3, pp. 42–53, 1998.

[9] A. P. Gasch and M. B. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *Genome biology*, vol. 3, no. 11, pp. research0059–1, 2002.

[10] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.

[11] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data," *Proceedings of the National Academy of Sciences*, vol. 97, no. 22, pp. 12079–12084, 2000.

[12] D. Huang and W. Pan, "Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data," *Bioinformatics*, vol. 22, no. 10, pp. 1259–1268, 2006.

[13] S. Wu, A.-C. Liew, H. Yan, and M. Yang, "Cluster analysis of gene expression data based on self-splitting and merging competitive learning," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 1, pp. 5–15, 2004.

[14] F. Lobo, "Fuzzy c-means algorithm," 2012. http://www.fernandolobo.info/dm/slides/fuzzy-c-means.pdf.

[15] K. Bataineh, M. Naji, and M. Saqer, "A comparison study between various fuzzy clustering algorithms.," *Jordan Journal of Mechanical & Industrial Engineering*, vol. 5, no. 4, 2011.

[16] J. Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," in *Proceedings of the Princeton conference in honor of Professor S. Bochner*, pp. 195–199, 1969.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, pp. 849–856, 2002.

[19] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Advances in neural information processing systems*, pp. 305–312, 2004.

[20] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[22] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[23] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[24] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, 1975.

[25] S. Anand, S. Mittal, O. Tuzel, and P. Meer, "Semi-supervised kernel mean shift clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1201–1215, 2013.

[26] D. Anderson, K. Burnham, and G. White, "Aic model selection in overdispersed capture-recapture data," *Ecology*, vol. 75, no. 6, pp. 1780–1793, 1994.

[27] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[28] J. A. Hartigan, "Clustering algorithms john wiley & sons," *Inc., New York, NY*, 1975.

[29] R. Xu, J. Xu, and D. C. Wunsch, "A comparison study of validity indices on swarm-intelligence-based clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1243–1256, 2012.

[30] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*, pp. 911–916, IEEE, 2010.

[31] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[32] J. Brooks-Bartlett, "Probability concepts explained: Maximum likelihood estimation," 2018. https://towardsdatascience.com/probability-concepts-explained-maximum-likelihood-estimation-c7b4342fdbb1.

[33] N. Bolshakova and F. Azuaje, "Cluster validation techniques for genome expression data," *Signal processing*, vol. 83, no. 4, pp. 825–833, 2003.

[34] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420, 2007.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[36] "Protein-protein interaction networks functional enrichment analysis." https://string-db.org/. Accessed: 2020-04-19.

[37] "The arabidopsis information resource." https://www.arabidopsis.org/. Accessed: 2020-04-10.

[38] A. McDermaid, B. Monier, J. Zhao, B. Liu, and Q. Ma, "Interpretation of differential gene expression results of rna-seq data: review and integration," *Briefings in bioinformatics*, vol. 20, no. 6, pp. 2044–2054, 2019.

[39] S. Dudoit and R. Gentleman, "Cluster analysis in dna microarray experiments," *Bioconductor Short Course Winter*, vol. 506, 2002.

[40] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proceedings of the 2005 SIAM international conference on data mining*, pp. 138–149, SIAM, 2005.

[41] D. M. Blei, "Hierarchical clustering," 2008. http://www.cs.princeton.edu/courses/archive/spr08/cos424/slides/clustering-2.pdf.

[42] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: 1. hierarchical systems," *The computer journal*, vol. 9, no. 4, pp. 373–380, 1967.

[43] G. J. Szekely, M. L. Rizzo, *et al.*, "Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method," *Journal of classification*, vol. 22, no. 2, pp. 151–184, 2005.

[44] R. Nishii, "Maximum likelihood principle and model selection when the true model is unspecified," *Journal of Multivariate analysis*, vol. 27, no. 2, pp. 392–403, 1988.

[45] T. B. M. Paul D.McNicholas, "Model based clustering of microarray expression data via latent gaussian mixture models."

[46] T. Thinsungnoena, N. Kaoungkub, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, "The clustering validity with silhouette and sum of squared errors," *learning*, vol. 3, p. 7, 2015.

[47] H. ŘEZANKOVÁ, "Different approaches to the silhouette coefficient calculation in cluster evaluation,"

[48] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[49] S. Romano, J. Bailey, V. Nguyen, and K. Verspoor, "Standardized mutual information for clustering comparisons: One step further in adjustment for chance," in *International Conference on Machine Learning*, pp. 1143–1151, 2014.

[50] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[51] V. Spruyt, "How to draw an error ellipse representing the covariance matrix,"

[52] G. Gan and M. K.-P. Ng, "K-means clustering with outlier removal," *Pattern Recognition Letters*, vol. 90, pp. 8–14, 2017.

[53] M. J. Garbade, "Understanding k-means clustering in machine learning," 2018. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

[54] M. J. Warrens and H. van der Hoef, "Understanding partition comparison indices based on counting object pairs," *arXiv preprint arXiv:1901.01777*, 2019.

[55] L. E. Peterson, "Clusfavor 5.0: hierarchical cluster and principal-component analysis of microarray-based transcriptional profiles," *Genome biology*, vol. 3, no. 7, pp. software0002–1, 2002.