CHALMERS | GÖTEBORG UNIVERSITY

MASTER'S THESIS

A Pipeline for Comparison of Clustering Methods in Flow Cytometry Analysis

Including Ensemble Clustering

Marjan Farahbod

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Thesis for the Degree of Master of Science

A Pipeline for Comparison of Clustering Methods in Flow Cytometry Analysis

Including Ensemble Clustering

Marjan Farahbod

CHALMERS | GÖTEBORG UNIVERSITY

Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Göteborg, April 2013

Mathematical Sciences
Göteborg 2013

A Pipeline for Comparison of Clustering Methods in Flow Cytometry Analysis

Including Ensemble Clustering

Marjan Farahbod

April 19, 2013

Abstract

This thesis is on applying different clustering algorithms and comparing their results with ensemble clustering in a Flow Cytometry(FCM) data analysis pipeline. Although FCM has been used for several decades in both clinical and research labs, there has not been many applications of bioinformatics in this field. The advances in technology and biochemistry has led to massive improvements in the field of FCM through the last decades, which has brought fast generation of high throughput data. However, the analysis of FCM data is mostly done manually and lack of computational techniques have become more evident. One of the most challenging aspects of FCM analysis is **gating**, which is the recognition of different cell populations in a tissue sample based on light scatter and fluorescent data collected from each cell. In the last few years, several techniques of automatic gating have been introduced, mostly based on popular clustering algorithms.

The evaluation of clustering algorithms is still considered to be a challenging issue, these algorithms are unsupervised learners and therefore, there are no pre-defined labels for each input to determine perfect results. In this project, a classification pipeline of FCM analysis is introduced and used to compare the results of the clustering algorithms. Clusterings are used in the feature extraction part for the classifier and the better the results of the classification are, the more practical clustering is considered to be. However, this inference can not be automatically generalized in this field but is an admissible method of specific performance comparison among different clustering techniques.

Apart from individual clusterings, a voting technique of **Ensemble Clustering** was applied which similar to voting technique in combining classifiers. To investigate the application of ensemble clustering, the results of classifications which used individual well known clustering approaches in FCM for

feature extraction were compared to each other and also to the results of classification which used some ensemble of these clustering algorithms. To be able to get more generalize conclusions, this pipeline was tested on two different FCM data sets, the data from controls and patients with Acute Myeloid Leukemia and data from DLBC [1] and Follicular Lymphoma. Also for each data set two different classifiers were tested and results were compared. The challenges of implementing this pipeline and the comparison of different clustering techniques on two data sets are discussed. It is also shown that although the results of this ensemble technique were generally better than the results of individual clustering methods, that is not always the case and the result depends on the choices of clusterings. Three individual clusterings were used, k-means, flowMeans and SamSpectral. The SamSpectral clustering was applied on the data with several different parameters, as it is a parametric algorithm and performs differently when its parameters are changed. Among these clustering methods, the popular k-means resulted in a significantly better classification for both of the data sets, followed by the ensemble clusterings and the SamSpectral clustering algorithm with the best choice of parameters.

keywords: FCM, Clustering, Ensemble Clustering, Classification, Cancer

[1] Diffuse Large B-Cell

Acknowledgments

Foremost, I would like to express my sincere gratitude to Prof. Olle Nerman, my coordinator in Bioinformatics program in Chalmers University of Technology, for being a great teacher and a patient advisor during my master studies and my master thesis.
My sincere thanks goes to Dr. Ryan Brinkman, who has been my thesis supervisor and it would not have been possible to do this project without his patience, understanding and knowledge.
I would like to thank all my friends in Chalmers, Gothenburg and also in BC Cancer Agency, Vancouver, for all the great time and good memories during my master studies and my thesis.
Last but not least, I would like to thank my family. My parents, Azra and Nemat, who brought me up to love and enjoy knowledge, wisdom and science. They have been there for me unconditionally and believed in me all through the hard times and gave me the strength to move on.

Contents

1 Introduction **7**

2 Background **8**
2.1 FCM 8
2.2 Applications of FCM 9
2.3 Analysis of FCM Data 9
2.4 Gating 10
2.5 Manual Gating 10
2.6 Automated Gating 10
2.7 Classification in FCM 12
2.8 Ensemble Clustering 12

3 Materials and Methods **15**
3.1 AML Data 15
3.2 The Lymphoma Data 15
3.3 Pre-processing 16
3.4 Clustering Algorithms 16
3.5 Cluster Matching and Feature Extraction 19
3.6 Classification and Evaluation 20
3.7 Mathematical Representation of the Pipeline 20

4 Results and Discussion **23**
4.1 The AML dataset 23
4.2 The Leukemia results 24

5 Conclusion and Future Work **25**

List of Figures

1 FCM analysis pipeline 27
2 Classification pipeline 28
3 Flowchart of classification using clustering 29
4 Feature Extracion for train and test data 30
5 Lymphoma MLP Heatmap 31
6 Lymphoma SVM Heatmap 32
7 AML SVM Heatmap 33
8 AML MLP Heatmap 34
9 Lymphoma MLP boxplots 35
10 Lymphoma SVM boxplots 36
11 AML SVM boxplots 37
12 AML MLP boxplots 38
13 List of the clusterins on AML data 39
14 List of the clusterings on Lymphoma data 40

List of Tables

1 List of the markers for the data sets. *AML*, *Lymphoma* 41
2 Mean for the AUC results of the clusterings on AML data . . 41
3 Mean for the AUC results of the clusterings on Lymphoma data 41

1 Introduction

Gating or identifying the discriminative cell populations in FCM data is one of the most challenging steps of the analysis. In order to automate gating, many clustering algorithms have been introduced. Mostly the goal of these algorithms is to be able to generate the results which are the most similar to the manual gating results, taking the manual gating as a gold standard in this field [27]. However, considering several drawbacks of the manual gating which are discussed in the background section, in order to examine the performance of different clustering techniques including Ensemble Clustering here, we used another method of evaluation which is using the classification results of the FCM data sets. Each FCM data file in these data sets is obtained from a FCM experiment on a tissue sample of a patient, knowing the results of diagnosis for each patient-which is either the type of the cancer between the cancer patients or if the sample is from a healthy or suffering patient- we use the results of clusterings for the feature extraction and then train and test the classifiers feeding these features and the labels.
The aim of this research was to develop a pipeline of evaluation for different clustering algorithms, which enables us to compare the results of different clusterings including Ensemble Clustering. Having the FCM files, the pipeline includes the preprocessing and normalization, clustering and feature extraction and classification.
The Ensemble Clustering techniques have been introduced and used on different data sets for more than a decade. The idea of combining several clustering algorithms in order to obtain better results comes from its sister field, classification, in which combining different classifiers is quite popular and has shown to improve the performance generally [33] [30]; however bringing this approach to the clustering field has been challenging. Here we used a simulated voting algorithm from Hornik [15], which modifies the voting system of combining classifiers and applies it for combining clusterings. We applied this pipeline on two data sets to be able to have a better comparison of different methods. Also, in the classification phase, two different classifiers were tested, with the set of the same clustering algorithms.
The FCM experiment and analysis are discussed in the background section, followed by a review on variant techniques of gating. Classification in FCM and use of voting technique for combining clusterings is also discussed in the background. In **Materials and Methods**, the analysis pipeline implemented in this project is explained in details. Comparison of the different clustering algorithms used in the pipeline are discussed in **Results and Discussion**.

2 Background

2.1 FCM

History

The term **Cytometry**[2] is the process of applying measurements to cells and studying their physical and chemical characteristics. **FCM**[3] refers to the application of these measurements to the cells contained in a fluid. The first flow cytometers were used in 1960 [29]. Since then, it has been used in research labs for studying different cell cultures and in clinical labs in pathology, for diagnosis of several types of cancers as well as many other diseases such as HIV [29]

The FCM experiment

In an FCM experiment, the suspension containing cells from a body tissue or a cell culture is exposed to biochemical markers. The cells in this suspension will pass through the laser interrogation points in a flow cytometer one by one. The 'reflections' from the laser beam are collected and converted to electronic signals which are digitized and sent to a computer to be saved in data files. The reflection from the forward light scatter indicates the size and shape of the cell and the side light scatter granularity gives information about the internal complexity of the cell. The fluorescence emitted from the cell surface defines the intensity of the markers on the cell membrane. [3]. Each row in an FCM data file has information of one cell, with columns of data for each marker and side and forward scatter.

Markers

FCM markers are mostly extracellular and bind to the proteins on the surface of the cells, but for some experiments intracellular markers are used as well. The number of markers to be used in a single experiment are limited and the numbers vary in different flow cytometers. Today there are flow cytometers which can measure up to forty markers in a single experiment. Markers are carefully chosen according to the studies done on the proteins exposed to the surface of different cells in the samples being tested. Since the number of markers is limited in each experiment, some samples are divided into

[2] *cyto* for cell and *metry* for measurements
[3] *flow* for fluid

numerous **tubes**. Each tube would be put through the experiment with different set of markers. Set of markers in different tubes usually overlap.

2.2 Applications of FCM

The ability to define subpopulations in a cell sample according to the size, shape and biochemical markers on the surface of the cell has made FCM a great tool for both clinical and research labs. It has been used for diagnosis and prognosis of Leukemia and Lymphoma and also in peripheral blood hematopoietic stem cell studies [35]. It has also been used in environmental sciences for the identification of new species. [18] [22]

2.3 Analysis of FCM Data

For each tube of the sample, the FCM technique generates multidimentional data for individual cells in that sample [17]. Nowadays, flow cytometers can generate data samples of up to a million cells for up to forty markers. However, the FCM files used for this project include data for tens of thousands of cells with 7 and 11 markers per tube. [4]

The goal in many FCM analyses is to identify cells in heterogeneous populations. These different cell populations defined within a cell sample become fingerprints of that sample for further analysis. In pathology, cell samples with similar populations are to be diagnosed with the same disease. Similarity between cells is determined by their size and shape and also the markers attached on their surface. Certain types of markers are used to diagnose or differentiate between cancers. As an example, finding a population with the CD22 marker means there is a $> 90\%$ chance that the sample comes from a tissue with T-cell Lymphoma[5] [13]. Cell population discrimination is known as *Gating*. Gating analyses are not usually straight forward. The values assigned to each cell for a particular marker are continuous and therefore to call a cell *positive* with a marker in a data set needs delicate processing. Also, once the *positive* cell populations are recognized in a sample, the diagnosis could not be made at 100% for just one marker. Many markers are not highly discriminative and could be present in various sub-types of a cancer. Gating analysis are mostly unsupervised and are done to discover new cell types or phenotypes of a known disease.

[4] there were two data sets used in this experiment which are discussed in details in **Materials and Methods**

[5] the Lymphoma that affects T-cells

2.4 Gating

Gating refers to defining different subpopulations within a sample tube. This subpopulation identification would be used as a characteristic of the sample. Gating is generally acknowledged to be one of the most powerful but also one of the most problematic aspects of FCM according to its subjective nature [12]. It is traditionally done by biologists by drawing boundaries (*gates*) in two dimensional projections of the data. Recently, there have been some automatic methods introduced for this stage of analysis [26] [11] [19] [35] [23]. These methods are mostly based on famous clustering algorithms like k-means.

2.5 Manual Gating

Manual gating is done by drawing boundries around the recognized clusters in the one or two dimensional projection of the data by researchers [26], or it is done based on density distributions. Since the number of channels are more than two, manual gating is done in a hierarchy. At each level only two dimensions are plotted and in the next level, the populations found in the previous plot, are plotted in the next two dimensions separately and new populations are recognized. Therefore the number of cells plotted at each level of the hierarchy are reduced.
Manual gating is a labor intensive process and the result varies depending on user experience and intuition [26] about the data and markers. Since it is done by individuals and is highly dependent on their experience in the field, it is not easy to reproduce. Also, since the data is visualized in just two or three dimensions at each level, the high-dimensional features can not be recognized. These drawbacks of manual gating can be summarized as *subjectivity*, *lack of efficiency* and *loss of information* [26].

2.6 Automated Gating

The drawbacks of manual analysis of FCM data have brought up the need for using automated analyzing techniques, among which are several clustering algorithms used for gating. Based on their unsupervised learning nature, clustering algorithms were reasonable choices for automated gating processes. However, most of the clustering algorithms used in FCM are based on known clustering techniques and have been through some modifications in order to meet the technical requirements of FCM analysis. By 2008 there were considerable number of algorithms developed for FCM analysis and many successful results were being reported. Murphy et al. [23] used k-means clustering and

reported good results. K-means clustering is a rather fast algorithm and is easy to implement and apply to FCM data considering the size of FCM data files. However, it has some shortcomings when used on non-elliptic data as it is a centroid based clustering technique. Also, it needs the number of clusters to be given as a parameter, hence it is not a proper choice for discovering unknown sub populations. Therefore, the use of more complex clusterings became vital in the field. In 2008 Chan et al. [7] used statistical clustering with Guassian mixture model on four color FCM data. Lo et al. [19] used t mixture model to generalize the previous guassian mixture model and therefore find non-elliptical subpopulations as well. The *FLAME* algorithm by Pyne et al. [26] also uses a distribution based clustering and models data into a *skew t mixture model*, where for each population the mixtures of 2 to 20 skew t distributions are modeled. The parameters of individual distributions in each mixture are calculated by Maximum Likelihood estimations via the Expectation Maximization(EM). For each sample the best of these models is chosen according to their Scale-free Weighted Ratio (SWR) which is the ratio between the intracluster distances and the intercluster distances. Although distribution based algorithms has been quite popular and practical in this field, there have been other noticeably good algorithms. SamSpectral [35] used a modified spectral algorithm and reports acceptable results for diagnosing subtypes of Lymphoma.

In Critical Assessment of Population Identification Methods (FlowCAP1) [27] a set of both centroid-based and density-based clustering algorithms including some of the above were compared against each other in four challenges, having the manual gating results as the reference. Challenges comprised *Completely Automated Algorithms*, where the algorithms were either parameter free or their parameters were set independent of the data sets; *The Manually Tuned Algorithms*, where the parameters could be set according to different data sets; *Assignment of Cells to Populations with pre-defined Number of Populations*, where the number of populations were given to the algorithms and finally the *Supervised Approaches Trained using Human-Provided Gates*, where 25% of the files with manual gating results were provided to participants for training their algorithms [27]. Some of the algorithms presented in FlowCAP1 are now available via *R* packages, among which are SamSpectral, FlowMeans and FlowCLUST.

When choosing a clustering algorithm, there are two challenges to overcome. First, many clustering algorithms are based on random initializations and have stochastic learning methods [8]. The famous k-means clustering chooses the first set of centers randomly and therefore performs differently each time

it clusters the same data. Secondly, most of the clustering algorithms have some initial parameters to be set. Choices of parameters affect the performance of the clustering and comprises both the results and execution time. Therefore, parameters must be chosen based on the type of data and the kind of populations one is looking for. Performing a k-means algorithm on FCM Data when we are looking for the small new population, requires putting a large k(number of clusters) as an input parameter. However, k-means algorithm does not work well in finding non-globular clusters and therefore is not considered to be a good option for the FCM analysis, where many of the clusters are non-globular.

2.7 Classification in FCM

In most of the clinical practice of FCM in cancer diagnosis, the goal is to identify the discriminative cell populations in each type of cancer and *classify* the examined samples by their type of cancer. Over the last two decades, FCM classification was done via the comparison of the light scatter profiles of control and patient groups [32]. These kinds of classifications were done by simply comparing the plots of different samples, hence it was mostly done manually. Gating was included in these analyses in order to identify common cell populations among different samples. However, with the advance of computational technology and its application in biology, the analysis of FCM could benefit from various machine learning algorithms and statistical analysis. Clustering algorithms could be used to overcome the drawbacks of manual gating, but also many types of classification algorithms which are known as *supervised learners* could be applied to FCM Data. By having the results of the earlier diagnosed samples of different types of cancers and using the proper feature extraction methods for their FCM data, one could train a classifier to discriminate between different cancer types.

In 2008, Pedreira et al. [25] implemented a multidimensional classification by using a divide and conquer approach on four color FCM data from peripheral blood lymphocyte samples; however, like most of the classifications used in FCM so far it is based on unsupervised statistical machine learning techniques.

2.8 Ensemble Clustering

The idea of using a method of combined clusterings in order to obtain better results when solving a clustering or unsupervised learning algorithm comes

from its sister field **classification** or supervised learning, where several techniques of combining classifiers have been introduced and are generally known to improve the results.

When having a complicated problem on which the individual classification algorithms do not perform well, one approach would be to use several algorithms and benefit from putting together their results through some *combining method* [30], hoping that one's strength would cover the others weakness and thereby reduce the overall error. As in classification, there are many different forms of applying a combining method when one is trying to benefit from several clusterings in a complicated problem.

Regarding the data, for reasons such as the different performance of clusterers on particular type of data or limited resources [31], the individual clusterings in the ensemble could be applied to different parts of the data. This does not create any difficulties for combining part since most of the combining techniques need only the labels and some information from the results of each algorithm and do not rely on having the original data points. Basically, the data could be divided based on the *feature set* or *data points*. The former is, where for the same data different subsets of features are given to each clustering. These subsets of features could have overlaps or be disjoint. This is most common when you have features of various resources, or features of different types. The latter, however, is when different subsets of data points are given to clusterers for clustering, while each of the data points have all the features. However, using the whole data set for each clustering algorithm is expected to improve the results of clustering as well.

There are three methods of combining clusterings[6] introduced in [31]. Their first method, named **Cluster-based Similarity Partitioning Algorithm-CSPA**, defines a pairwise measure of similarity between the objects, based on the counts of their presence in the same cluster. As simple as this method is to implement, since it requires the memory of the order $O(n^2)$ of the input, it becomes impractical for the large input. Therefore, it is not convenient for FCM data where the number of objects to be clustered could be more than hundreds of thousands. Their second approach is **HyperGraph Partitioning Algorithm-HGPA**, based on a hypergraph presentation of the clusterings and defining n-way relationship between the data points.

[6] the combining function is also called *consensus clustering* or the combiner in some of the litreture

Diversity in the approach. Inorder to achieve the most results from the combined approach, one does not only look for clusterings which perform well individually, but also tries to bring the most diversity possible among them. We cannot expect to get much improvement in our combination if all the individual algorithms are quite similar and cluster the data in the same manner. It is only with a great diversity between the individual algorithms and using a proper combining method that one could except better performance using an ensemble of clusterings. Having diversity could be achieved by using different clustering methods (e.g., using a mixture of a centroid based clustering and a density based) or applying the same methods of clustering with different input parameters.

Voting has been one of the well-known methods for combining classifiers [33]. Although voting schemes are rather straightforward procedures in combining classifiers, when it comes to clusterings they are not as simple. The difference between classification algorithms and clustering algorithms brings up certain issues to be taken care of. Unlike a classification problem, there are no predefined labels for the data in a clustering and therefore there is a problem of defining which cluster of one clustering algorithm corresponds to which clusters in the other algorithms, the problem is even using the same method of clustering could not result in the same set of labeling. [8] Therefore, to apply voting in combining clusterings, first we need to match the same clusters from different algorithms and then we can compare the labels the algorithms have assigned to a particular data point and vote between them.

3 Materials and Methods

There are two pipelines of FCM analysis. While the goal in **Gating** is to identify cell populations, in **Classification** the goal is to find the discriminative features between the samples of different type (e.g., different cancer types). Several methods of clustering could be applied in both of these pathways (Figure1).

The pipeline coded and used for this project is a classification pipeline which uses clustering for feature extraction. The general framework of the pipeline is shown in Figure2, which also includes the steps of the **Ensemble Clustering**. There are four major steps in the pipeline (Figure 3), however there are some minor differences between the *training* pipeline and *testing* pipeline, regarding the **Cluster Matching and Feature Extraction** phase.

3.1 AML Data

The AML[7] dataset was used in FlowCAP2 competition [27], where the participants presented their automated analysis methods for different challenges in FCM analysis, including the gating. The manual gating labels were used to rate the performance of each algorithm. The AML challenge was to find cell populations that can be used to discriminate between AML positive and AML negative patients. Peripheral blood or bone marrow aspirate samples were collected over a one year period using eight tubes, from 43 AML positive patients and 316 healthy people [11]. Tubes were different in their choice of markers. The data from each tube could be analyzed separately. In this experiment the data of tube two was used. This tube was chosen randomly and the comparison of the results from different tubes of a single FCM experiment was beyond the scope of this project. Tube two had 7 markers which listed in Table11

3.2 The Lymphoma Data

The DLBCL[8] and Follicular Lymphoma data is the data from 118 patients, 33 of which were diagnosed with DLBCL and 85 were diagnosed with a type of Follicular Lymphoma. The FCM experiment on these patients had 11 markers, the markers and information about data sets are given in Table11

[7] Acute Myeloid Leukemia
[8] Diffuse Large B-cell Lymphoma

3.3 Pre-processing

Based on the data set, FCM files need some pre-processing. There are three steps of pre-processing which are commonly applied to FCM files, biexponential transformation, compensation and normalization. FCM data is usually presented using a log-scale, but some of the data points are on or below the axis with this presentation, therefore a biexponential transformation is applied to make use of all the data points. Also, signals from different markers in an FCM experiment could have overlap with each other, which generates problems in both manual and automated gating. To fix this, there is usually a matrix provided with the FCM files, called the compensation matrix, which is a transformation matrix used to eliminate the effects of overlapping signals.

The AML dataset was already compensated and did not need such a process, for the biexponential transformation function *estimateLogicle* and *transfrom* where used in R. Normalization was also applied to transform the data within the range of $[0, 1]$. The same tools were used for the Lymphoma data set; however, it also needed to be compensated.

3.4 Clustering Algorithms

There were three individual clustering algorithms used on this pipeline for each data set: flowMeans, KM and SamSpectral. SamSpectral [36] is a parametric clustering and it was applied with different input parameters. Based on the results of the classification, which is considered to be the evaluation of these clusterings, sets of individual clusterings were chosen to be used in the **Ensemble Clustering**(Figures14,13).**Ensemble Clustering** technique used here is from package CLUE in R. Package CLUE was released in 2007 by Kurt Hornik, for creation and analyzing cluster ensembles. The ensemble functions is easy to apply for different datasets with minor modifications, introduced in the package manual. Results of the clustering algorithm for each file are saved separately, so that they could be fetched in the next step which is *Cluster Matching and Feature Extraction*. The output of these algorithms, including the *Ensemble Clustering*, are not the same; however, all of them return a vector of the clustering labels for the data containing an integer label for each data point. Given the labels and the data file itself, centers and sizes of the clusters for each file are calculated and found. The center of a cluster is the mean of the data points in that cluster. The vector of labels along with the center and size of each cluster are saved for each file.

flowMeans Clustering. flowMeans is a non-parametric FCM clustering

which is based on k-means clustering but unlike k-means, it allows for concave clusters, by using several clusters to model a single population. It also finds the number of clusters by taking the number of modes found individually in every eigenvector of the data [1]. flowMeans clustering is available via the R package flowMeans [2]

K-means Clustering. K-means clustering was applied to the data using R function kmeans. k-means is a parametric clustering and needs the number of clusters to be defined as an input parameter. The number of clusters were given 15 for both of the data sets. This number was given after observing the dot-plots of the data in experimental k-means results. From each data set two frames[9] were chosen to observe the results of k-means clusterings for different k on the dotplots[10]. Most of the populations were identified when k was chosen at 8, 10 or 12, but 15 was chosen since it did not affect the identification of large cell populations negatively, while it could have improved the chance of finding small discriminative populations.

SamSpectral Clustering. [37] [36] SamSpectral is a parametric spectral clustering. Its performance is tuned mostly by two parameters which are **normal-sigma** and **separation-factor**. Normal-sigma is a scaling parameter, increasing it results in recognizing more the smaller clusters and it could be any integer from one to several hundreds. However, depending on the data set, choosing a large normal sigma could make the algorithm impractical due to its computational complexity. Separation-factor, on the other hand, controls the combining phase of clustering, where the smaller clusters are merged together and make up the final clusters. It defines the extent to which clusters should be kept separately or be merged together. According to the manual for SamSpectral R package [37], an appropriate range for the separation-factor is $[0.3-2]$. In order to find the proper input parameters for the SamSpectral algorithm, dot-plots of different combinations of the parameters on three different FCS files for each data set were made and observed. Plots were made for 17 normal sigma from 0.1 to 1.8, for every 0.1 and for 12 different separation-factor, starting from 80 to 1040 for every 80.

Clue. [15] Is an ensemble clustering algorithm which is implemented in the R package clue [14]. The algorithm resembles a voting scheme similar to voting technique for combining classifiers [34]. Having the below:

[9] For AML data one frame with AML positive and one with AML negative were chosen. For Lymphoma data one frame with DLBCL and one frame with Lymphoma were chosen

[10] k were chosen at 6, 8, 10, 12, 15, 18

- $c_{i,j}, i \in 1, ..., n$: cluster i in the clustering j and n is the number of clusters.
- $C_j, j \in 1, ..., m$: clustering j where m is the total number of clusterings to be combined.
- $D_l(c_v, c_r)$ the similarity distance between two clusters with the same label in clusterings v and r, having the labeling l.
- $D_{total,l} = \Sigma_{labeling} D_l(c_v, c_r)$
- labeling l : each permutation of labels in a clustering.

When combining different clustering results, it must be considered that the same cluster in different clusterings might have different labels. Therefore, combining clusterings has two steps. Step one is to optimally match the similar clusters in different clusterings, in a way that the sum of differences $D_{total,l}$ between each cluster and its matched clusters among all the clustering methods in the ensemble are minimum. There are several methods for calculating $D_{total,l}$ and therefore choosing the labeling which minimizes it for *soft clustering* [11] algorithms. Our clusterings here are *hard clustering*, where each data point only belongs to one cluster in the clustering method. The method used is the *transfer distance*, which is for a set of labelings the minimum number of objects to be removed so the clusterings with the left objects are identical. Therefore, applying different permutatoins of labelings to the clusterings, the one which results in the minimum transfer distance is chosen.
Once having the same labels for the most similar clusters in differnt clusterings, the second step is to combine them into one labeling and clustering result. Combining technique here was *voting* which is basically to choose the label which is most common among all the clusterings for each object and return it as the label of that object.

Having the labels for each file from CLUE and using the FCS file itself, centers of the clusters and their sizes are calculated using two functions **findClustersCenters.R** and **getClusterSizes.R**. The center of a cluster is the vecter of the mean values for the markers of the cells on that cluster. Size of

[11] *soft clustering* algorithms or *Fuzzy clustering* are clusterings where each object belongs to a cluster with a value from $[0, 1]$ which sums up to 1 for each object. Manhattan partition dissimilarity, Euclidean distance, angle and diag are the methods named and briefly explained in the package

a cluster is the number of cells in that cluster.

3.5 Cluster Matching and Feature Extraction

Each file represents a sample to be classified and therefore has a feature vector based on the results of the clustering. This feature vector is obtained by the features of individual clusters it has. The features with which a cluster is described or presented could vary by the method of clustering it was clustered with, for example it could be the parameters of the distribution, if we use distribution based clustering techniques. Here we chose the center and sizes of clusters for the features, since we are using different method of clustering and these two features could be calculated fast and easy regardless of the method of clustering used. Therefore, for each data file, features are the percent of the total data points in each of the clusters. Percentage was used instead of the actual size since the number of cells in different files are not necessarily the same. However, since each file is clustered individually, even the same clusters, which are the cell population with the same characteristics or **homologous cell populations**, would not necessarily have the same cluster labels in different files and we needed to have consistency among the labels. This means that we need the homologous clusters to have unified labels among all the files. The problem of matching homologous clusters to each other among different data-sets (here each file is a data set) is known as **Cluster Matching**. The cluster matching phase is different for train and test pipelines. For training, the pipeline fetches the whole training data files and apply the cluster matching, which is done by clustering the cluster centers in the train data files altogether. Centers which are clustered together would obtain the same label among all the files and the clustering information for each data file would be changed accordingly. This means summing up the sizes of two clusters in the same file if they gained the same label in the cluster matching and therefore not every file would have members in all the clusters. The clustering algorithm for the cluster matching could be any clustering algorithm; here we used K-means clustering. As the number of the clusters should be given as a parameter to the K-means algorithm, this number was defined as twice the number of the clusters the data file with the most clusters. Centers of the cluster matching from the training phase -centers of the centers of clusters- would be kept and used as a frame for cluster matching for the test files. Since the files go through the test phase individually and the classifiers are trained with the training feature sets, the centers from the cluster matching phase in the training is used to obtain features of the test files. For each cluster center in a test file, the cluster

label would be the label of the cluster from from the train cluster matching which has the minimum euclidean distance to it (Figure4).

3.6 Classification and Evaluation

In order to have a better comparison of the clusterings and study the consistency of clusterings performance, the pipeline was applied with the same set of clustering algorithms with two different classifiers **Support Vector Machines - SVM** and **Multilayer Perceptron Neural Networks - MLP**, which are both popular as general purpose classifiers and are known to perform well with different data sets [5] [21]. To perform statistical analysis on the results of the classifiers, MLP classifiers were applied for 400 bootstrap sets of train-test samples and SVM classifiers were applied for 100 train-test samples, using 80% of the data for the train and 20% for test each time (Figure 13, 14). For classifiers, R function **train** and **test** from the base package were used, choosing the algorithms to be SVM and MLP accordingly. Number of internal nodes for the MLP classifier was 15, which was based on the size of the feature vector. For each train-test sample of the data, the AUC[12] [16] result of the test samples were saved into files, along the trained classifier and the resulting labels for the record. Two-sided t-test for the difference between the means were applied, with the null hypothesis being "true difference in mean is equal to zero".

3.7 Mathematical Representation of the Pipeline

An FCM dataset contains several .FCS files. Each file is a result of FCM experiment on a sample and includes a data matrix, where rows represent the cells and colums are markers. Terms used in mathematical explanation of the pipeline are defined as follow:

- $D = M^1, M^2, ..., M^m$: an FCM data set, containg several data matrix.
- M^i: data matrix i, where $i \in 1, ..., n$ and n is the number of files in the data set.
- c_j^i: cell j in data matrix i and $j \in 1, ..., p$ where p is the total number of cells(rows) in the data matrix.
- cc_l^i: cluster center for cluster l, in data matrix i.

[12] area under the curve

- cs_l^i: cluster size for cluster l, in data matrix i.
- MC^i: set of cluster centers cc^i and their sizes cs^i for a data matrix.
- M_{train}: a subset of D, including the data matrces used in the training phase.
- M_{test}: a subset of D, including the data matrices used in the testing pahse.
- $CL(M)$: a clustering fuctions, taking in a data matrix M^i, returning a vector of labels l for the cells, containing an integer cluster label for each cell c in the matrix.
- $CM()$: a cluster matching function. The application of this function is different for train and test data. **train:** $CM_{train}(d)$ taking a matrix of cluster centers d, clusters them and returns the labels for each center. This could be any clustering function. **test:** $CM_{test}(x, d)$ having the centers of several previously known clusters and their labels x, given the centers of clusters for one file d, it will assing to each cluster center in d, the labels of the closest cluster center in x(Figure 3)
- $SELECT(D, r)$ select function randomly choses a subset of r matrix from the set D.

Having had the above, the pipeline has 8 steps:

1. clustering each data file individually:
   for i in $1 : m$
   $MC^i = CL(M^i)$
2. selecting the train and test subset:
   $M_{test} = SELECT(D, r)$
   $M_{train} = D - M_{test}$
3. cluster matching for the train data:
   $result_{train} = CM(M_{train})$
4. cluster matching for the test data:
   $result_{test} = CM(M_{test}, result_{train})$
5. extracting feature matrix for train and test data
6. train the classifier with the train feature matrix

7. test the classifer with the test data

8. save the AUC for the test data

Steps $2-8$ where repeated 400 times for each triple of the dataset, clustering and classifier.

4 Results and Discussion

4.1 The AML dataset

The results of AML data was quite variant. Mean of the AUC was between 0.798 and 0.9582 for MLP and 0.768 and 0.972 for SVM classifier. Although the results for MLP are generally better, the maximum and minimum of AUC means belong to k-means and SamSpectral with normal-sigma 110 and separation-factor 0.1 for both classifiers. K-means clustering seems to perform significantly better than other clusterings(Figures1112). The results for other clusterings are similar for both classifiers as well, other than the CC03, CC04 and FM clusterings (Figure13). The CC04 ensemble clustering results are much better when used with the MLP classifier rather than SVM. Among the SampSPECTRAL results, the one with the higher separation-factor obtained better results. But using the greater separation-factor is not practical according to the computational complexities.

For the ensemble clusterings, results seem to be better than the individual clusterings and the ensembles performances are among the highest, however, they are still lower than the best clustering in the ensemble. K-means result is better than the results of all ensembles, including the ones it was part of them. This was not expected since according to FlowCAP1 [27] results, the ensemble clustering had better results than all the individual clustering. However, in FlowCAP1 results of the clusterings were compared to the manual gating results and were not compared to the classification results as we did here.

Ensemble techniques are used to improve the performance of clusterings, but based on the data and individual clusterings, they must be tuned carefully to perform well. In a voting ensemble technique when there is a particular clustering with significantly better results than all the other clusterings, it is better to use a *weighted* voting scheme, in which there is a weight assigned to the results of each clustering technique when building the ensemble based on the individual clustering results and as a result, better clusterings would have more impact on the final ensemble result than the other clusterings.

K-means clustering is a relatively fast clustering and also easy to implement and therefore is a popular clustering, but according to its two drawabacks which are being highly dependant on its input parameter k and only being able to recognize the spherical clusters, it is not the best choice for the FCM analysis, where the number of clusters are not known previously to

the analysis and the clusters are not neccessarily spherical. However, we see that with our data sets here and with a good choice of k, we could obtain acceptable results.

4.2 The Leukemia results

The AUC results from the Lymphoma data are less variant, having their mean between 0.683 and 0.869 for MLP and 0.728 and 0.816 for SVM classifier (Table3). But using k-means algorithm resulted in better AUC for both of the classifiers as well. Unlike the AML data, SVM classifier had better results here3 and there was more consistency among the classifier results generally. Also the best SamSpectral here was not the one with the greatest separatoin-factor(Figure**????**). The mean of Ensemble clustering results for all sets are less than the best clustering result and more than the worst one, showing that like the other data set, a weighted voting system might be a better approach.

Results here are similar to those from the AML data, with k-means clustering having the best classification performance, followed by the best SamSpectral and flowMeans. FlowMeans and SamSpectral results were of the best clustering when the standard was manual gating results in FlowCAP1. However, flowMeans algorithm has an advantage compared to SamSpectral and k-means and that is it could be applied as a non-parametric clustering and still generate relatively good results(based on both FlowCAP1 and this experiment), but SamSpectral algorithm would perform poorly if parameters are not carefully tuned. Even from a set of chosen parameters which was based on prior observation on the data and comparing several SamSpectral results, SamSpectral does not have be best performance.

5 Conclusion and Future Work

Unsupervised learners or clustering algorithms are a well known choice when it comes to categorizing the data sets where we do not have predefined labels for each datapoint. In FCM analysis, these algorithms are used to identify the discriminative cell populations among different samples, as a replacement for the manual techniques and to overcome their drawbacks. Although clustering algorithms are very practical for this field, due to their unsupervised method of learning, it is challenging to define a proper evaluation method for them. One approach is to set the manual analysis results as the gold standard and trying to achieve the results similar to manual gating, but in order to improve the results (even better than manual gating), we need other standards. In this experiment, through the previous results of diagnosis using FCM data, we built a classification pipeline, which used clustering algorithms for its feature extraction part by finding discriminative cell populations and the counts of cells in them using clustering algorithms. To take the analysis one step further, we also applied an ensemble clustering technique, to see how this technique could improve the results compared to the already known clustering algorithms in this field.

This pipeline was implemented to study the performance of different clusterings in FCM gating, knowing that gating is a critical and one of the most challenging parts of FCM analysis and although new clustering techniques have been introduced to this field recently, based on the diversity among the FCM data, it is hard to choose one of them as the ultimate clustering approach. Because of that, most of the clusterings introduced in this field are parametric and parameters are to be tuned based on the data. Ensemble clustering techniques have been used to improve the clustering results in complicated problem, here we used a modified version of popular voting technique for combining classifiers. Although the ensemble clustering result was better than most of the individual clusterings, it could not beat the best clustering results. Based on these results, applying a weighted voting scheme for ensemble clustering could be considered as a future work for this project. There are also other combining techniques which were briefly discussed here, which could be applied to FCM data.

Tables and Graphs

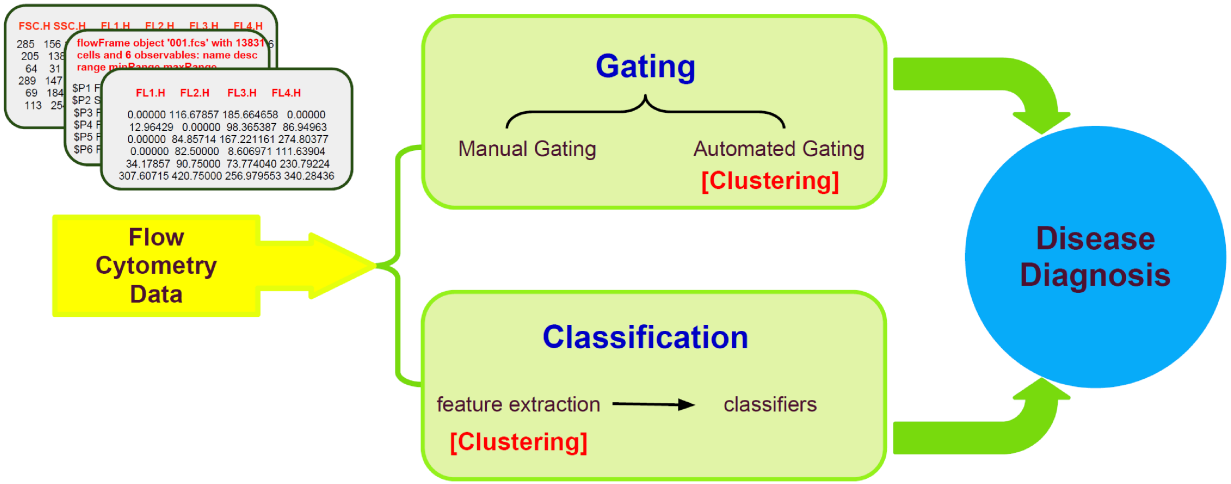


Figure 1: FCM analysis pipeline of gating and classification. FCM data come in .fcs files. Gating and classification are two paths in data analysis. Clustering could be used in feature extraction, for classification. Also, clustering is a way of automatic gating.

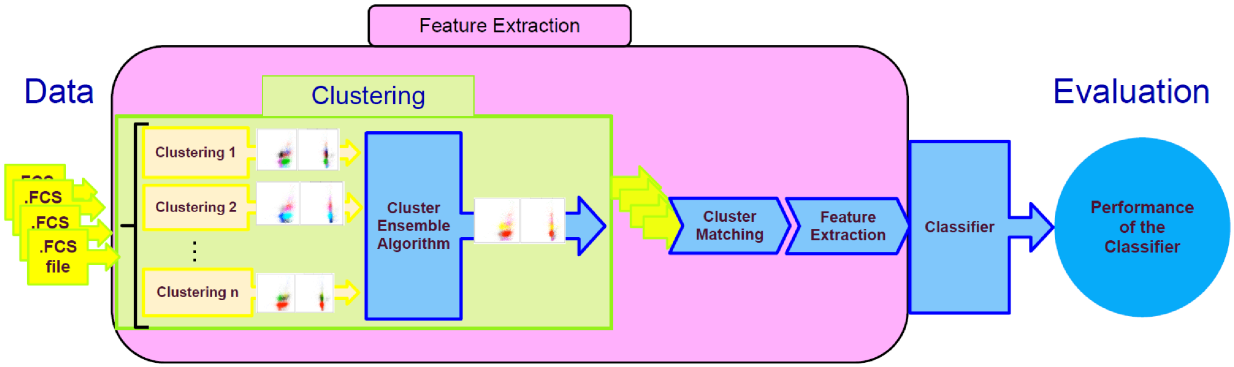


Figure 2: Data comes into the pipeline as .FCS files. Feature extraction includes three parts, clustering, cluster matching and finally feature extraction from the clustering results. Evaluation of the clusterings is done by comparing the classifier results.

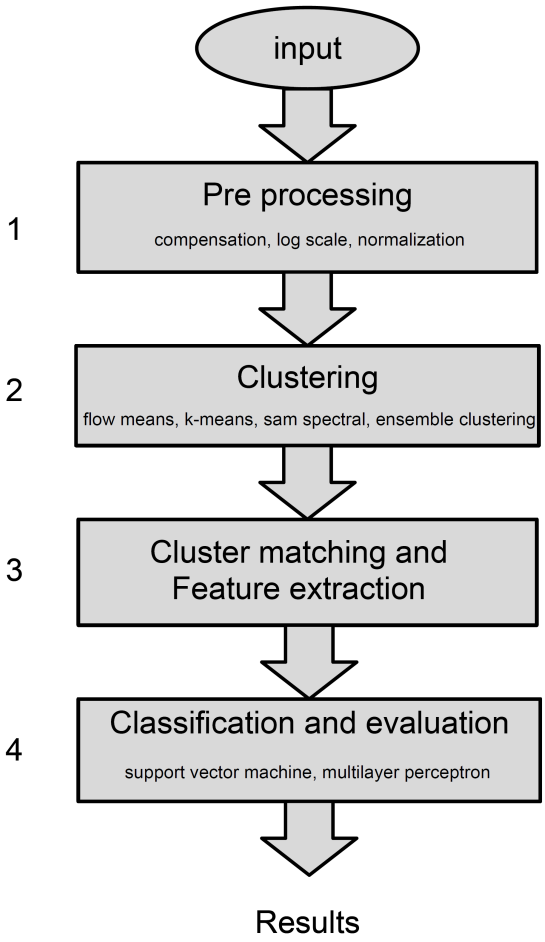
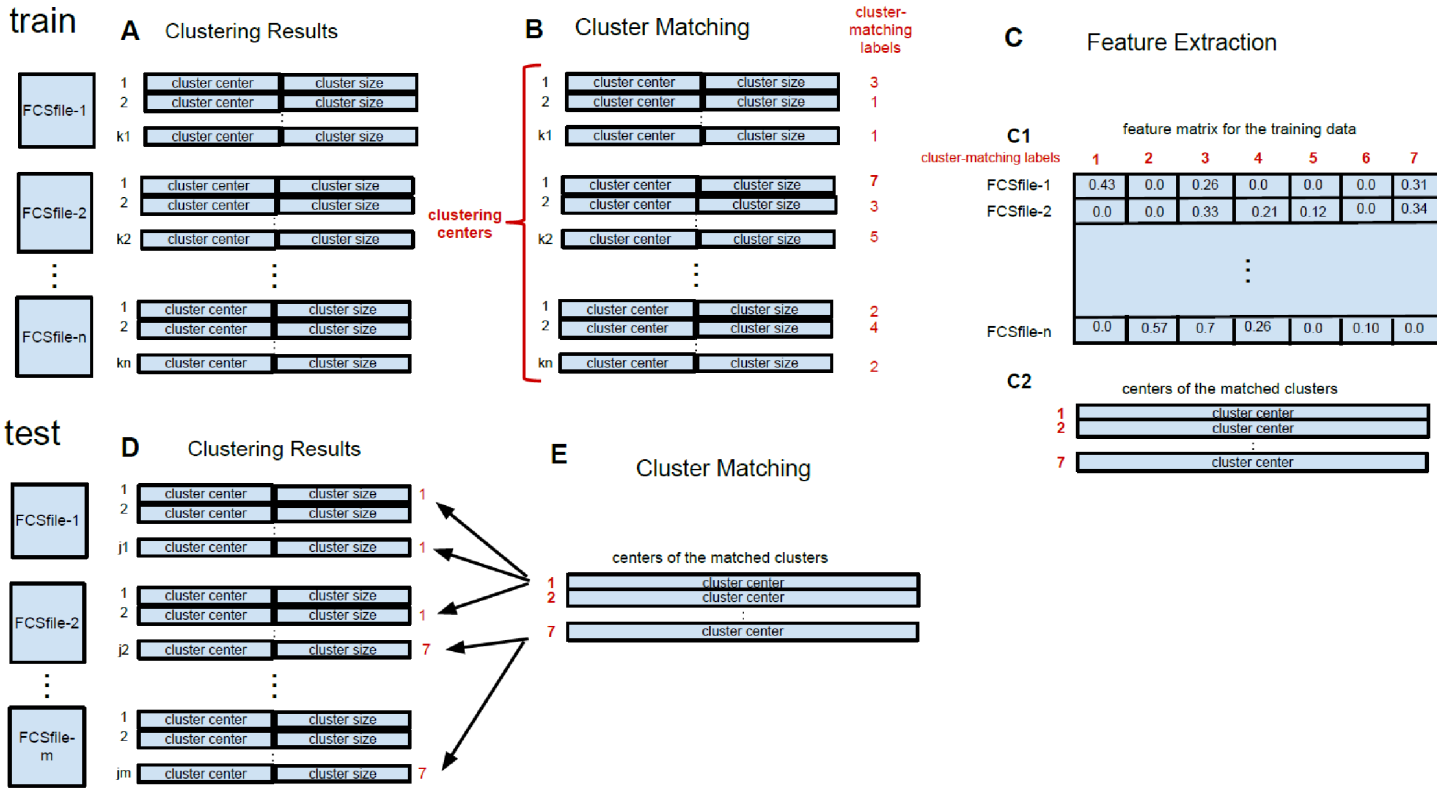


Figure 3: The pipeline has four modules.



book

Figure 4: Feature Extraction for train(A, B, C) and test(D, E). In train cluster matching is done by meta clustering through the centers of clusters. Center of clusters are saved to be used as reference for cluster matching in the train.

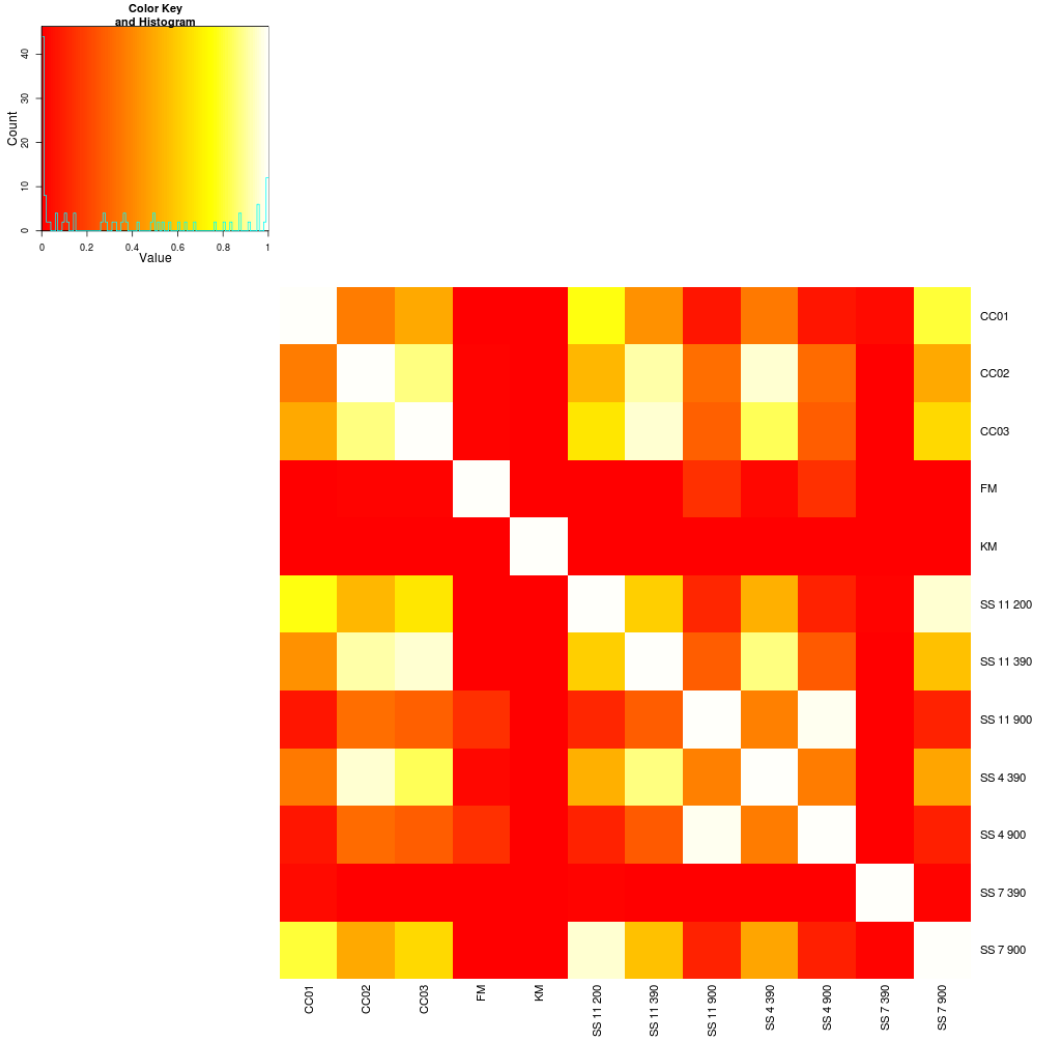


Figure 5: Comparison of p-values for the classification results on Lymphoma data, using MLP classifier

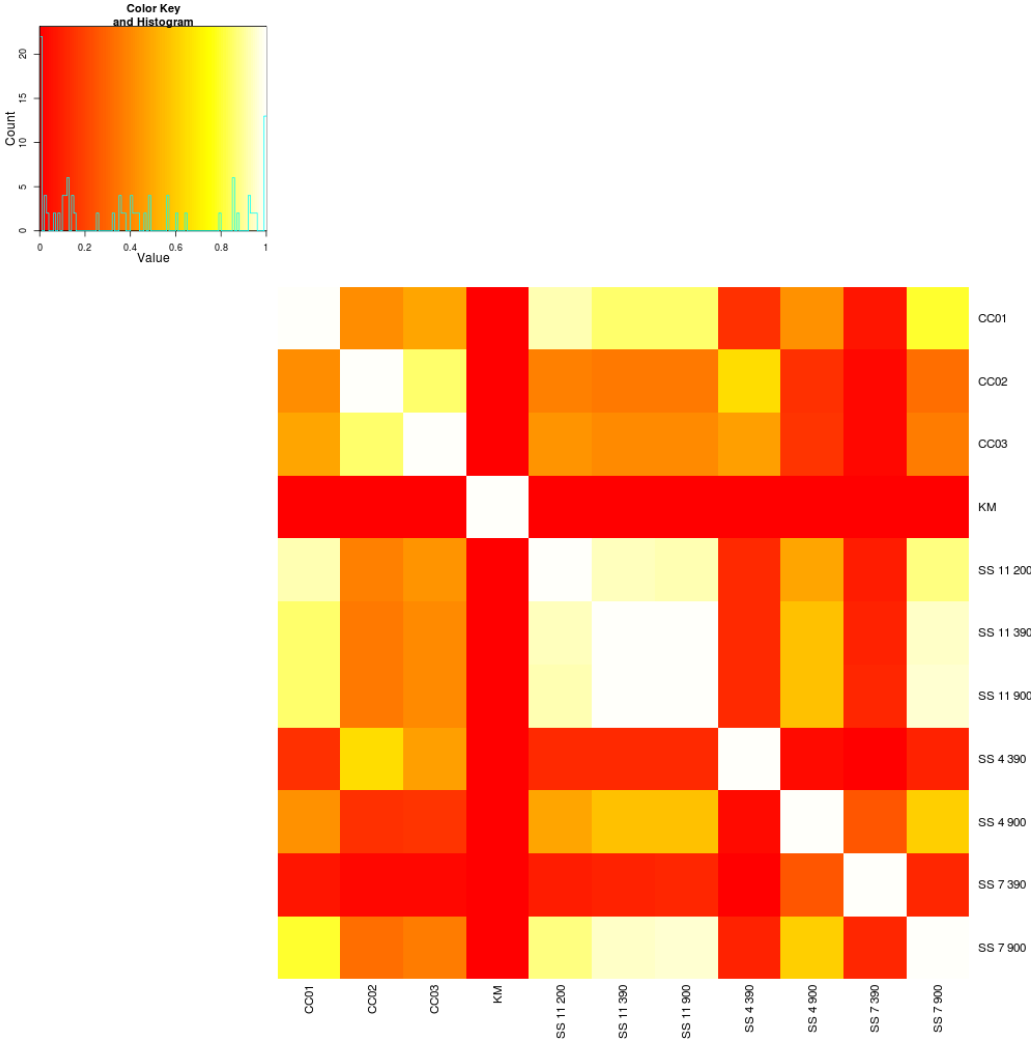


Figure 6: Comparison of p-values for the classification results on Lymphoma data, using SVM classifier

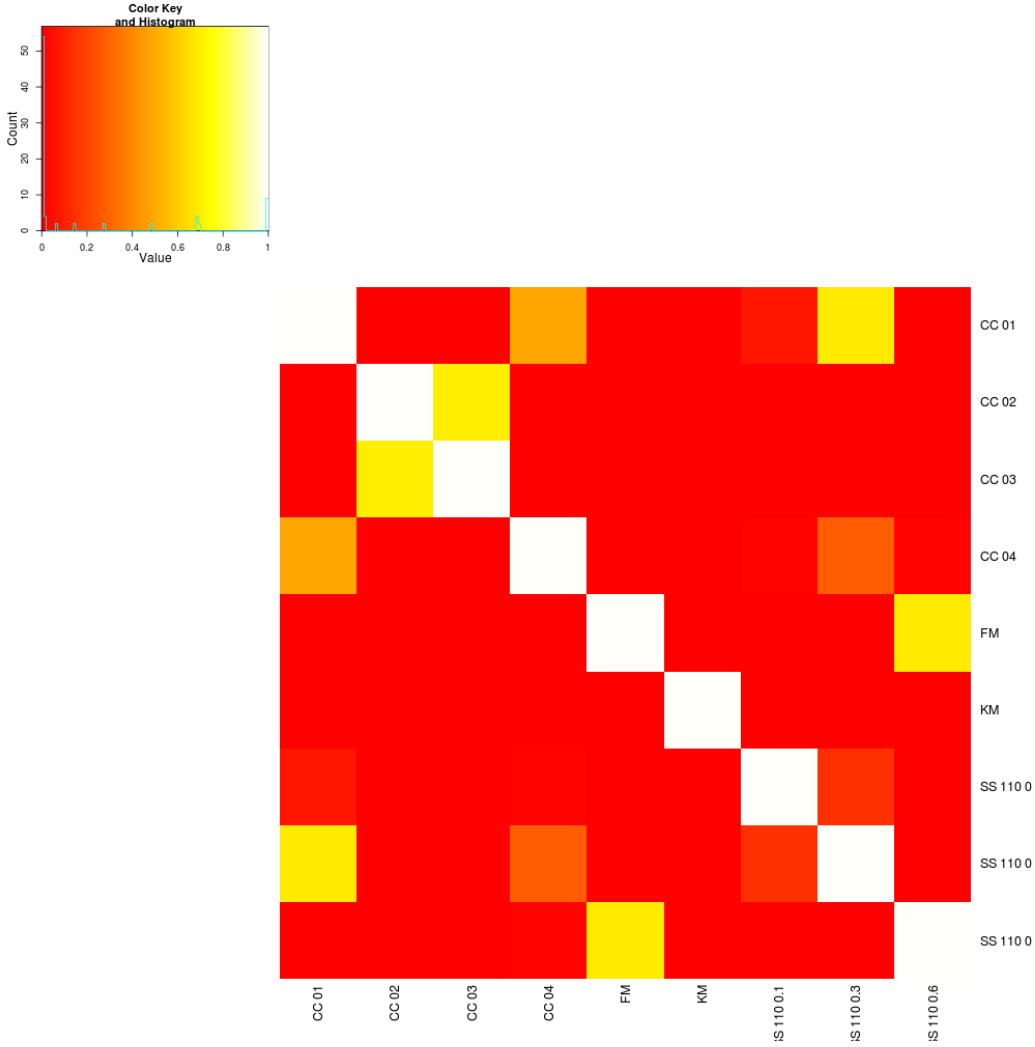


Figure 7: Comparison of p-values for the classification results on AML data, using SVM classifier

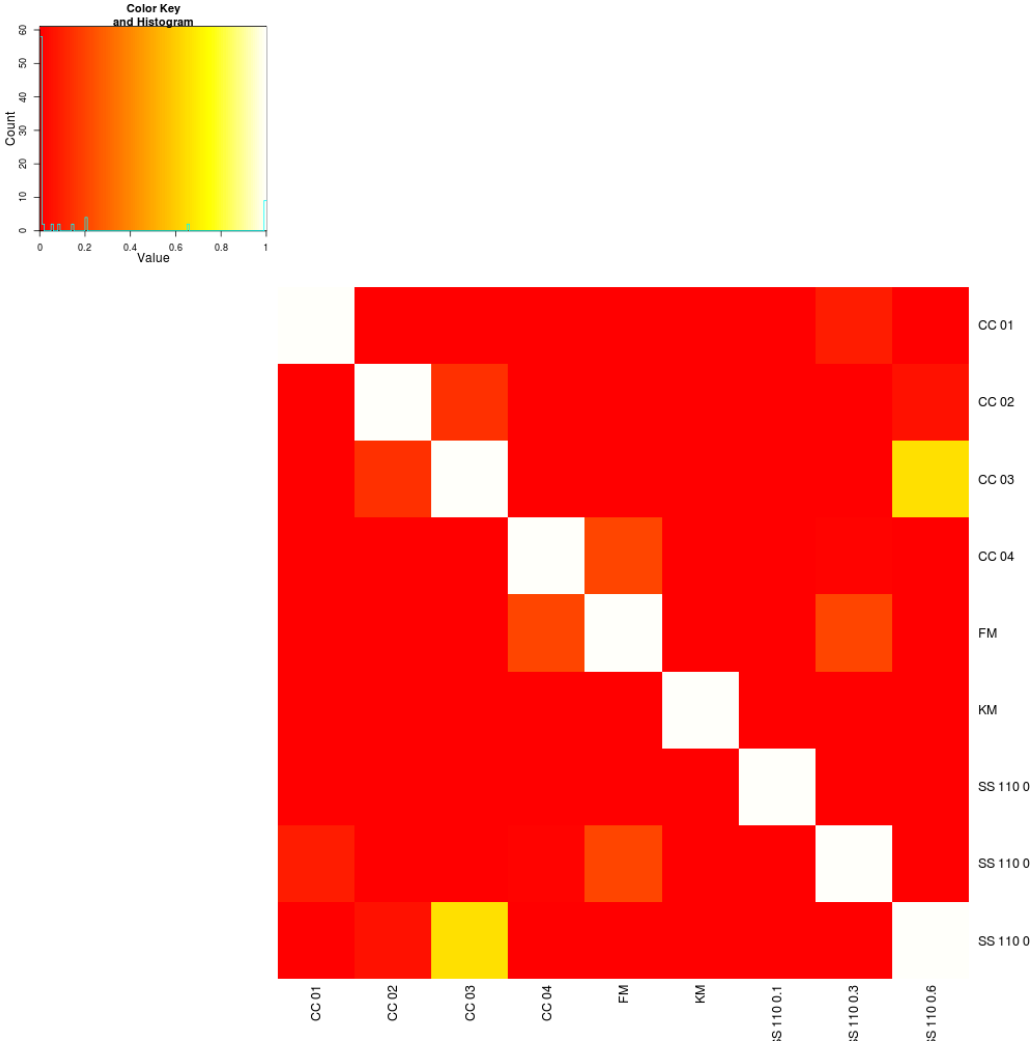


Figure 8: Comparison of p-values for the classification results on AML data, using MLP classifier

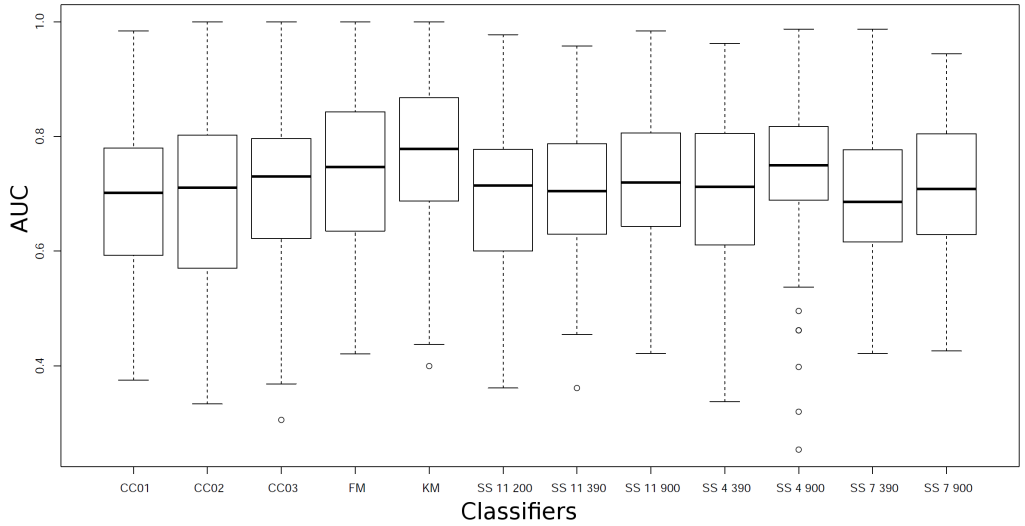


Figure 9: Boxplots of the AUC for MLP classifiers, on Lymphoma data set. Classifiers were trained and tested for 400 bootstrap samples.

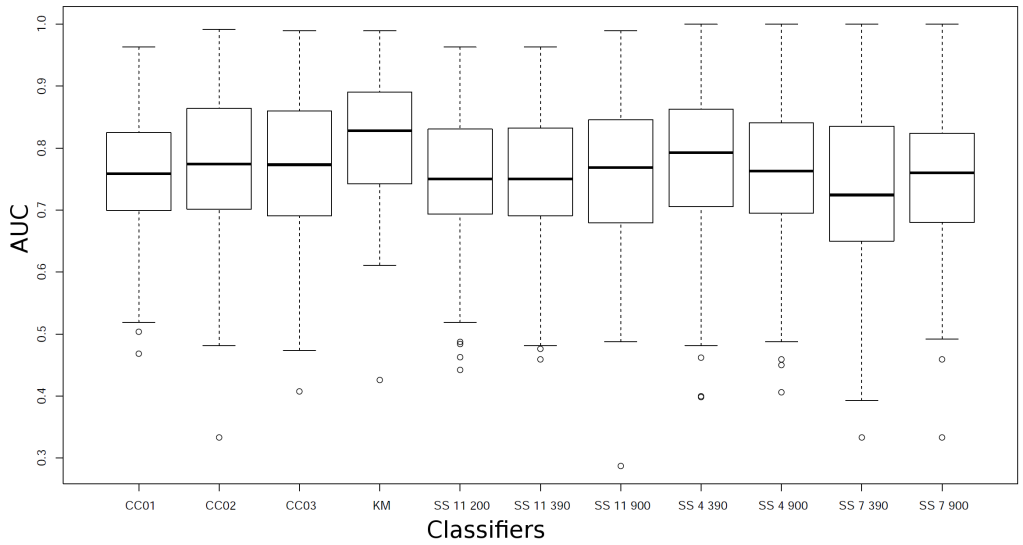


Figure 10: Boxplots of the AUC for SVM results on Lymphoma data set. Classifiers were trained and tested for 400 bootstrap samples.

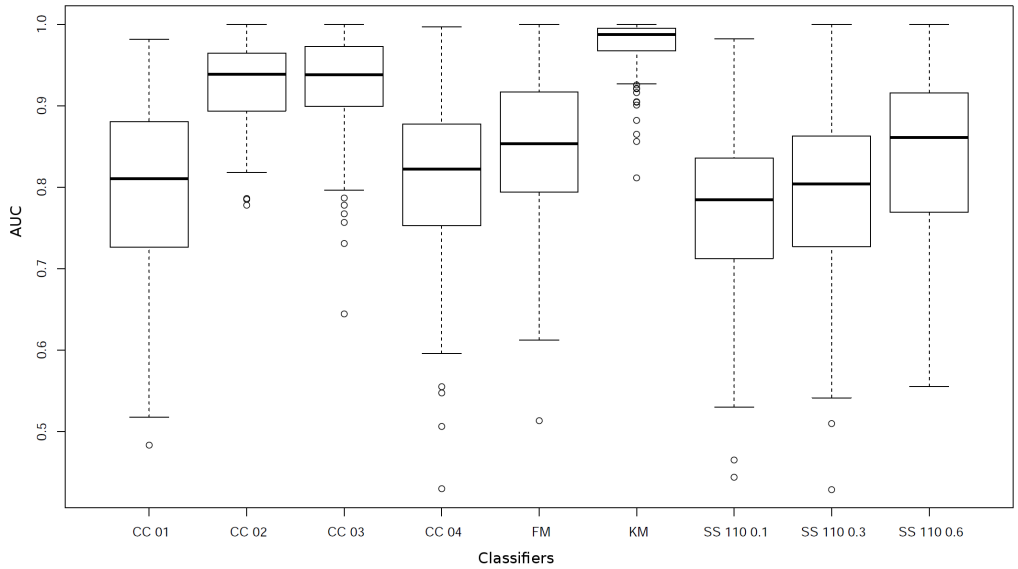


Figure 11: Boxplots of the AUC for SVM results on AML data set. Classifiers were trained and tested for 400 bootstrap samples.

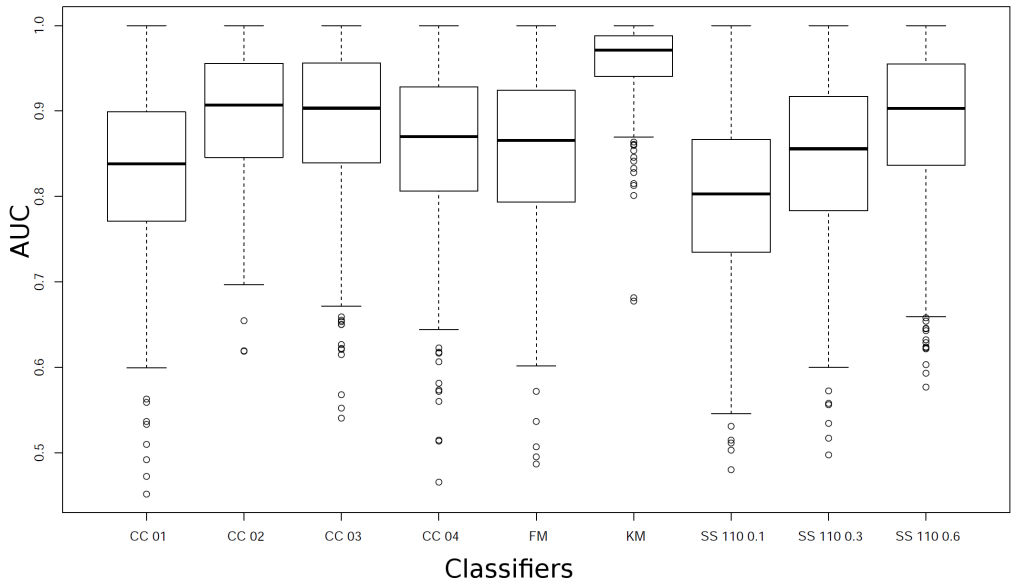


Figure 12: Boxplots of the AUC for MLP results on aML data set. Classifiers were trained and tested for 400 bootstrap samples.

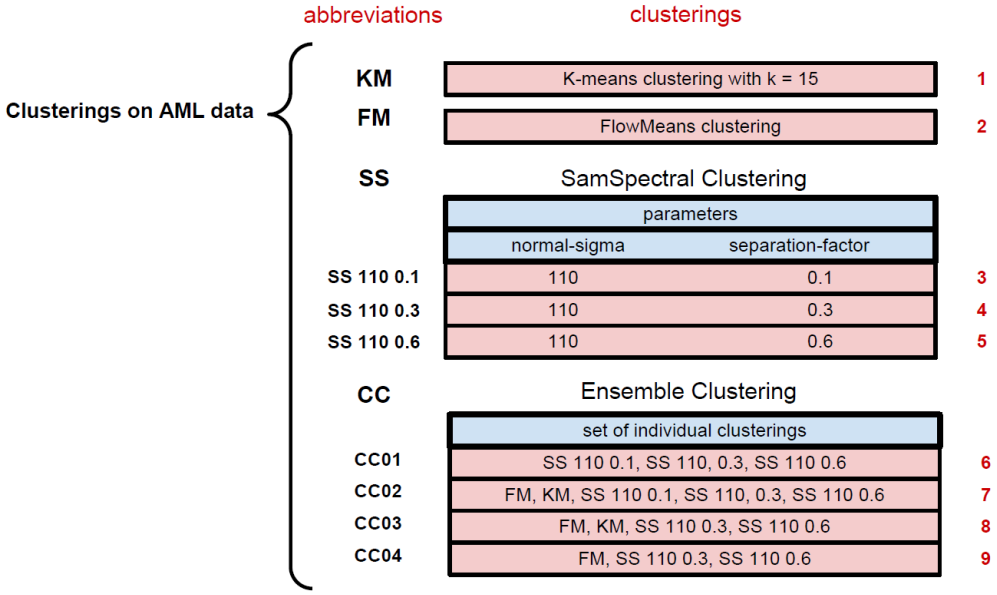


Figure 13: list of AML clusterings and the abbreviation used in the plots. Nine clusterings were applied on AML data, including k-means(KM), flowMeans(FM), three SamSpectrals(SS) and four combination of ensembe clusterings(CC). Rows 3-5 are having SamSpectral abbreviations and their parameters normal-sigma and separaiton factor. Rows 6-9 have the Ensemble Clusterins abbreviations and their set of individual clusterings.

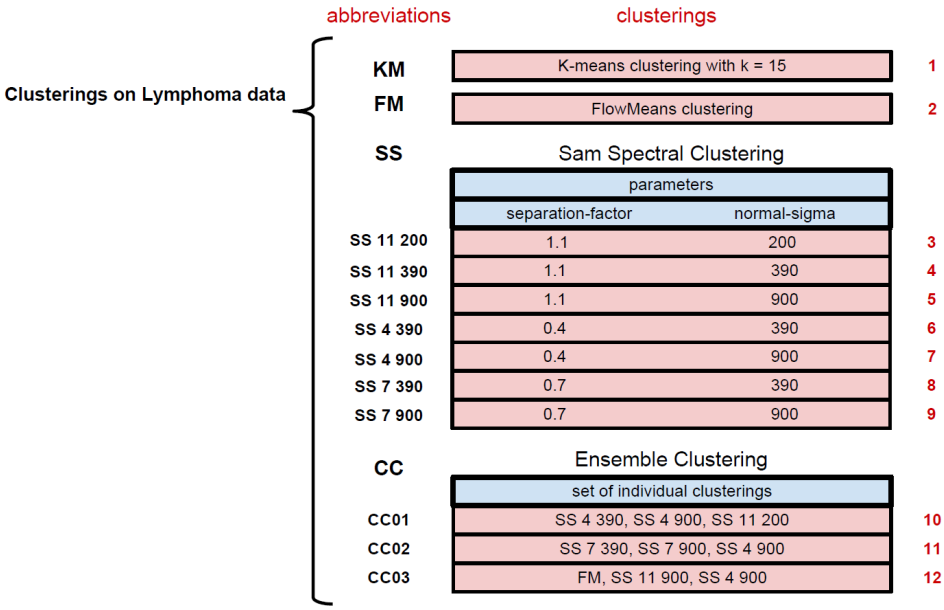


Figure 14: List of Lymphoma clusterings and the abbreviation used in the plots. Twelve clusterings were applied on Lymphoma data, including k-means(KM), flowMeans(FM), seven SamSpectrals(SS) and three combination of ensembe clusterings(CC). Rows 3-9 are having SamSpectral abbreviations and their parameters normal-sigma and separaiton factor. Rows 10-12 have the Ensemble Clusterins abbreviations and their set of individual clusterings.

dataset	marker
AML	FSC, SSC, lgG1-FITC, lgG1-PE, CD45-ECD, lgG1-PC5, lgG1-PC7
Lymphoma	FSC-A, FSC-W, SSC, FITC, PE, PerCP-Cy5-55, PE-Cy7, APC, APC-CY7, PacificBlue, AmCyan

Table 1: List of the markers for the data sets. *AML*, *Lymphoma*

clustering	CC01	CC02	CC03	CC04	FM	KM	SS 110 0.1	SS 110 0.3	SS 110 0.6	mean
SVM	0.795	0.928	0.925	0.806	0.849	0.972	0.767	0.789	0.843	0.853
MLP	0.830	0.894	0.885	0.859	0.851	0.958	0.798	0.842	0.882	0.867
SVM-MLP	-0.035	0.034	0.039	-0.053	-0.002	0.014	-0.032	-0.053	-0.039	-0.014

Table 2: Mean for the AUC results of the clusterings on AML data

clustering	CC01	CC02	CC03	FM	KM	SS 11 200	SS 11 390	SS 11 900	SS 4 390	SS 4 900	SS 7 390	SS 7 900	mean
SVM	0.758	0.770	0.767	0.774	0.816	0.756	0.755	0.755	0.777	0.747	0.728	0.754	0.762
MLP	0.702	0.710	0.708	0.730	0.769	0.705	0.709	0.718	0.710	0.718	0.683	0.704	0.714
SVM-MLP	0.056	0.06	0.059	0.044	0.047	0.051	0.046	0.037	0.067	0.029	0.045	0.050	0.048

Table 3: Mean for the AUC results of the clusterings on Lymphoma data

References

[1] N. Aghaeepour, R. Nikolic, H.H. Hoos, and R.R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79(1):6–13, 2011.

[2] Nima Aghaeepour. *flowMeans: Non-parametric Flow Cytometry Data Gating*, 2010. R package version 1.8.0.

[3] BD Bioscience. Flow cytometry . `http://www.bdbiosciences.com/support/training/itf_launch.jsp`.

[4] C.M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[5] G. Blanchard, O. Bousquet, and P. Massart. Statistical performance of support vector machines. *The Annals of Statistics*, 36(2):489–531, 2008.

[6] C. Boulis and M. Ostendorf. Combining multiple clustering systems. *Knowledge Discovery in Databases: PKDD 2004*, pages 63–74, 2004.

[7] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T.B. Kepler. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73(8):693–701, 2008.

[8] E. Dimitriadou, A. Weingessel, and K. Hornik. Voting-merging: An ensemble method for clustering. *Artificial Neural Networks—ICANN 2001*, pages 217–224, 2001.

[9] B. Ellis, P. Haaland, F. Hahne, N. Le Meur, and N. Gopalakrishnan. *flowCore: flowCore: Basic structures for flow cytometry data*. R package version 1.22.3.

[10] A.D.A.M. Medical Encyclopedia. Acute myeloid leukemia . `http://www.ncbi.nlm.nih.gov/pubmedhealth/PMH000156`.

[11] FlowCAP. Flowcap. `http://flowcap.flowsite.org`.

[12] A.L. Givan. *Flow cytometry: first principles*. John Wiley & Sons, 2001.

[13] Non hodgkin's Lymphoma Cyberfamily. Cd markers for b-cell lymphoma. `http://www.nhlcyberfamily.org/tests/cdmarkers.htm`.

[14] Kurt Hornik. clue: Cluster ensembles. r package version 0.3-44. `http://CRAN.R-project.org/package=clue`.

[15] Kurt Hornik. A clue for cluster ensembles. *Journal of Statistical Software*, 14:65–72, 2005.

[16] J. Huang and C.X. Ling. Using auc and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 17(3):299–310, 2005.

[17] L. Liu, L. Xiong, J.J. Lu, K.M. Gernert, and V. Hertzberg. Comparing and clustering flow cytometry data. In *Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE International Conference on*, pages 305–309. IEEE, 2008.

[18] G. Lizard. Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. *Cytometry Part A*, 71(9):646–647, 2007.

[19] K. Lo, R.R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73(4):321–332, 2008.

[20] K. Lo, F. Hahne, R.R. Brinkman, and R. Gottardo. flowclust: a bioconductor package for automated gating of flow cytometry data. *BMC bioinformatics*, 10(1):145, 2009.

[21] M. Markou and S. Singh. Novelty detection: a review—part 2:: neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003.

[22] LL Muldrow, RL Tyndall, and CB Fliermans. Application of flow cytometry to studies of pathogenic free-living amoebae. *Applied and environmental microbiology*, 44(6):1258–1269, 1982.

[23] R.F. Murphy. Automated identification of subpopulations in flow cytometric list mode data using cluster analysis. *Cytometry*, 6(4):302–309, 2005.

[24] F. Naeim, P.N. Rao, and W.W. Grody. *Hematopathology: Morphology, Immunophenotype, Cytogenetics, and Molecular Approaches*. Academic Press, 2008.

[25] C.E. Pedreira, E.S. Costa, M.E. Arroyo, J. Almeida, and A. Orfao. A multidimensional classification approach for the automated analysis of flow cytometry data. *Biomedical Engineering, IEEE Transactions on*, 55(3):1155–1162, 2008.

[26] S. Pyne, X. Hu, K. Wang, E. Rossin, T. I. Lin, L. M. Maier, C. Baecher-Allan, G. J. McLachlan, P. Tamayo, D. A. Hafler, P. L. De Jager, and J. P. Mesirov. Automated high-dimensional flow cytometric data analysis. *Proc. Natl. Acad. Sci. U.S.A.*, 106:8519–8524, May 2009.

[27] Ryan Brinkman Raphael Gottardo Tim Mosmann Yu Qian Jill Schoenfold Richard Scheuermann, Nima Aghaeepour. Flowcap: critical assessment of flow cytometry population identification methods. *The Journal of Immunology*, 186(65.2), 2011.

[28] RM Sakia. The box-cox transformation technique: a review. *The statistician*, pages 169–178, 1992.

[29] H.M. Shapiro and R.C. Leif. *Practical flow cytometry*, volume 4. Wiley Online Library, 2003.

[30] SY Sohn and HW Shin. Experimental study for the comparison of classifier combination methods. *Pattern Recognition*, 40(1):33–40, 2007.

[31] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[32] LW Terstappen, M. Safford, S. Könemann, MR Loken, K. Zurlutter, T. Büchner, W. Hiddemann, B. Wörmann, et al. Flow cytometric characterization of acute myeloid leukemia. part ii. phenotypic heterogeneity at diagnosis. *Leukemia: official journal of the Leukemia Society of America, Leukemia Research Fund, UK*, 6(1):70, 1992.

[33] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann. Review of classifier combination methods. *Machine Learning in Document Analysis and Recognition*, pages 361–386, 2008.

[34] L. Xu and S. Amari. Combining classifiers and learning mixture-of-experts. *Encyclopedia of artificial intelligence*, pages 318–326, 2009.

[35] H. Zare. Automatic analysis of flow cytometry data and its application to lymphoma diagnosis. 2011.

[36] H. Zare, P. Shooshtari, A. Gupta, and R.R. Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC bioinformatics*, 11(1):403, 2010.

[37] Habil Zare and Parisa Shooshtari. *SamSPECTRAL: Identifies cell population in flow cytometry data.*, 2009. R package version 1.7.4.3.