Visualization of Clustering Solutions for Large Multi-dimensional Sequential Datasets

by

Maninder Dornala

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master

of

Computing and Information Systems

Program

YOUNGSTOWN STATE UNIVERSITY

May, 2018

Visualization of Clustering Solutions for Large Multi-dimensional Sequential Datasets

Maninder Dornala

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Maninder Dornala, Student Date

Approvals:

Alina Lazar, Thesis Advisor Date

Bonita Sharif, Committee Member Date

John R. Sullins, Committee Member Date

Dr. Salvatore A. Sanders, Dean of Graduate Studies Date

ABSTRACT

The research presented is focused on the development of algorithms targeted towards analyzing data in the form of categorical time series or sequences. The wide availability of mobile devices and sensors connected to the Internet, makes it easier to collect datasets to model long-term user behavior. Nevertheless, performing fundamental analytical operations, such as clustering for grouping these data based on similarity patterns, has proved challenging due to the categorical nature of the data, the multiple variables to consider and their corruption by missing values. The classical metric type similarity distances have to be replaced with "edit" type distances, such as Optimal Matching. We developed this approach with the aim of studying the effect of similarity measure choice on clustering and dimensionality reduction methods applied to long-term life cycle trajectories. The discovered patterns can help providing better decision making and public policy design.

Acknowledgements

Firstly, I would like to express my true thankfulness to my advisor, Dr. Alina Lazar. I feel very honored to work with her, learn from her and I am grateful for her support, patience, and knowledge shared to complete my research. I appreciate her for giving me an opportunity to work with her and I could not have imagined having a better advisor for my Master Program at Youngstown State University.

I would like to thank my committee members Dr. Bonita Sharif, Dr. John Sullins for taking time out of their schedule to share their Knowledge and Insights.

A special thanks to my parents, family, and friends for the support they have given to me during my graduate career. I would also thank the Department of Computer Science and Information Systems and the College of Graduate Studies for the financial assistance during my graduate studies.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Overview

Constant advancement in fields of science and technology is resulting in larger collection of data than ever before. This led to the formation of immense datasets in science, government and many other industries, which should be analyzed, processed and sorted to extract useful information out of it. In recent years, interest in the study of large multi-dimensional datasets is growing significantly. Given a very large dataset consisting of high dimensional elements, how could one be able to analyze or extract patterns the data and come up with useful information and results. To perform this task, clustering of this huge data can be used. Clustering techniques are useful in grouping the closely related (similar) data points. In today's world, the large amount of information available has an enormous potential. Hence, it becomes more and more complicate to perform clustering on these data.

Clustering of these data is helpful because the output data objects in each cluster exhibit high degree of similarity between them and the similarity between clusters is reduced. For example, if we search for some query on the internet, there are hundreds of pages displayed. If we can apply clustering for all these pages and divide them into different clusters based on their similarities, then the end user will be displayed with only a few pages which are similar with the query posted. Clustering of data is helpful for many fields including classification of web documents, news articles, financial market, social sciences, health sciences etc.

The data which is used for clustering should have some set of rules otherwise we may have to face issues. There are some challenges to perform clustering and one of such is a missing data problem. The Swiss Household Panel conducted a Biographical Survey on individual and collected a data frame. This data frame is formatted with 2000 rows, 16 state variables, 1 id variable and 7 covariates and 2 weights variables. In 2007, Muller et al constructed a data named *biofam* from this data, which contained sequences of family states in columns 10 to 15. Distribution and sequential analysis was carried out on all these sequences.

Panel Study of Income Dynamics (PSID) collected data from individuals from 1968 through 2015, which contained over 17,000 records. In this survey, age is one of the key variables concentrated throughout. Variable age is found to be missing and has some noise throughout. Because of this missing data, data were preprocessed, and screening was done to make the data into the required format. After this process, the experimental data was brought down to 1034 individual records (considered as sequences) having individuals between ages 20 and 60. Experiments are conducted between these age groups, so the individuals who are older than 20 years during 1968 (when the survey was started) have missed the latter part of the survey which is called as *alignment missing*. From 1997, instead of conducting the survey every year, it was conducted in alternate years. Because of this missing survey every year, I have observed some Short survey gaps in the latter part of the sequences.

The distances between all these sequences are calculated and a data frame is formed known as the distance matrix. Hierarchical clustering techniques are used to cluster the sequences in distance matrix. To measure the dissimilarities between the sequences an *edit* type distance known as Optimal Matching (OM) is used. Optimal Matching (OM) is one of the methods in dissimilarity measures which shows the distances between objects or sequences as the minimum work which is calculated in the form of edit operations, required for changing two sequences to make them identical.

The research is done on high-dimensional datasets. But, practically it is not possible to plot the results in more than two or three dimensions. t-Distributed Stochastic Neighbor Embedding (t-SNE) is an algorithm which is used for dimensionality reduction, which is non-linear and used for experimenting on high-dimensional data and reducing it to low dimensions, which are easily understood by human observation. In the experiments, I have used t-SNE dimensionality reduction algorithm to handle the large multi-dimensional datasets and created the required results from it.

The Research Questions I would like to answer is:

Research Questions: Can we identify and extract representative sequences from categorical sequence datasets using clustering and dimensionality reduction techniques? What are the best similarity distances and dimensionality reduction methods to visualize the data in a meaningful way?

To answer these questions, I have taken four different datasets and conducted experiments. The datasets are as from different surveys as follows: a study conducted by McVicar and Anyadike-Danes on transition from school to work (*mvad*), family life states from the Swiss Household Panel Biographical Survey (*biofam*), and a survey conducted by Panel Study of Income Dynamics (PSID) ($totalFUSmall$) datasets.

First I have loaded the datasets into RStudio along with **TraMineR** package and observed the Individual sequences information. After this, I have used Partitioning Algorithmic technique (PAM) for clustering the datasets using the distance measures of Optimal Matching (OM), Localized Optimal Matching (OMloc), Number of Matching Subsequences (NMS) and Time Warp Edit Distance (TWED). Observing on the results of PAM, I have calculated peak values among ASWw, HG, PBC, HC for all distance measures and took the corresponding number of clusters as the Optimum number of clusters for that dataset. Later, using the optimum number of clusters, I have used Partitioning Algorithmic technique (PAM) for clustering the datasets in the second iteration using the same distance measures. From the output of second iteration of PAM, I have observed the individual clusters in all datasets. At last, I have plotted the results from the second iteration of PAM, using Dimensionality reduction techniques of Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA), Rt-SNE (t-Distributed Stochastic Neighbor Embedding).

1.2 Organization

This thesis is organized as follows. The next chapter explains the structure and description of datasets that were used previously by researchers and all the datasets used in my thesis. Chapter 3 presents different similarity measures used in clustering techniques, which is followed by the description of clustering algorithmic techniques in Chapter 4. Chapter 5 briefs about the dimensionality reduction techniques helpful for carrying cluster analysis. Chapter 6 describes the experiments carried out and presents the results. Chapter 7 concludes the thesis.

2 Description of Datasets

2.1 Family Life States from the Swiss Household Panel Biographical Survey

The Swiss Household Panel (SHP) is a study having a importance in the Swiss social survey landscape. Since 1999 SHP started collection of data from an individual households, that is useful for mid-term to long-term longitudinal study of various types of topics. In 1999, SHP conducted a study with around 5,074 individual households. During this study, most of the questions that were asked were manual. After that, in 2004 58 percent of the initial study sample were interviewed for the sixth time. [13]

The SHP survey is conducted annually from September to February by the institute M.I.S. Trend in Lausanne and Bern. The languages of interviews are (Swiss) German, French, and Italian. Computer-assisted telephone interviewing (CATI) is used as the primary mode of interview. The reason CATI was used as the method of survey was due to cost and quality considerations and national restrictions in Switzerland (Scherpenzeel 2000). The SHP team maintained close contacts between M.I.S during the interviews such that they can monitor the survey progress. There is no direct incentive for the interviewers till now and they were paid an hourly basis salary then.

The household questionnaire contains questions about the composition of the household, the standard of living, financial status, accommodation and information about the family. The average length of the interview for the household questionnaire is about 12 minutes. The individual questionnaire is carried out by every member in the house aged 14 or older. The individual questionnaire contains questions about the household and the family, health and quality of life, social origin (asked at first interview only), employment, education, income, integration, participation, and networks, leisure and media, politics and values, and psychological scales. [13]

Format and Details: *Muller et al* (2007) constructed the *biofam* dataset based on the biological survey conducted by SHP. This data frame consists of 2000 rows, 1 id, 16 state variables, 7 covariates and 2 weights. In the columns from 10 to 25 sequences of family states of the age between 15 and 30 are formatted along with a series of covariates[17].

Variables in *biofam* dataset are listed below

Variable	**Label**
idhous	household number
sex	sex of the individual
birthy	year of birth of the individual
nat.1.02	first nationality of the individual
plingu02	language of the interview
p02r01	Religion of the individual
p02r04	Individual's Frequency of Participation in religious activities
cspfaj	Swiss socio-professional category: Father's job
cspmoj	Swiss socio-professional category: Mother's job
a15	status of the formation at age 15
..	status of the formation
a30	status of the formation at age 30

Table 1: Variables in *biofam* dataset

The combination of five basic states like Living with parents (Parent), Left home (Left), Married (Marr), Having Children (Child), Divorced are defined with states numbered from to 7 as follows:

State	**Label**
0	Parent
1	Left
2	Married
3	Left+Marr
4	Child
5	Left+Child
6	Left+Marr+Child
7	Divorced

Table 2: States in *biofam* dataset

Distribution and Sequence plots: **seqdplot()** function is used to show the graphical representation of different states in a dataset at each point of time. First, we define a vector named *biofam.lables*. Using this vector the seven states shown above are plotted. *xtstep* sets up the x-axis distance for the plot.

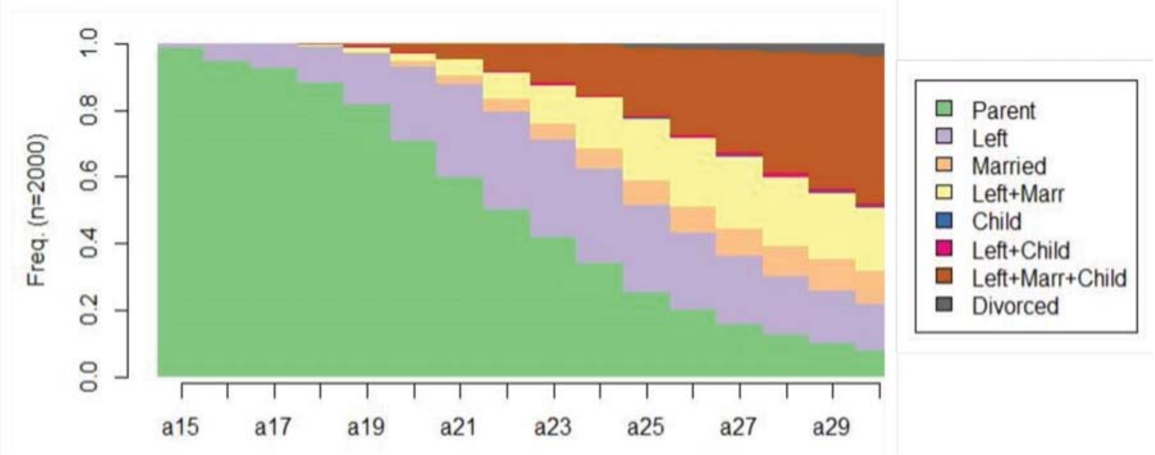

Figure 1: Sequential distribution plot in *biofam* dataset

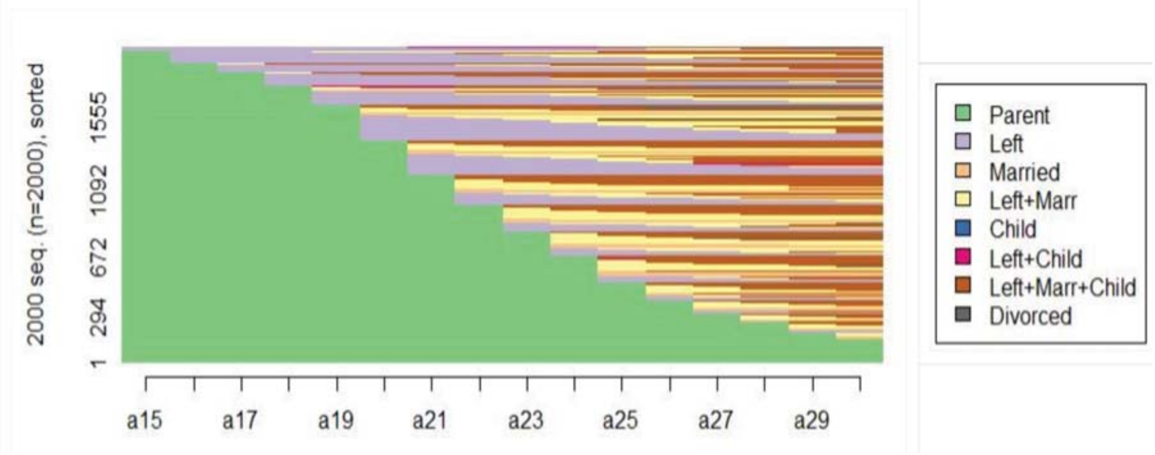

Figure 2: Individual Sequence distribution plot in *biofam* dataset

2.2 Transition from School to Work

This dataset is collected from the study conducted by *McVicar and Anyadike-Danes* on **transition from school to work**. In 1999, there was a survey conducted with interviews face-to-face among a sample of 980 individuals who have completed their compulsory education. For every individual, their monthly activity (e.g. at school, at further education college (FE), in training, employment status) was collected for next 2 years. After the sweep in 1999 June, this monthly activity and background data were updated by adding higher education (HE) and the age groups from 16-22 are considered. This changed the sample data size to 712. Appropriate weights adjust for response bias. The final data format of this data-frame contains 712 rows, 72 state variables(individual time series sequences of monthly labor activities of 72 months), 1 id variable and 13 covariates. [11]

The data set contains the following sample weights (weight), ids and the following binary covariates:

Variable	**Label**
id	unique individual identifier
weight	sample weights
male	binary dummy for gender, 1=male
catholic	binary dummy for community, 1=Catholic
Belfast	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
N.Eastern	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
Southern	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
S.Eastern	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
Western	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
Grammar	binary dummy indicating type of secondary education, 1=grammar school
funemp	binary dummy indicating father employment status at time of survey, 1=father unemployed
gcse5eq	binary dummy indicating qualifications gained by the end of compulsory education, 1=5+ GCSEs at grades A-C, or equivalent
fmpr	binary dummy indicating SOC code of father?s current or most recent job,1=SOC1 (professional, managerial or related)
livboth	binary dummy indicating living arrangements at time of first sweep of survey (June 1995), 1=living with both parents
jul93	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE
..	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE
jun99	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE

Table 3: Variables in *mvad* dataset

State	**Label**
EM	employment
FE	further education
HE	higher education
JL	joblessness
SC	school
TR	training

Table 4: States in *mvad* dataset

Distribution and Sequence plots:

seqdplot() function is used to show the graphical representation of different states in a dataset at each point of time. First, we define a vector named *mvad.lables*. Using this vector the six states shown above are plotted. *xtstep* sets up the x-axis distance for the plot.

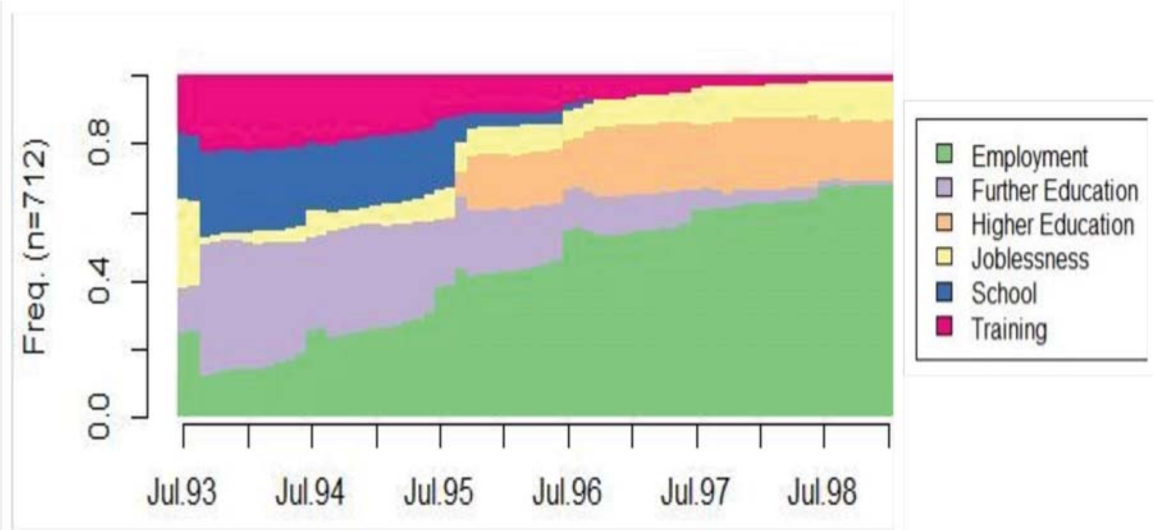

Figure 3: Sequential distribution plot by age in *mvad* dataset

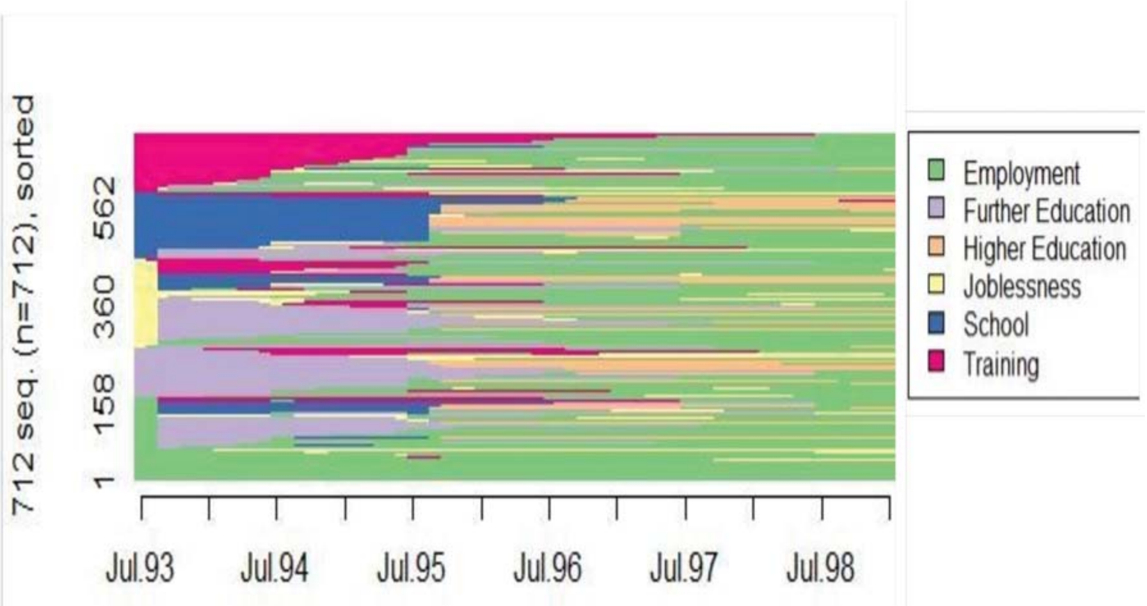

Figure 4: Individual Sequence distribution plot in *mvad* dataset

2.3 Number of Family Members from Panel Study of Income Dynamics

Panel Study of Income Dynamics (PSID) is the world's largest household panel survey, conducted a survey from 1968 through 2015 and collected information about employment, number of children, total family members etc. Initially, the dataset was about 18,000 records of individual people. Among all the variables age is concentrated throughout the survey. From 1997, the survey was conducted in alternate years which resulted in missing data of the age variable. because of the human error and the month in which survey was conducted age is vulnerable to noise.

Many pre-processing steps were done such that the data was brought into the required structure suitable for experiments. After this step, the data was brought to 1034 individuals between the age of 20 and 60. As the sequences were aligned according to age, concentration was made on the individuals whose age is from 20 to 60 years. During 1968, individuals who are above 20 years of age have missed the latter part of the survey which is called as *alignment missing*. From 1997, instead of conducting the survey every year, it was conducted in alternate years and it is known as *short survey gaps*[8].

So, the actual dataset *totalFUSmall* was divided into two sets with missing and no missing termed as *totalFUSmall with no missing values* and *totalFUSmall with missing values* respectively.

The variables and States in this dataset are as follows:

Variable	**Label**
Interview number 68	interview number of individual
Sequence number 68	sequence number of individual
20	age of individuals
21	age of individuals
..	..
59	age of individuals
60	age of individuals

Table 5: Variables in *totalFUSmall* dataset

State	**Label**
1	total number of family members is 1
2	total number of family members is 2
3	total number of family members is 3
4	total number of family members is 4
5	total number of family members is 6
>=6	total number of family members is 6 or more
missing	missing data

Table 6: States in *totalFUSmall* dataset

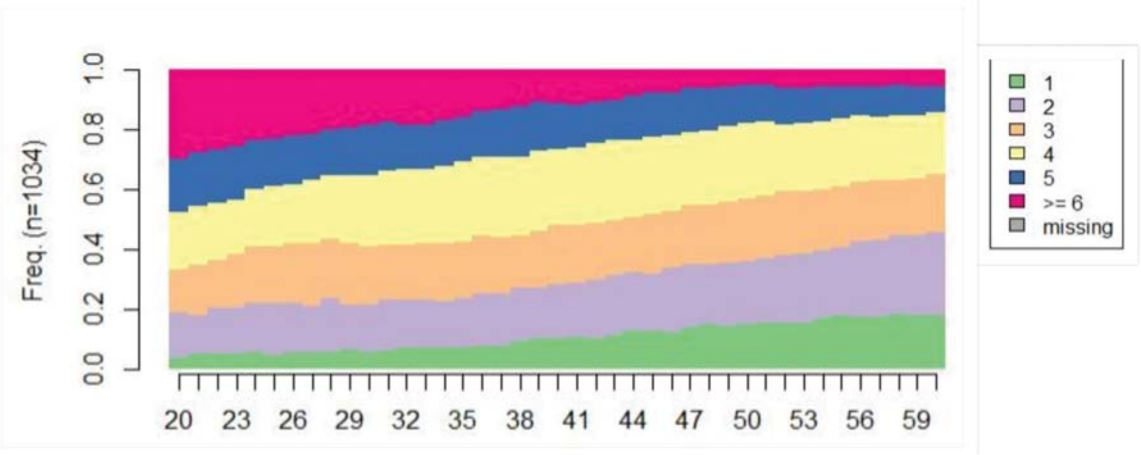

Figure 5: State distribution plot with no missing values in *totalFUSmall* dataset

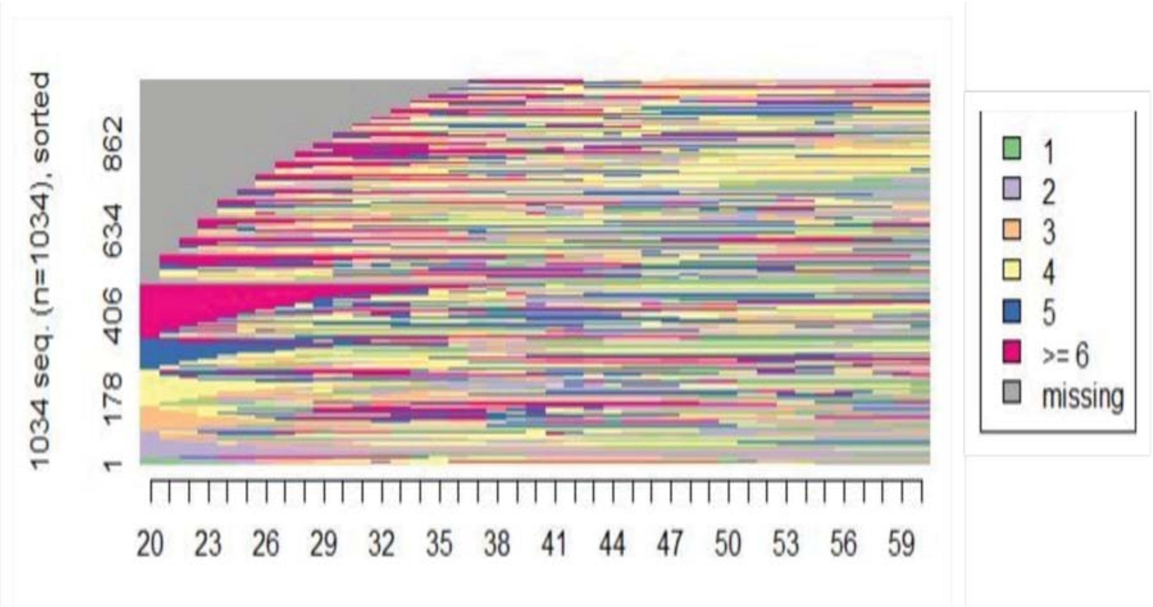

Figure 6: Individual Sequence distribution plot with no missing values in *totalFUSmall* dataset

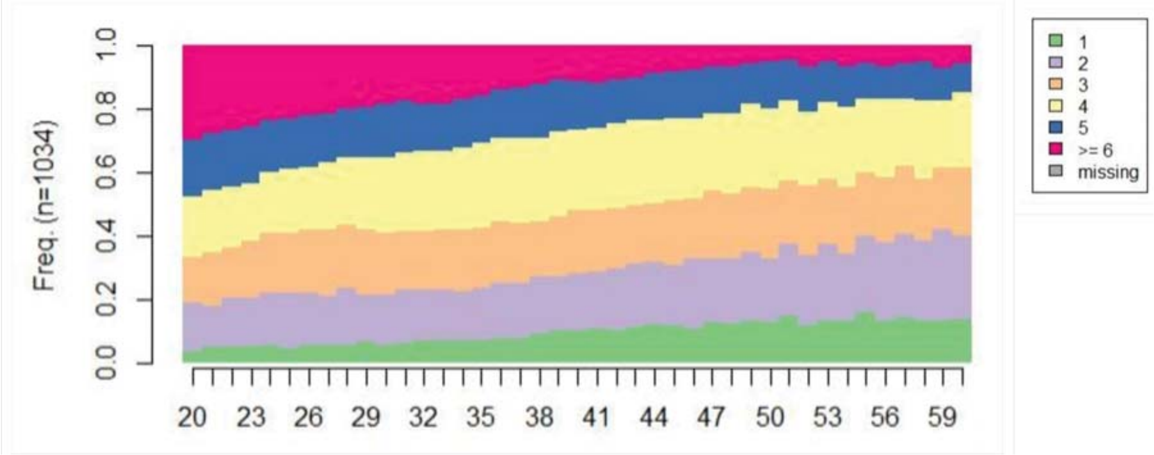

Figure 7: State distribution plot with missing values in *totalFUSmall* dataset

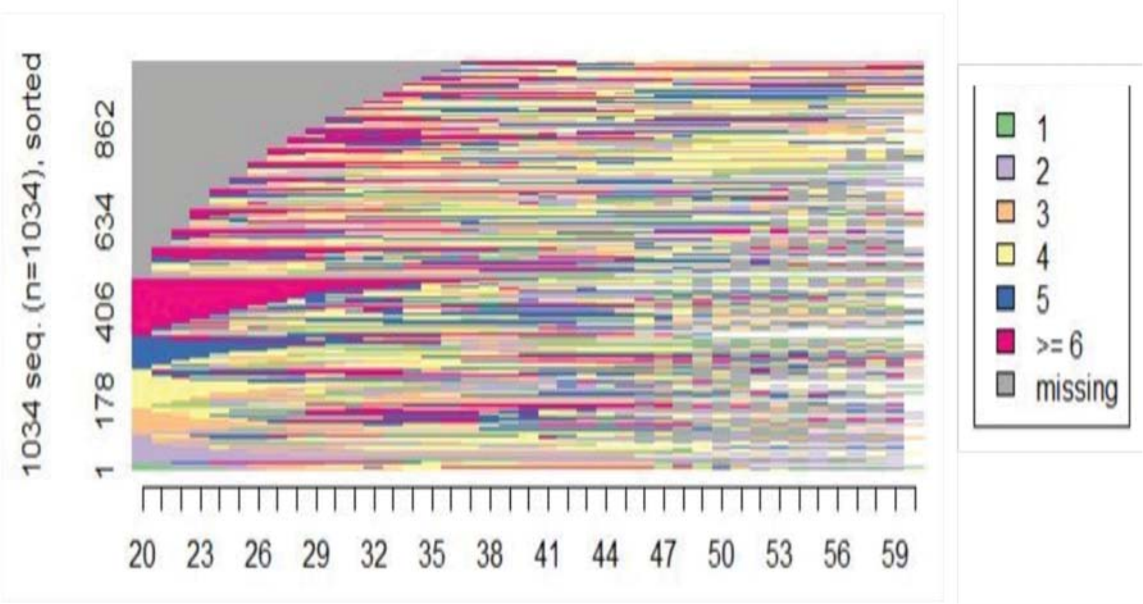

Figure 8: Individual Sequence distribution plot with missing values in *totalFUSmall* dataset

3 Similarity Measures

In the similarity measures section, I have divided the section into two subsections, in which I will discuss different edit methods namely *Optimal Matching (OM)*, *Localized Optimal Matching (OMloc)*.

3.1 Optimal Matching (OM) and Localized Optimal Matching (OMloc)

Optimal Matching (OM): Optimal Matching (OM) is one of the families of measures to calculate the similarity between sequences derived from the distance. It was initially proposed in Information Theory and Computer Science by Vladimir Levenshtein (1965) and known as sequential analysis in social sciences field. Later in 1983 OM was developed by Kruskal. Technically, OM shows the distances between objects or sequences as the minimum work which is calculated in the form of edit operations, required for changing two sequences to make them identical. [1]

Consider a set which performs three operations to transform the sequences: O $= i$, d, s where i denotes inserting one state into a sequence, d denotes deleting a state from a sequence and s denotes substituting one state by replacing with another state in a sequence. For each operation, there is a specific cost assigned and the sum of all these specific costs are calculated together to get the cost required for a single operation. Hence, the distance between two sequences is defined as the minimum amount of cost that is required for transforming one sequence into another sequence.

Therefore, the output resulting matrix is a symmetric matrix containing pairwise distances which is useful for future statistical analysis[8, 9].

Localized Optimal Matching (OMloc): Hollister(2009) proposed an extension for the OM measure, having the target to make the insert-delete(indel) costs to rely on the two states that are adjacent. There is a concept that, if a state is inserted or deleted, then we will only observe the change only in the length of spell in that particular state. It means that the insertion or deletion of any state does not affect the sequencing. But, there are more consequences on the indel cost of a state different to its neighbors' state. Therefore, a higher cost must be charged for indel cost. Generally, the cost of inserting z between p and q is defined as $c_i(z|p,q)$[15, 18]

$$c_i(z|p,q) = e\gamma_{max} + g\frac{\gamma(p,z)+\gamma(q,z)}{2} \quad [15]$$

here, $\gamma()$ indicates the cost of substitution, $\gamma()_max$ is the maximum cost of substitution, g and e are user-defined costs. The expression $e\gamma_{max}$ is the indel cost that is fixed and e is considered as the spell expansion cost. In 2009, Hollister conducted experiments and got the good results with some small shift penalization, g and e near to $1-2e$. The method stops OM from the use of two indels instead of a cost of substitution and if e and g satisfy the limits $1-2e \leq g$. In all experiments conducted by Hollister, even though surrounding states are changed, indel costs remain same. By changing the indel cost after each operation results in the rise of computational issues. This is because the total cost will change along with the order in which successive operations are applied. By construction, if the differences in spell length are compared then localized OM is considered to be less sensitive than classical OM[15][16].

3.2 Number of Matching Subsequences (NMS)

In this section, I will discuss a Metrics depending on the number of count of similar elements called *from Number of Optimal Clusters).*

In 2003, Elzinga proposed a measure (dissimilarity) which is dependant on the count of matching sub-sequences, NMS. The concept of this measure is that, how frequently a particular order of tokens from one sequence is observed in another sequence, the closer two sequences are to each other.

Studer and Elzinga (2015) introduced a model of NMS which is termed as SVR-spell (sub-sequence vector representation-based metric). The distance is dependent on the subsequences that are matching between DSS sequences and these matching sequences are given weight according to the duration of spells involved and their length. The behavior of this measure is controlled by two parameters. First, $a \geq 0$, which is expressed as an exponent for the length of subsequence weights. Second, $b \geq 0$ which is the exponent for the duration of spells. Apart from these weights, SVR (subsequence vector representation) also considers the proximities of states.

SVRspell and NMS are the Euclidean distances. They are sensitive to differences in duration and also differences in sequences. Considering the increase of duration, the original version will increase the count of embeddings of concerned subsequences. This is done by the second form by considering the spell durations. Contrast, calculating NMS between DSS sequences is equivalent to SVRspell with $b = 0$[16].

3.3 Time Warp Edit Distance (TWED)

In this section, I will discuss an edit method called *Time warp edit distance (TWED).*

In 2009, P.-F. Marteau developed Time Warp Edit Distance (TWED) measure. TWED is a distance measure between discrete time series. Unlike other measures (DTW , LCS) TWED is a metric measure. In other distance measures, it is assumed that all the data points present in the measured time series must be sampled at the same frequency and present at equidistant sampling times. In the context of clustering, the main problem with the processing of time series is the determination of the similarity of time series to one another.

In time series, we look at a sequence of edit operations that allows the transformation of 2 time series parallel, such that they are superimposed with minimum cost. If we represent this in a 2-dimensional graphical representation of time series, then the horizontal axis will represent the time stamp scale and vertical axis will represent spatial projection of 1-dimensional spatial co-ordinates[16].

TWED gives the following conclusions:

The cost or effort of editing any deletion operation is directly proportional to the length penalty added vector. Because of sampling rate variations, we can face a situation in time series data, where an event can be registered more times or few times depending on the number of occurrences. This will conclude that the deletion cost of an event is proportional to the distance in the previous sample.

4 Clustering Algorithms

Clustering a data means, all the data points in a data space are grouped together considering information found in the dataset which explains the relationship between its data points. Target is to create a group with similar objects by forming a relation between objects from same or other groups. The uniqueness of the clustering increases with greater the homogeneity within in the created group. In most of the applications, view of the cluster in datasets is not defined properly.

Every clustering technique follows a different set of rules in order to define the similarity among all the data points in data space. Practically, in data science, there are about more than hundred clustering algorithms. Among all those few models are listed below [5, 7, 14].

- **Connectivity models:** Connectivity models follow the rule that all data points in a data space which are closer to each other have more similarity than those points which are far away from each other. There are two approaches for this model. The first approach is to organize all points in a data space into separate groups called clusters and then grouping all these data points as the distance between them decreases. The second approach is that, all the data points are organized in such a way, to form a singleton cluster and then the singleton cluster is partitioned as the distance between the points increases.Connectivity models are easy to implement but they cannot handle big datasets. Hierarchical clustering and its variants are the examples of this model.

- **Centroid models:** Centroid models are known as iterative clustering algorithms because the view of similarity between objects is obtained by the closeness of each object in data space to the centroid of clusters. K-Means clustering algorithm falls into this category. In these models, we must know about the dataset before hand, because the number of clusters required has to be mentioned at first.

- **Distribution models:** Distribution clustering models follow the view that, what is the chance of having all data points in data pace in a cluster belong to the same distribution. These models have a disadvantage of over-fitting. Expectation-maximization algorithm is an example of this category which uses multivariate normal distributions.

4.1 Hierarchical Clustering

Hierarchical clustering approach follows the clustering procedure that produces a clustering result, which starts with every single point in data space as a singleton cluster and performs the grouping these points continuously on 2 nearest clusters. This process will be continued until a single cluster remains in the complete dataset. These techniques are divided into two groups. First, in terms of graph-based clustering for one group having a natural interpretation. Second, having an interpretation in terms of a prototype-based approach. [5] There are two methods for performing hierarchical clustering:

Agglomerative method: This approach considers all points as separate clusters and at each step, the nearest pair of groups (clusters) are merged together. For this process, the notion of cluster proximity is to be defined first.

Divisive: This process starts with a single cluster (having all clusters in it) and at every step, each cluster is split into two and the process continues until all the clusters having single points are remained. In this method, we should decide which cluster is to be split and how the splitting is carried out.

From the two clustering techniques stated above, Agglomerative hierarchical clustering technique is often common and is discussed here. A hierarchical clustering is generally displayed in the form of graphs using a tree-like structure called a dendrogram. Every dendrogram displays all the cluster-sub-cluster relationships and the order in which the clusters were merged (in case of agglomerative) or split (in case of divisive). [2, 12] . Following Algorithm explains more in detail about the Hierarchical clustering.

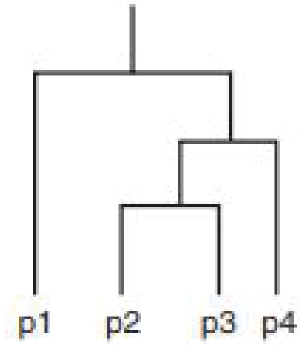

Figure 9: Dendrogram showing four points using hierarchical clustering

Algorithmic steps for Agglomerative Hierarchical clustering:

Let us consider a set X having the data points as X = x_1, x_2, x_3, ..., x_n.

1. Start with the initial (disjoint) clustering level L(0) = 0 and a sequence number m = 0.

2. Search for the distance pair of clusters that is minimum in the present clustering, assume pair (r), (s) are the required clusters. According to distance d[(r),(s)] = min d[(i),(j)] where the min is total pairs of clusters in the present clustering.

3. Increase the number of sequence: m = m +1. Group the two clusters (r) and (s) to form a new cluster m. Change this clustering level to L(m) = d[(r),(s)].

4. The distance matrix D is updated by deleting the rows and columns which corresponds to clusters (r), (s) and the new rows and columns which corresponds to the newly formed cluster should be added. The expression d[(k), (r,s)] = min (d[(k),(r)], d[(k),(s)]) defines the distance between the new cluster (r,s) and old cluster(k).

5. If all the data points are grouped in one cluster then algorithm terminates, else repeat al the steps from step 2.

Some of the other factors that must be considered while performing hierarchical clustering are *Defining Proximity between clusters* and *Time and Space Complexity.*

Proximity between clusters: The main goal of this algorithm is to compute the proximity between two clusters, and that is one which differentiates various agglomerative hierarchical techniques. Proximity between clusters is defined with a special type of cluster. A graph-based view of the groups (clusters) is the basic structure of agglomerative hierarchical cluster methods like MIN, MAX and Group Average. The proximity between any two close points in a data space that are in different nodes or clusters is defined by MIN. Similarly, the proximity between any two farthest points in a data space that are in different nodes or clusters is defined by MAX[5].

If suppose we take a view for which every cluster is shown by a centroid, then there are many definitions for defining the proximity between clusters. If the centroid is used to define proximity between clusters then the cluster proximity is defined as the proximity between centroids of a cluster. Another clustering method called, Ward's method, proposes that a centroid is used to represent a cluster, but the cluster proximity is measured by the closeness between two clusters which is measured in terms of increase in the SSE.

Time and Space Complexity : We discussed that, agglomerative hierarchical clustering algorithm makes use of a proximity matrix. So, by assuming that the matrix is symmetric, we require to store m_2 proximities, where the number of individual data points are represented by m. The total space required to store all the clusters is directly proportional to the number of clusters. Therefore, the space complexity of all the data points is given by the expression $\mathrm{O}(\mathrm{m}^2)$ [4].

4.2 K-means Clustering

In K-Means clustering choose K initial centroids, where K is the parameter specified by user which represents the number of clusters. Every point in the dataset is assigned to the centroid that is closest. A centroid that is assigned by every such collection of points is a cluster. Based on the points assigned to the cluster, the centroid of each cluster is then updated. These steps are continuously performed until there are no point changes in the clusters i.e. until the centroids remain the same. [3]

$$J(V) = \sum_{i=1}^{C} \sum_{j=1}^{C_i} (|x_i - v_j|)^2$$

where,

$|x_i - v_j|$ is the Euclidean distance between x_i and v_j.

C_i is the number of data points in i^{th} cluster.

C is the number of cluster centers.

Algorithmic steps for k-means clustering:

Consider the set of data points X $= x_1, x_2, x_3, ..., x_n$ and the set of centers V $= v_1, v_2, ..., v_c$.

1. Select a random number of clusters (say C).

2. Calculate the distance between the center of clusters and each data points.

3. Observe the cluster center whose distance is minimum of all the cluster centers and assign the data point to that cluster center.

4. Recalculate the center of the new cluster using:

$$V_i = (1/C_i)\sum_{j=1}^{C_i} x_i$$

where, C_i represents the number of data points in i^{th} cluster.

5. Recalculate the distance between newly obtained cluster centers and each data point.

6. If there is no data point that is reassigned then terminate, otherwise continue from step 3.

Time and Space Complexity: The required space for K-means clustering is not too high. This is because we need to store only the data points and centroids. The storage required is given as O((a+b)z), where the number of points is represented by *a*, the number of attributes is denoted by *z*. Time needed for k-means clustering is given by O(I x k x m x n), where number of iterations required for k-means is denoted by I, which is often small and maximum changes generally occur in the initial few iterations. Therefore, K-means clustering is considered to be linear the number of points (m) and is simple and efficient if the number of clusters (k) is practically less than m.

5 Dimensionality Reduction Methods

Dimensionality Reduction Techniques are used to map a high dimensional data space to a low dimensional data space. It is expressed as follows: Suppose X ϵ $R^{(\mathrm{pxq})}$, is considered to be a set of p data points in a q-dimensional data space, δ_{p}, δ_{t} are two metric distance (or dissimilarity) function, given δ_{p} : $\mathrm{R^p}$ x $\mathrm{R^p}$ $\rightarrow$R and δ_{t} : $\mathrm{R^t}$ x $\mathrm{R^t}$ $\rightarrow$R, where $\mathrm{R^p}$ is data space and $\mathrm{R^t}$ target space respectively, with p, t ϵ $\mathrm{N^*}$, and t « p, be given. A mapping function ϕ that maps the p-dimensional data points (x_i ϵ X) to t-dimensional target points (y_i ϵ Y), i.e.,

ϕ: $\mathrm{R^p}$ $\rightarrow \mathrm{R^t}$

$x_i \rightarrow y_i$, for 1$\leq$i$\leq$n,

5.1 Multi-dimensional Scaling

Multi-dimensional scaling (MDS) is a technique that is used when we are given a table of distances between several objects and to map them. This mapping may contain dimensions like 1,2,3, or more. The technique computed one of the solutions from metric or non-metric. The table formed by distances is known as **proximity matrix**. Proximity matrix can be taken either from experiments or as a correlation matrix. Multi-dimensional scaling is divided into two groups called *Metric, or Classical, Multi-dimensional Scaling (CMDS)* and called *Non-Metric Multi-dimensional Scaling (NMMDS)*. Metric MDS tries to reproduce the original distances. Non-Metric Multi-dimensional Scaling (NMMDS), is based on the concept it produces a map which tries to reproduce the ranks of the distances[6].

Goodness of Fit: Expressing a data how good it is to be represented by the model which is used in analysis. For MDS, modeling of the distances is the aim. Hence, the best option for goodness-of-fit is dependent on the difference of the actual distances and predicted values. It's termed as **stress** and can be computed as follows:

$$stress = \sqrt{\frac{\sum (d_{ij} - \hat{d}_{ij})^2}{\sum {d_{ij}}^2}}$$

Here the predicted distance depending on the MDS model is given by $\hat{d}_{ij}$.

Number of Dimensions: One of the important goals for an analyst using MDS model is to determine the number of dimensions. It is important to keep the number of dimensions low as possible. Generally, one selects 2 or 3 dimensions. The concept of MDS is to solve for values with several dimensions and get a small number of dimensions that give a lesser value of stress[6].

The size of the eigen values which are obtained at the time of the solution process is also considered by some of the researchers. As these eigen values are used to compute the number of factors during factor analysis, and also used to calculate the number of dimensions.

Proximity Measures: These are used to calculate the closeness of two objects. MDS defines three types of proximity values: dissimilarities, similarities, and correlations.

Dissimilarities: Dissimilarities represent the distance between two objects in a dataset.

Similarities show how closely two objects are arranged in a dataset. Similarities must obey the rule: $similarity_{pq} \leq similarity_{pp}$ and $similarity_{qq}$ for all p and q. Similarity matrices are symmetrical.

Similarities are converted to dissimilarities using the formula:[6]

$$d_{pq} = \sqrt{s_{pp} + s_{qq} - 2s_{pq}}$$

where d_{ij} represents a dissimilarity and s_{ij} represents a similarity.

5.2 Principal Component Analysis

Principal component analysis (PCA) is an example of a statistical procedure that converts a set of observations using orthogonal transformation. The set of variables which are possibly correlated are mapped to a set of values that are linear and uncorrelated variables known as principal components. The count of distinct principal components is given by min (n-1, p) where, n is the number of observations with p variables. The orthogonal transformation is defined as the principal component which first has the possible variance of highest and each component that succeeds the first has the highest variance having a restriction that it should be perpendicular to the preceding components. The set of all resulting vectors are a set of uncorrelated orthogons. PCA is very much flexible to the relative scaling of the original variables[16].

Let us assume that a data matrix is given with variables count of q and observations count of r, then all data points are centered by the means of each variable. It makes sure that most of the data is at the center of the means of each variable. The first principal components (Y_1) is given by variables of linear combination as X_1, X_2, ...,X_q

$$Y_1 = a_{11}\ X_1 + a_{12}\ X_2 + ... + a_{1\mathrm{p}}\ X_q$$

or, in matrix notation

$$Y_1 = a_1^T\ \mathrm{X}$$

The principal component that is calculated first will account in the variance in the data set which is maximum possible. By choosing large values for the weights a_{11}, a_{12}, ... $\mathrm{a}_{1\mathrm{p}}$ variance of Y_1 can be made as big as possible. To avoid this, there is a constraint on the weights such that their sum of squares is 1.

$$a_{11}^2 + a_{12}^2 + ...+ a_{1q}^2 = 1$$

The Second principal component is also calculated in the similar fashion, by having a condition that is perpendicular to the first component and it accounts for next maximum variance.

$$Y_2 = a_{21}\ X_1 + a_{22}\ X_2 + ... + a_{2\mathrm{q}}\ X_q$$

This process is continued until we calculate q principal components, which are same as the number original variables. At this step, both the sums of variances of all the variables and variances of all the principal components will be equal. So, together the transformations of the variables that are original to principal components is given by

$$\mathrm{Y} = \mathrm{AX}$$

5.3 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor(t-SNE) Embedding is a non-linear dimensionality reduction algorithm that is used for reducing high-dimensional data into two or three-dimensional data. t-SNE algorithms help in plotting graphs with fewer explanations and data analysis, that are useful for analysis[10].

Algorithm:

1. t-Stochastic Neighbor Embedding (t-SNE) algorithm starts taking high-dimensional Euclidean distances between data points and converts them into conditional probabilities that represent similarities between these data points. Suppose a distance matrix D is given between two input objects x_i and x_j, the conditional probability is given by $p_{j|i}$ and is mathematically represented as

$$p_{j|i} = \frac{exp(-|x_i - x_j|^2/2\sigma_i^2)}{\sum_{k\neq 1} exp(-|x_i - x_k|^2/2\sigma_i^2)}$$

where σ_i is the Gaussian variance which is centered on data point x_i. The σ for each object is chosen in such a way that the perplexity of $p_{j|i}$ has a value that is close to the user-defined perplexity. This value determines how many neighbors in the near space are considered for constructing the embedding in the low-dimensional space[10].

2. For the low-dimensional points in data space like y_i and y_j of the high-dimensional data points x_i and x_j the Cauchy distribution (t-distribution with one degree of freedom) is used to calculate the similar conditional probability, which is denoted by $q_{j|i}$

   $$q_{j|i} = \frac{exp(-|x_i - x_j|^2)}{\sum_{k \neq 1} exp(-|x_i - x_k|^2)}$$

   The difference between the data points $p_{j|i}$ and $q_{j|i}$ should be made zero for the perfect representation of the plot in low and high dimensions, which is also termed as the conditional probabilities $p_{j|i}$ and $q_{j|i}$ and should be same for a perfect replication of the similarity of data points.

3. t-SNE minimizes the divergences of Kullback-Leibler (KL) over the data points by using a method called gradient descent, to measure the minimum sum of difference of conditional probability and also these divergences are asymmetric in nature.

4. Hence, t-SNE minimizes the sum of differences in conditional probabilities and is done by the symmetric version of the SNE cost function with simple gradients. t-SNE performs high distribution in the low-dimensional space exaggerate the crowding problem and the optimization problems of SNE.

5. The entropy of this distribution increases with increase in σ_i. t-SNE uses binary search to find the value of σ_i which will produce a p_i with a perplexity that is fixed perplexity and is specified by σ_i by the user. Therefore, we define the perplexity as

   $$Perp(p_i) = 2^{H(P_i)}$$

   where, Shannon entropy of p_i measured in bits is defined by H(P_i)

   $$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i}$$

6 Experiments and Results

For my experiments, I have used a total of four datasets. Experiments and results from each dataset are described in four sub-sections below. First two datasets are the data collected from Swiss Household Panel Biographical Survey (*biofam* data) and Transition from School to Work (*mvad* data) respectively. Other dataset is the data collected from Panel Study of Income Dynamics, named *totalFUSmall* dataset. This dataset is divided into two parts: with missing values and without missing values. The following distance measures and dimensionality reduction techniques were used to extract the results.

Distance Measures:

- Optimal Matching (OM)
- Localized Optimal Matching (OMloc)
- Number of Matching Subsequences (NMS)
- Time Warp Edit Distance (TWED)

Dimensionality Reduction Measures:

- Principal Component Analysis (PCA)
- Multi-Dimensional Scaling (MDS)
- Rt-SNE

6.1 Experiments on *biofam* dataset

Following steps are done to experiment the data.

Step 1: I have used TraMineR package to start the experiments. *biofam* data comes along with the TraMineR package. Initially *biofam* data is loaded and sequential data was built using four distance measures.

Step 2: Partitioning Clustering (PAM) is done on *biofam* for first iteration, with a random selection of 20 clusters using the distance measures Optimal Matching (OM), Localized Optimal Matching (OMloc), Number of Matching Subsequences (NMS), Time Warp Edit Distance (TWED) separately.

Step 3: Internal clustering validity measures of *biofam* were graphically represented for ASWw, HG, PBC, HC using all four distance measures and are shown in the figures 10, 11, 12, and 13.

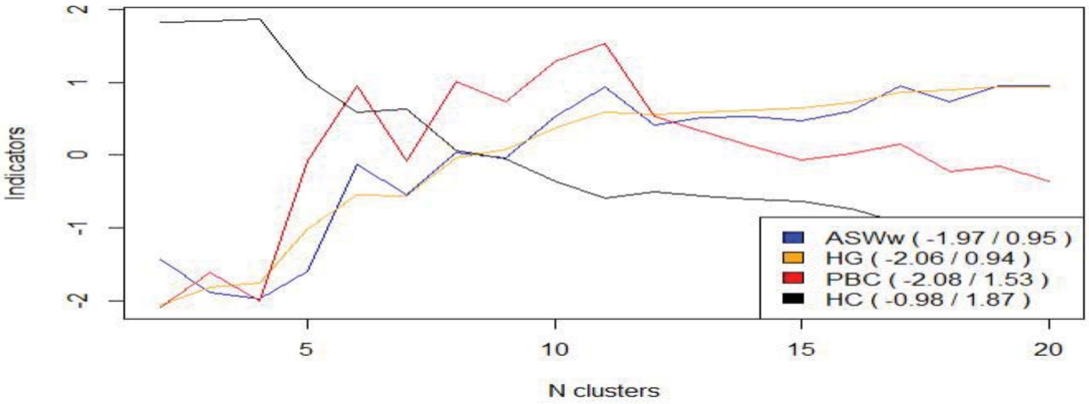

Figure 10: Internal Clustering validity measures for biofam dataset using Optimal Matching (OM)

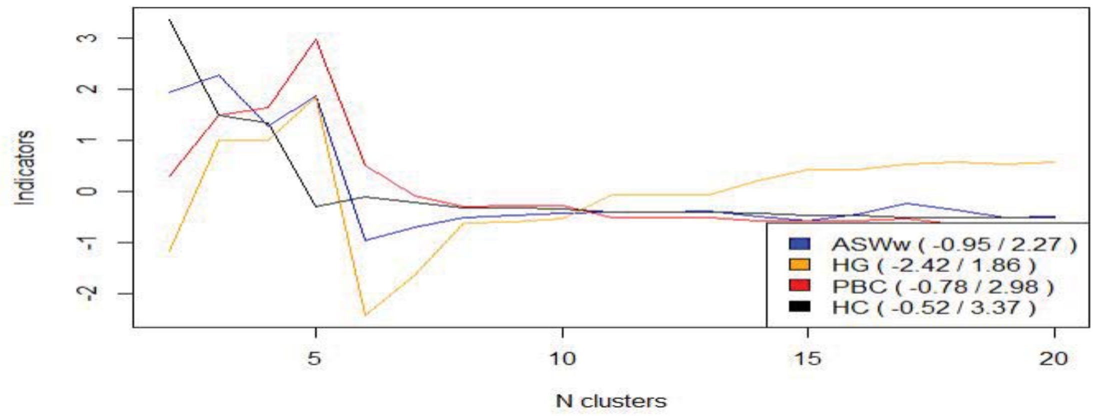

Figure 11: Internal Clustering validity measures for biofam dataset using Localized Optimal Matching(OMloc)

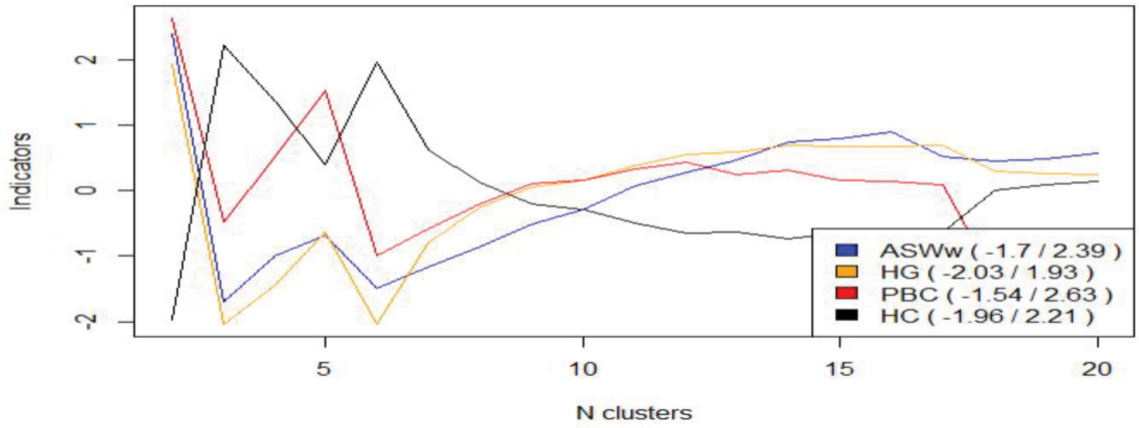

Figure 12: Internal Clustering validity measures for biofam dataset using from Number of Optimal Clusters)

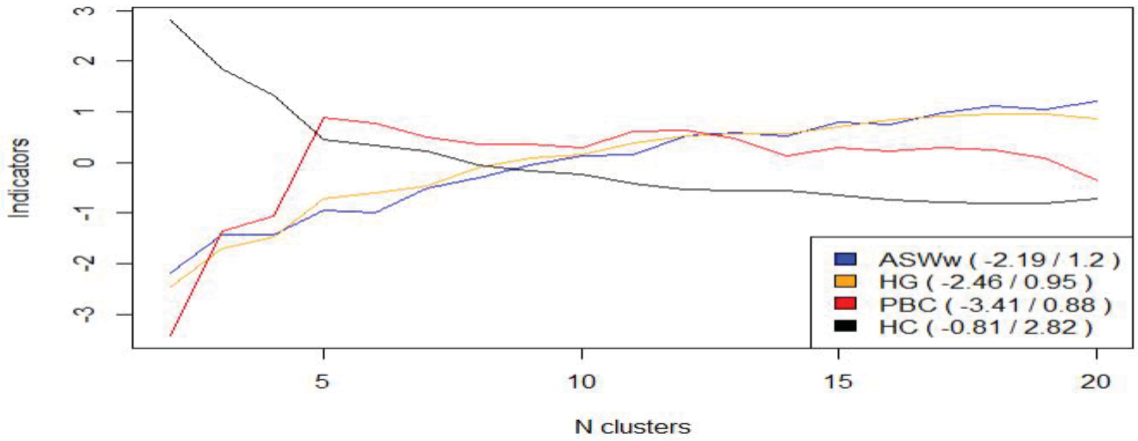

Figure 13: Internal Clustering validity measures for biofam dataset using Time Warp Edit Distance (TWED)

Step 4: From the figures 10, 11, 12, and 13, I have considered the highest peak value among ASWw, HG, PBC, HC (for all four distance masures) and took the corresponding number of clusters as *Optimum number of Clusters* for that particular distance measure.

Step 5: By taking the number of clusters from Step 4, PAM is ran for 2$^{\text{nd}}$ iteration on *biofam* data for OM, OMloc, NMS, TWED respectively and individual clusters are observed, shown in the following sequential distribution figures 14, 15, and 16.

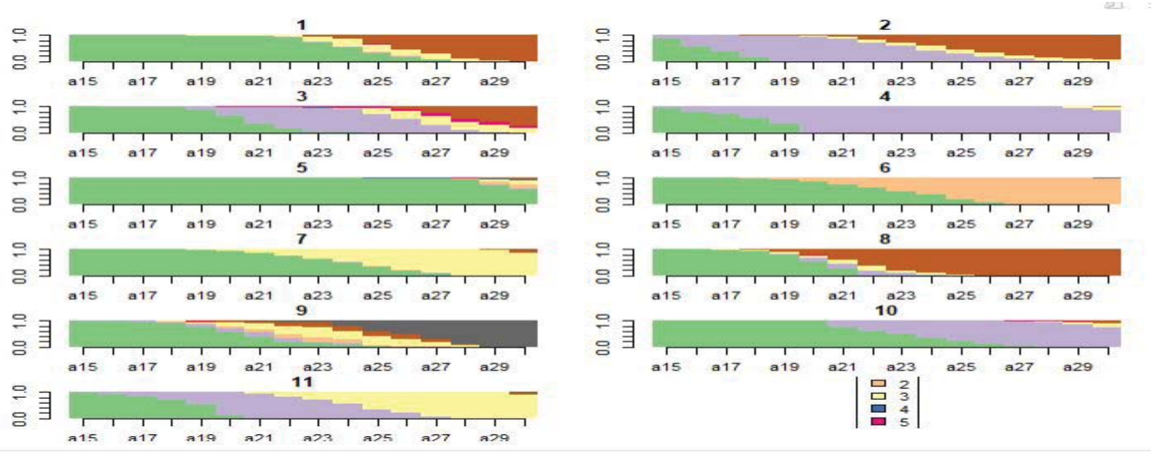

Figure 14: Sequential Distribution Plots from Number of Optimal Clusters for OM in *biofam* Dataset

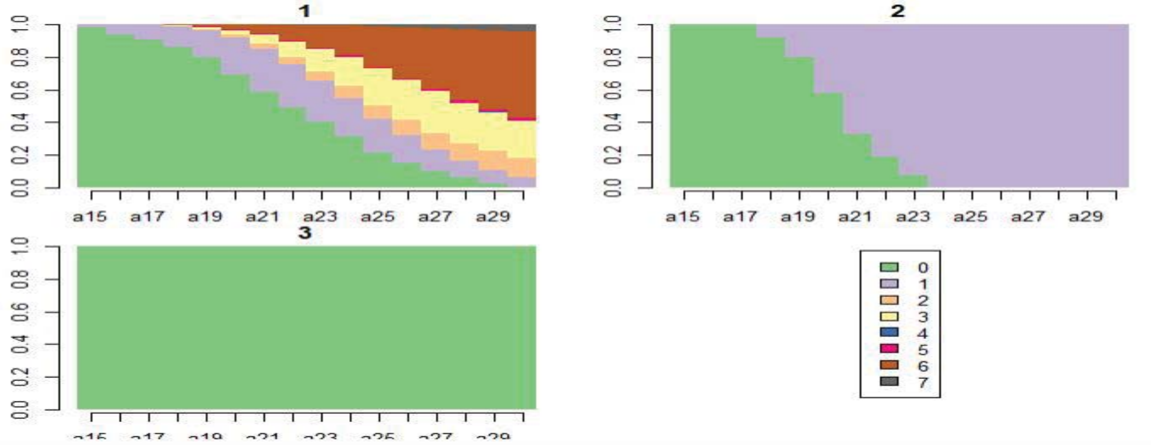

Figure 15: Sequential Distribution Plots from Number of Optimal Clusters for NMS in *biofam* Dataset

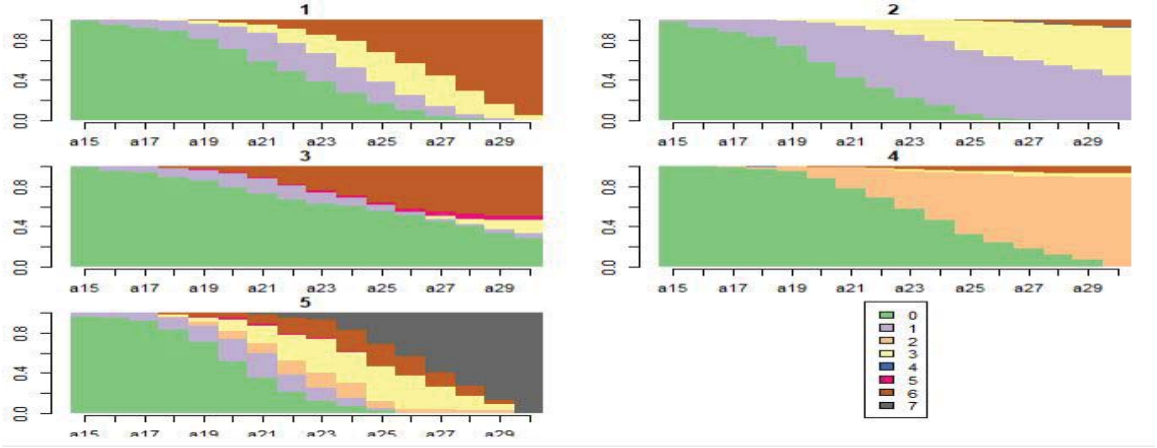

Figure 16: Sequential Distribution Plots from Number of Optimal Clusters for TWED in *biofam* Dataset

Step 6: By taking the output from Step 5, Dimensionality Reduction techniques are implemented using Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA) and Rt-SNE shown in the following graphs 17 - 25.

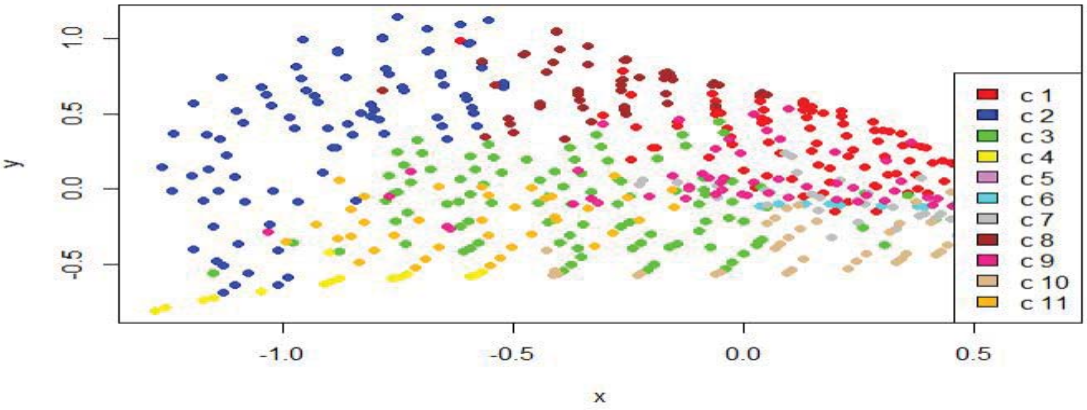

Figure 17: MDS Technique Using Optimal Matching (OM) on *biofam* Dataset

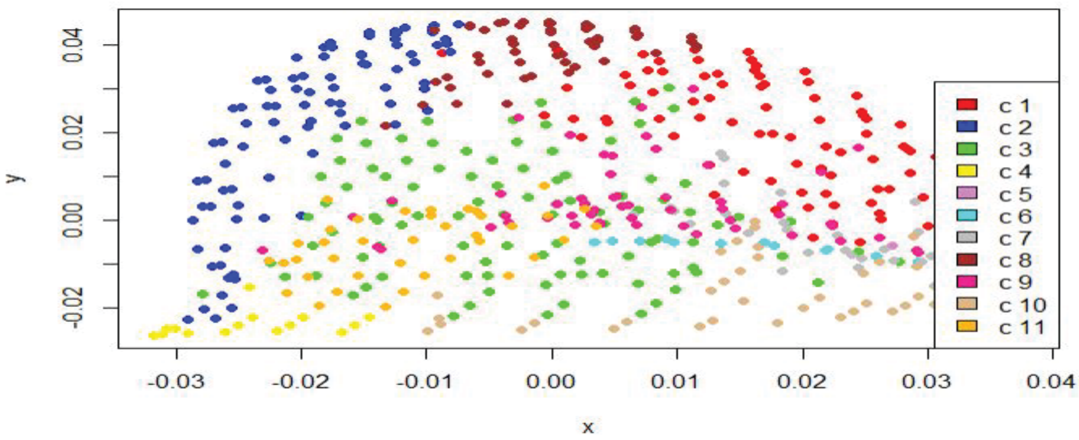

Figure 18: PCA Technique Using Optimal Matching (OM) on *biofam* Dataset

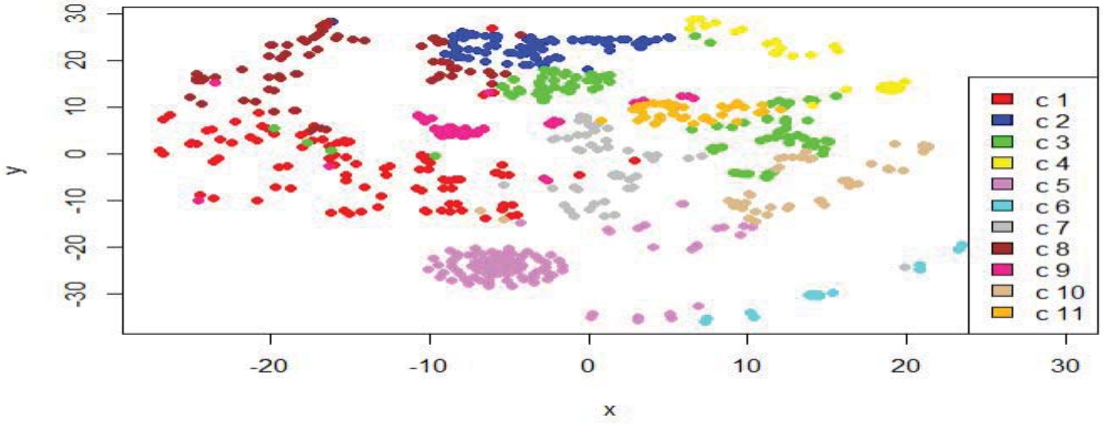

Figure 19: Rt-SNE Technique Using Optimal Matching (OM) on *biofam* Dataset

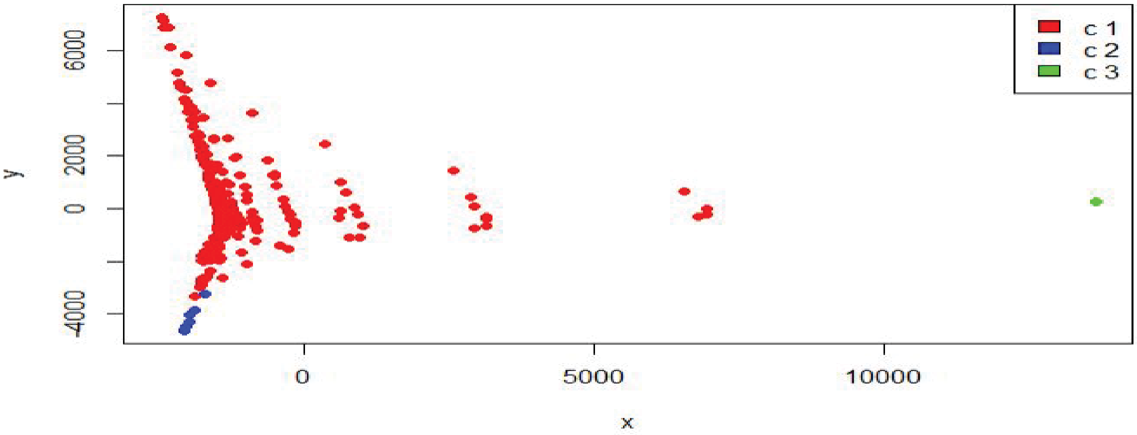

Figure 20: MDS Technique Using from Number of Optimal Clusters) on *biofam* Dataset

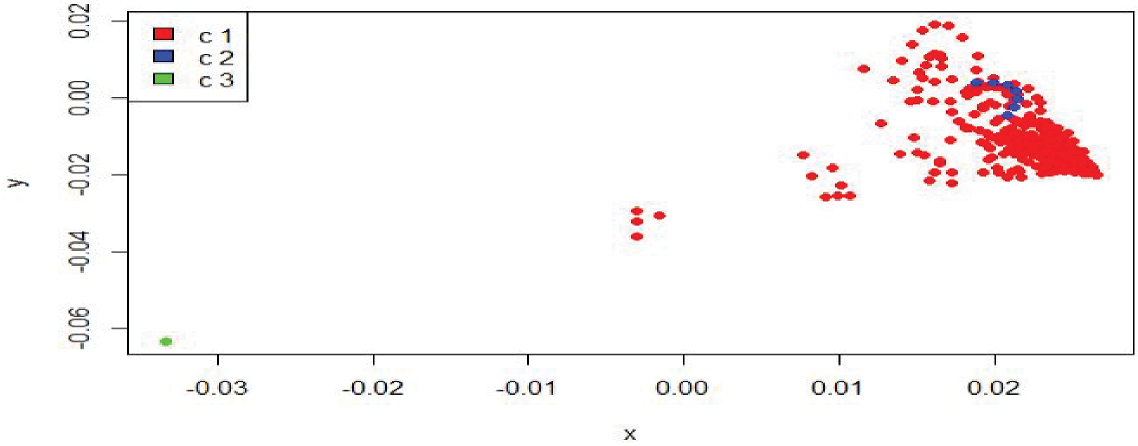

Figure 21: PCA Technique Using from Number of Optimal Clusters) on *biofam* Dataset

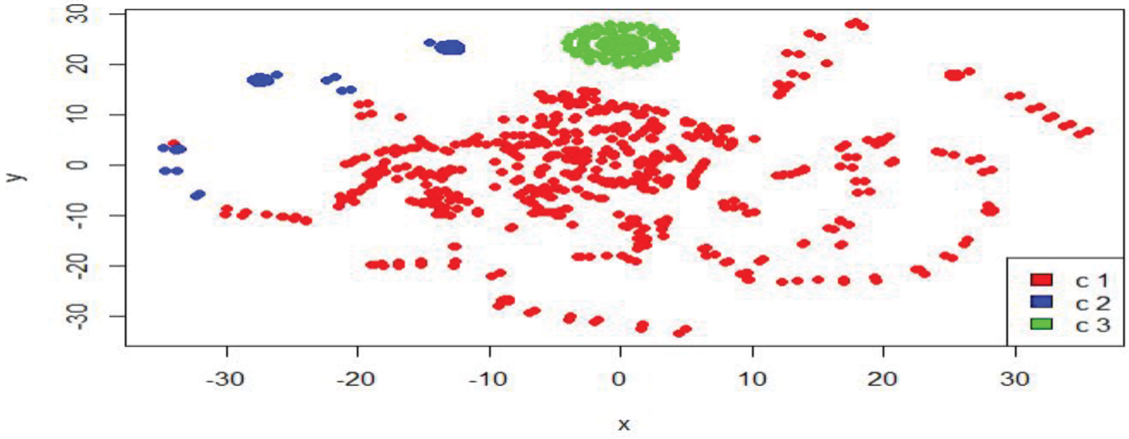

Figure 22: Rt-SNE Technique Using from Number of Optimal Clusters) on *biofam* Dataset

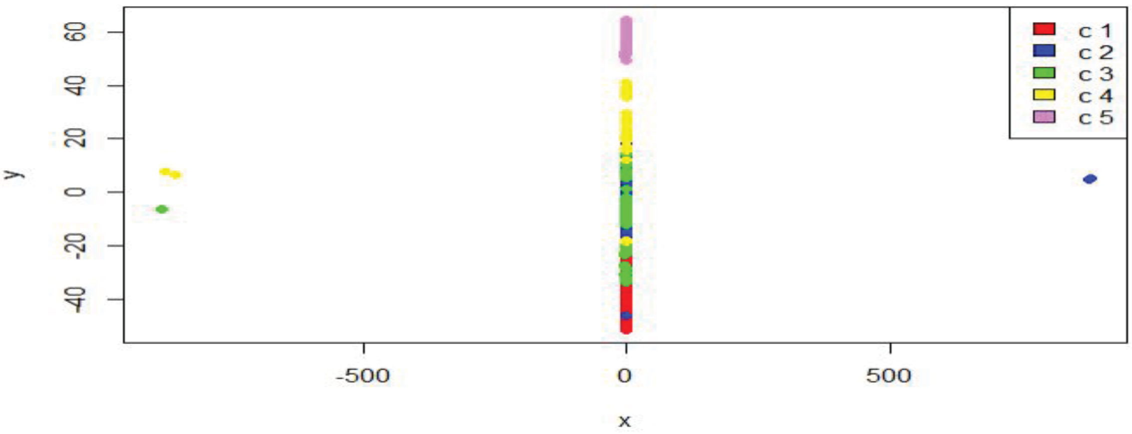

Figure 23: MDS Technique Using Time Warp Edit Distance (TWED) on *biofam* Dataset

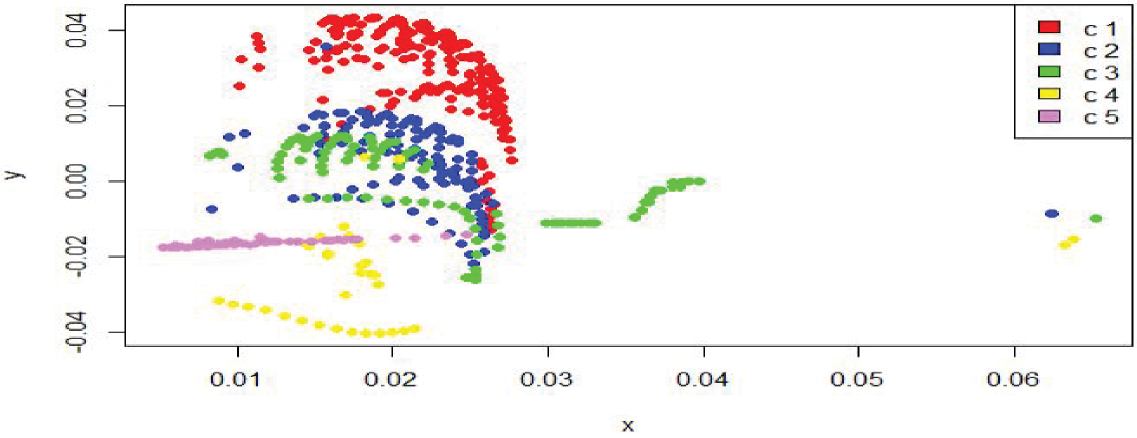

Figure 24: PCA Technique Using Time Warp Edit Distance (TWED) on *biofam* Dataset

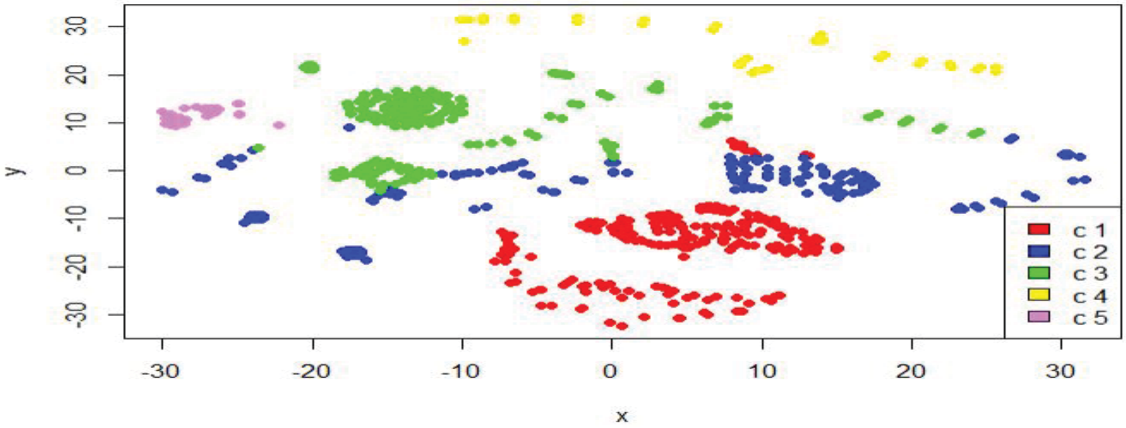

Figure 25: Rt-SNE Technique Using Time Warp Edit Distance (TWED) on *biofam* Dataset

6.2 Experiments on *mvad* dataset

Following steps are done to experiment the data.

Step 1: I have used TraMineR package to start the experiments. *mvad* data comes along with the TraMineR package. Initially *mvad* data is loaded and sequential data was built using four distance measures.

Step 2: Partitioning Clustering (PAM) is done on *mvad* for first iteration, with a random selection of 20 clusters using the distance measures Optimal Matching (OM), Localized Optimal Matching (OMloc), from Number of Optimal Clusters), Time Warp Edit Distance (TWED) separately.

Step 3: Internal clustering validity measures of *mvad* were graphically represented for ASWw, HG, PBC, HC using all four distance measures and are shown in the figures 26, 27, 28, and 29.

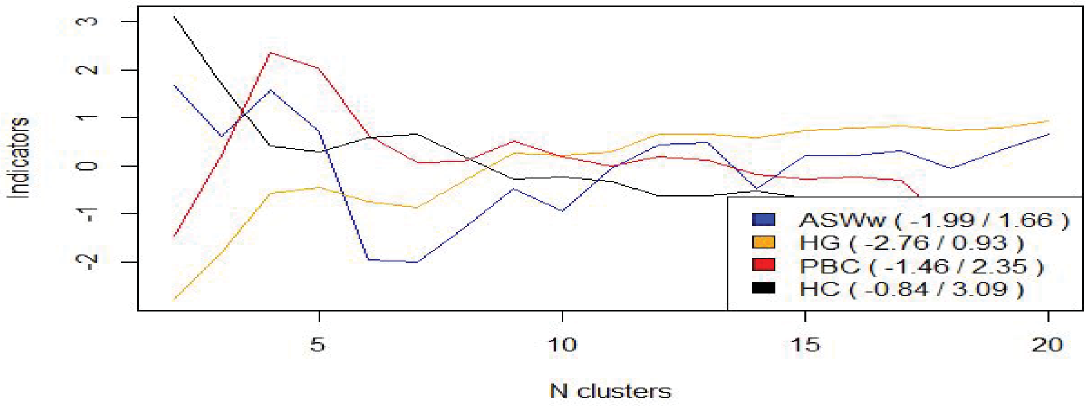

Figure 26: Internal Clustering validity measures for mvad dataset using Optimal Matching (OM)

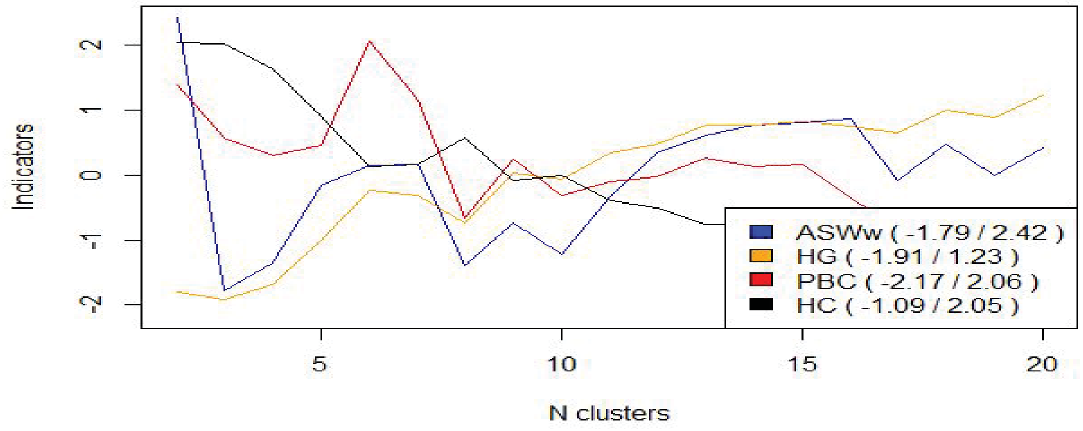

Figure 27: Internal Clustering validity measures for mvad dataset using Localized Optimal Matching(OMloc)

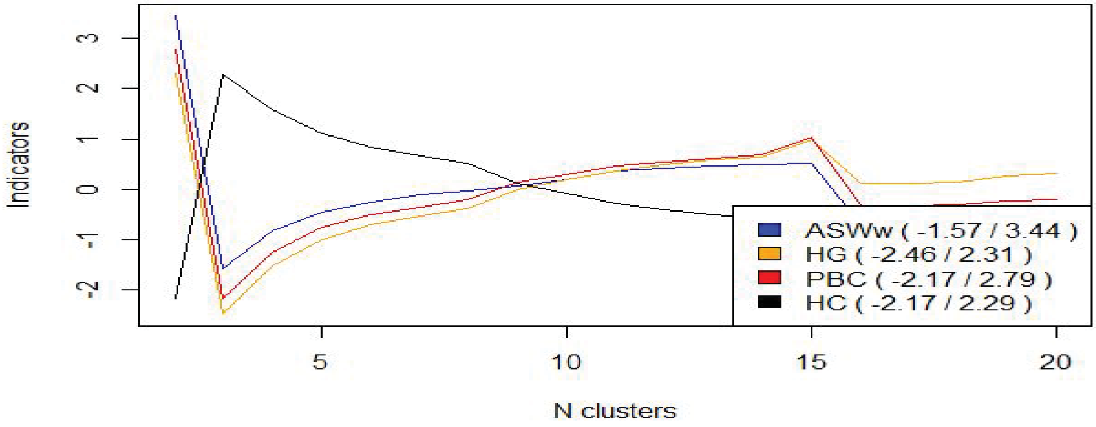

Figure 28: Internal Clustering validity measures for mvad dataset using from Number of Optimal Clusters)

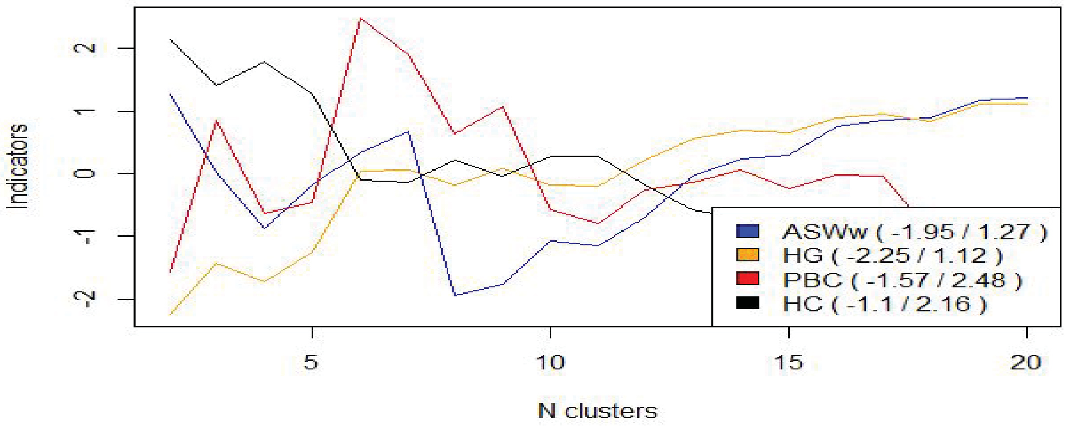

Figure 29: Internal Clustering validity measures for mvad dataset using Time Warp Edit Distance (TWED)

Step 4: From the figures 26, 27, 28, and 29, I have considered the highest peak value among ASWw, HG, PBC, HC (for all four distance masures) and took the corresponding number of clusters as *Optimum number of Clusters* for that particular distance measure.

Step 5: By taking the number of clusters from Step 4, PAM is ran for 2$^{\text{nd}}$ iteration on *mvad* data for OM, OMloc, NMS, TWED respectively and clusters are observed, shown in the following sequential distribution figures 30, 31, 32, and 33.

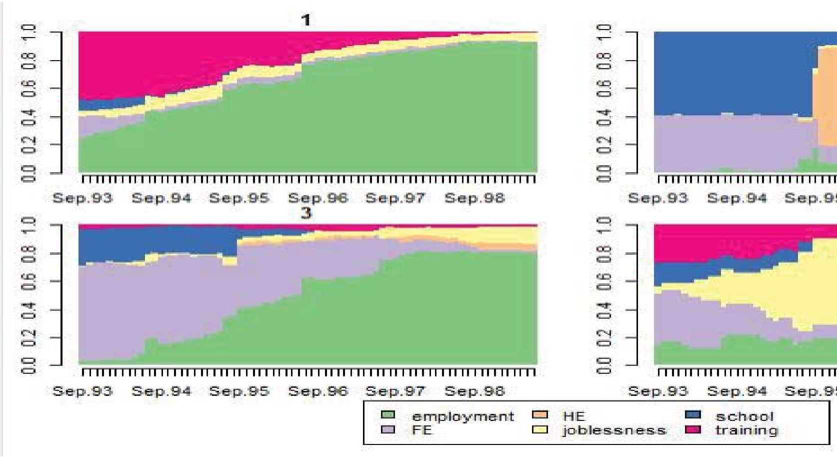

Figure 30: Sequential Distribution Plots from Number of Optimal Clusters for OM in *mvad* Dataset

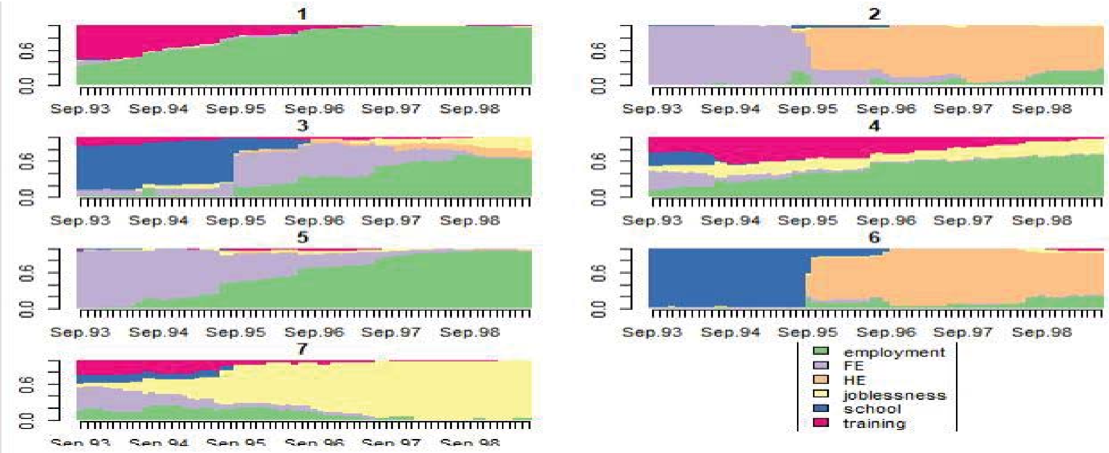

Figure 31: Sequential Distribution Plots from Number of Optimal Clusters for OMloc in *mvad* Dataset

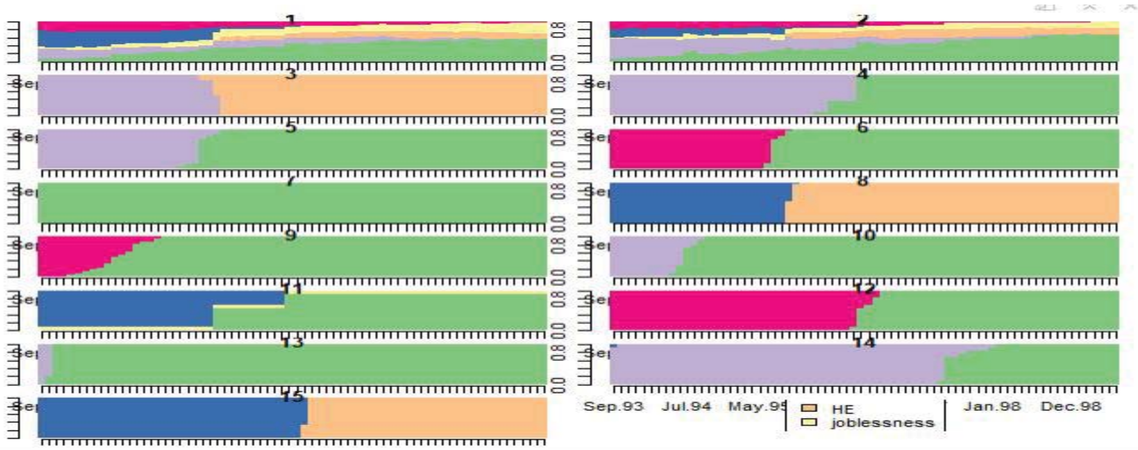

Figure 32: Sequential Distribution Plots from Number of Optimal Clusters for NMS in *biofam* Dataset

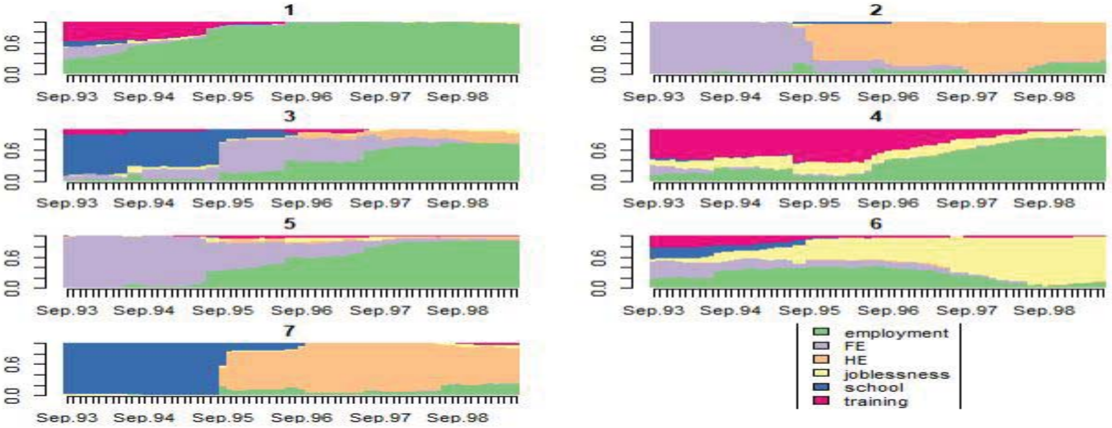

Figure 33: Sequential Distribution Plots from Number of Optimal Clusters for TWED in *biofam* Dataset

Step 6: By taking the output from Step 5, Dimensionality Reduction techniques are implemented using Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA) and Rt-SNE shown in the following graphs.

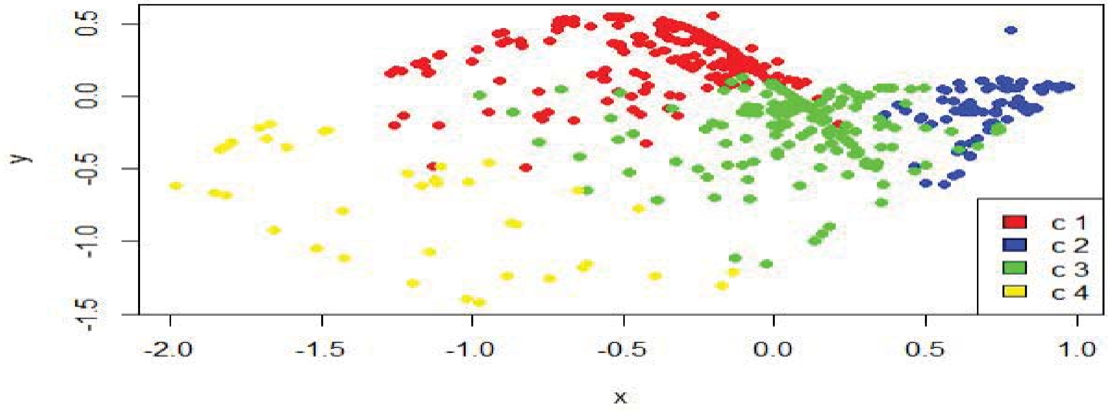

Figure 34: MDS Technique Using Optimal Matching (OM) on *mvad* Dataset

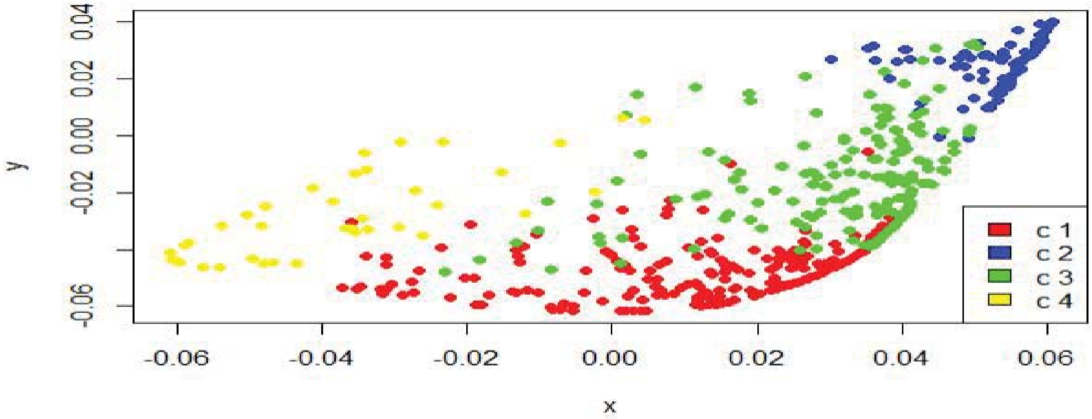

Figure 35: PCA Technique Using Optimal Matching (OM) on *mvad* Dataset

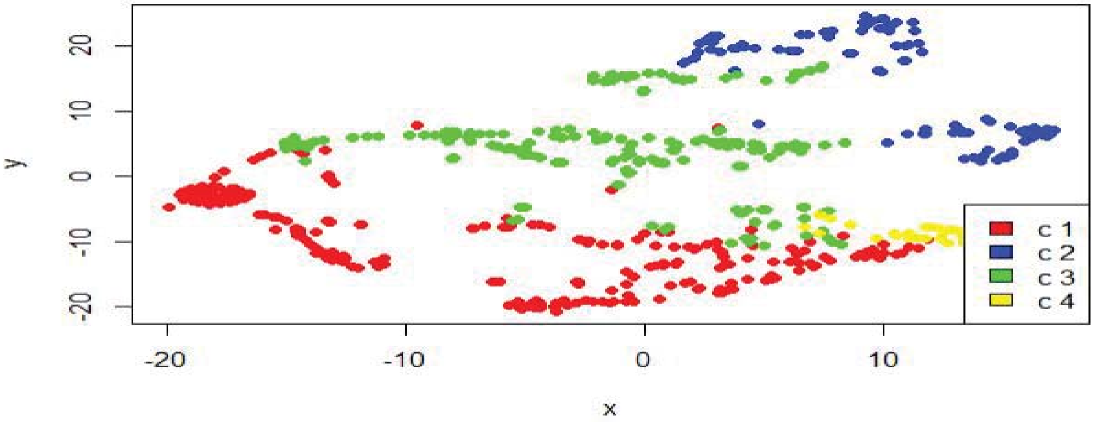

Figure 36: Rt-SNE Technique Using Optimal Matching (OM) on *mvad* Dataset

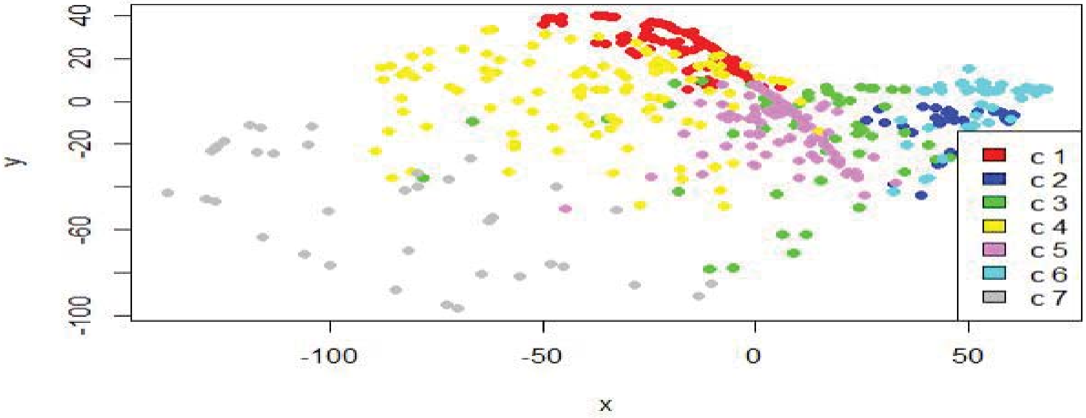

Figure 37: MDS Technique Using Localized Optimal Matching (OMloc) on *mvad* Dataset

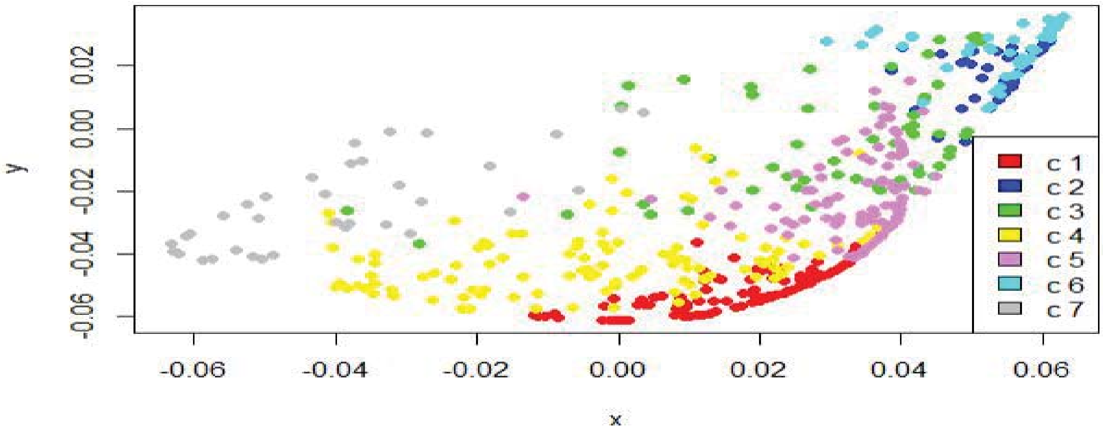

Figure 38: PCA Technique Using Localized Optimal Matching (OMloc) on *mvad* Dataset

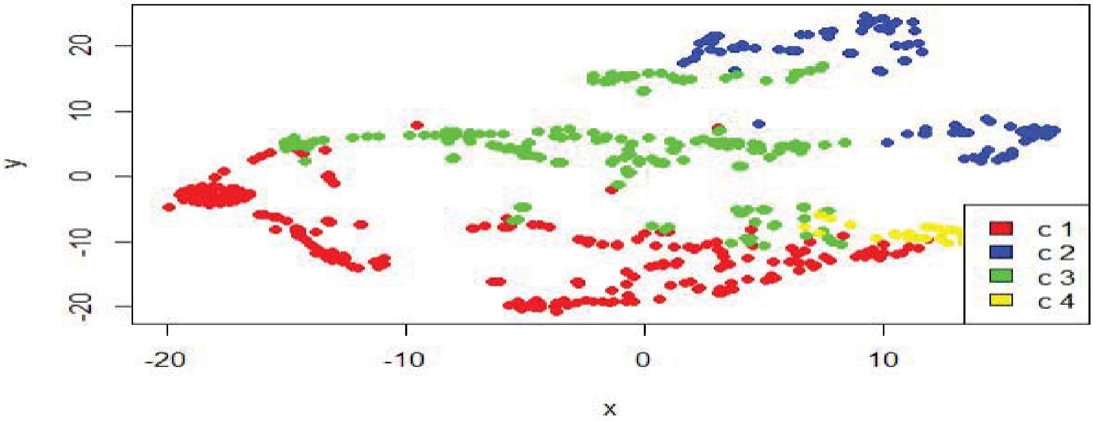

Figure 39: Rt-SNE Technique Using Localized Optimal Matching (OM) on *mvad* Dataset

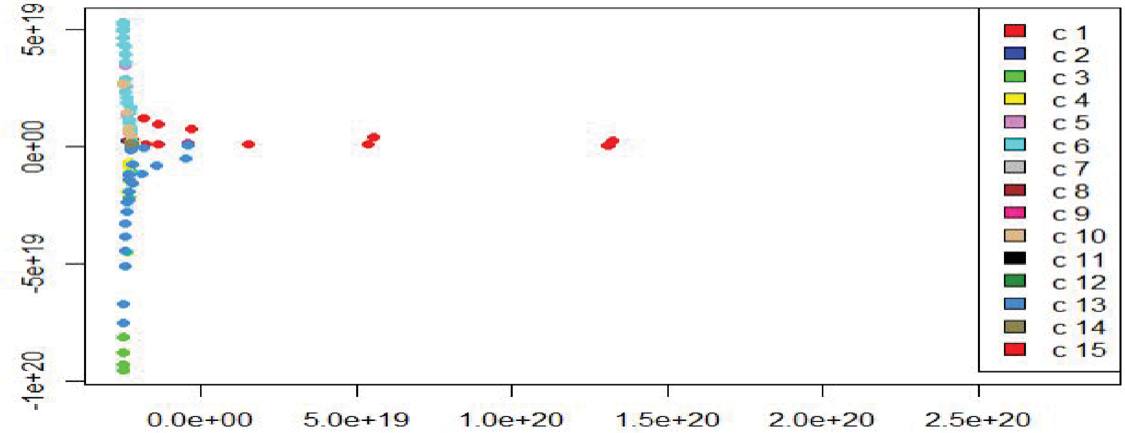

Figure 40: MDS Technique Using from Number of Optimal Clusters) on *mvad* Dataset

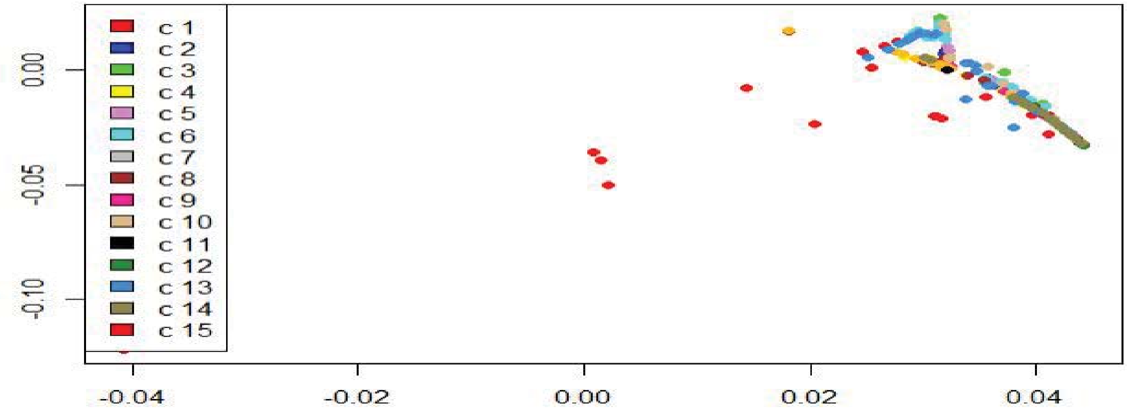

Figure 41: PCA Technique Using from Number of Optimal Clusters) on *mvad* Dataset

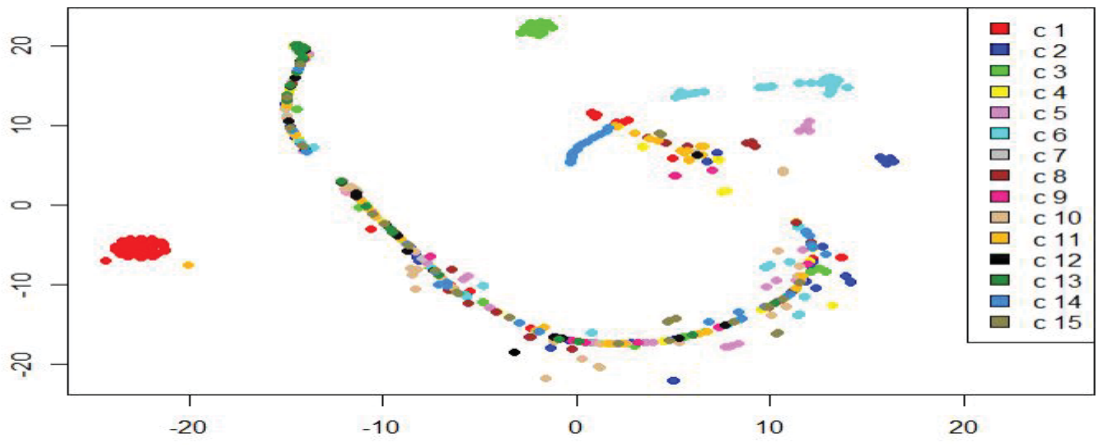

Figure 42: Rt-SNE Technique Using from Number of Optimal Clusters) on *mvad* Dataset

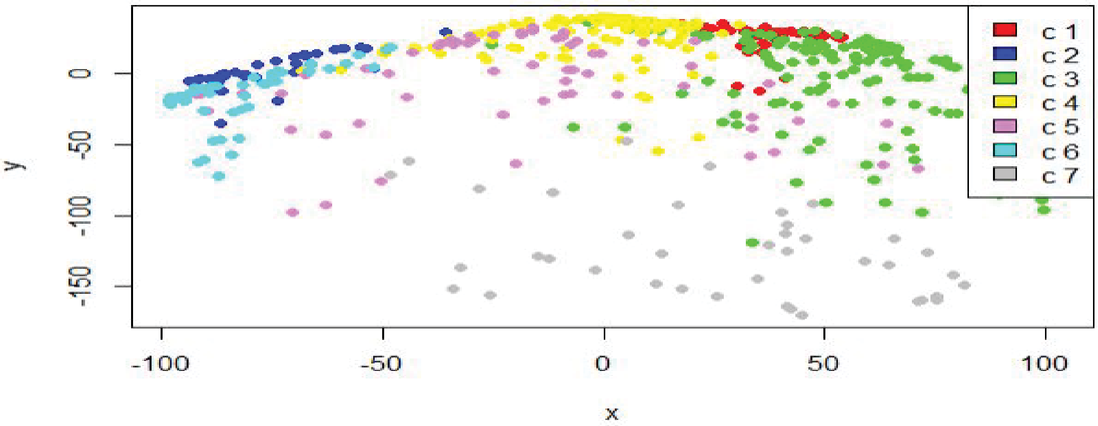

Figure 43: MDS Technique Using Time Warp Edit Distance (TWED) on *mvad* Dataset

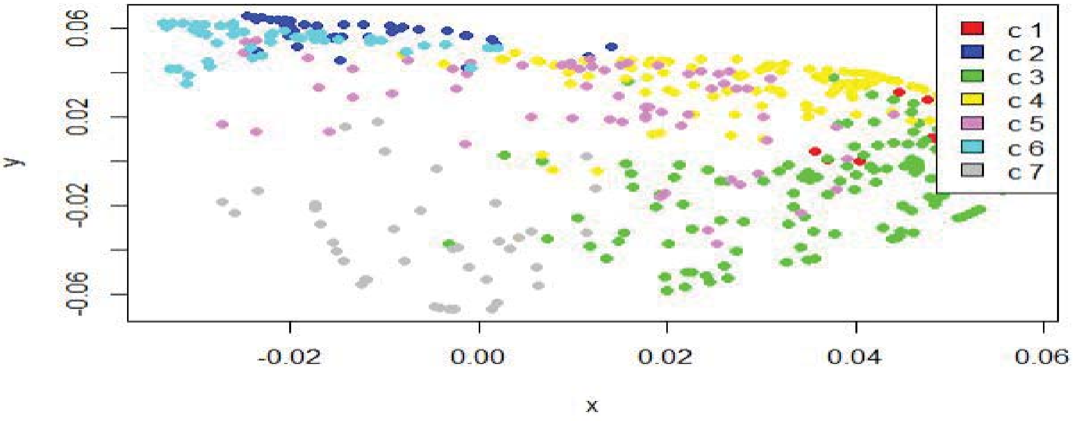

Figure 44: PCA Technique Using Time Warp Edit Distance (TWED) on *mvad* Dataset

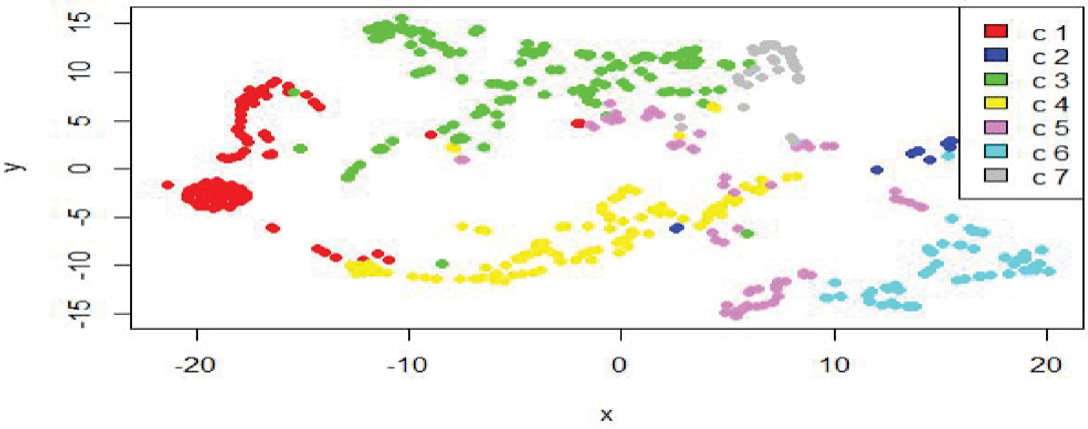

Figure 45: Rt-SNE Technique Using Time Warp Edit Distance (TWED) on *mvad* Dataset

6.3 Experiments on *totalFUSmall* dataset with no missing values

Following steps are done to experiment the data.

Step 1: Initially *totalFUSmall* data with no missing values is loaded and sequential data was built using four distance measures.

Step 2: Partitioning Clustering (PAM) is done on *totalFUSmall* with no missing values, for first iteration, with a random selection of 20 clusters using the distance measures Optimal Matching (OM), Localized Optimal Matching (OMloc), Number of Matching Subsequences (NMS), Time Warp Edit Distance (TWED) separately.

Step 3: Internal clustering validity measures of *totalFUSmall* with no missing values were graphically represented for ASWw, HG, PBC, HC using all four distance measures and are shown in the figures 46, 47, 48, and 49.

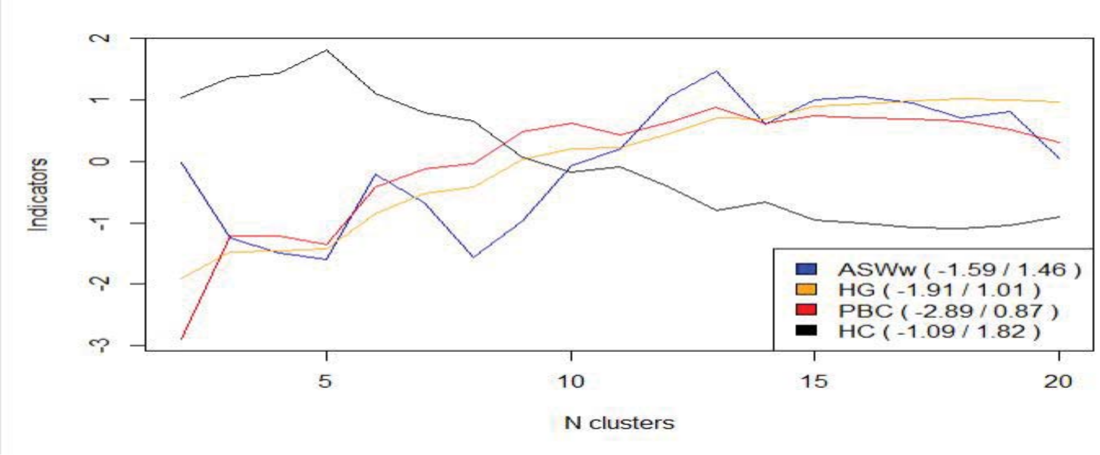

Figure 46: Internal Clustering validity measures for *totalFUSmall* dataset with no missing using Optimal Matching (OM)

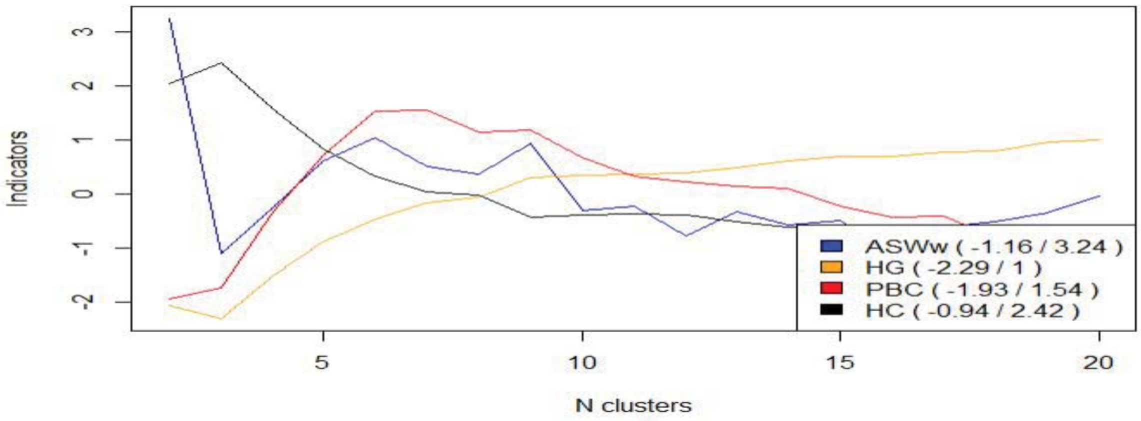

Figure 47: Internal Clustering validity measures for *totalFUSmall* dataset with no missing values, using Localized Optimal Matching(OMloc)

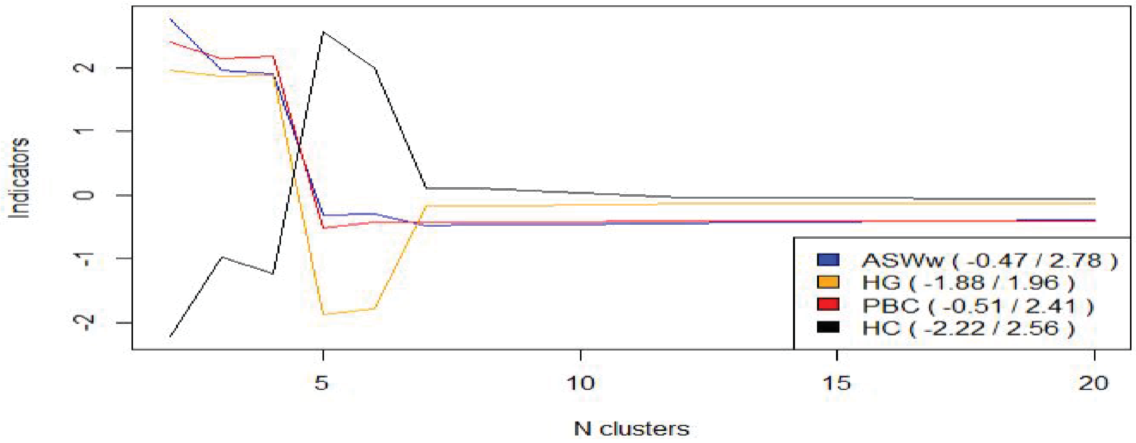

Figure 48: Internal Clustering validity measures for *totalFUSmall* dataset with no missing values using from Number of Optimal Clusters)

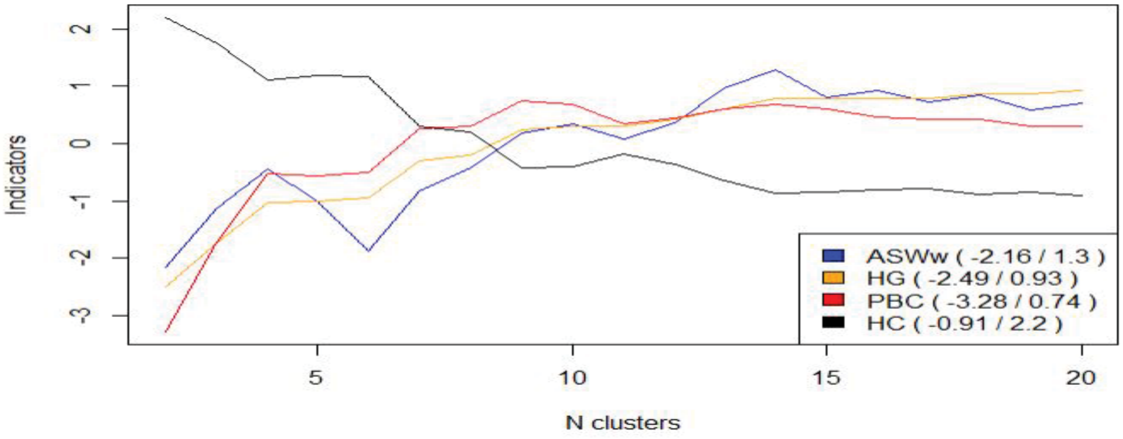

Figure 49: Internal Clustering validity measures for totalFUSmall dataset using Time Warp Edit Distance (TWED)

Step 4: From the figures 46, 47, 48, and 49, I have considered the highest peak value among ASWw, HG, PBC, HC (for all four distance masures) and took the corresponding number of clusters as *Optimum number of Clusters* for that particular distance measure.

Step 5: By taking the number of clusters from Step 4, PAM is ran for 2$^{\text{nd}}$ iteration on *totalFUSmall* data with no missing values, for OM, NMS, TWED respectively and individual clusters are observed, shown in the following sequential figures 50, 51, and 52.

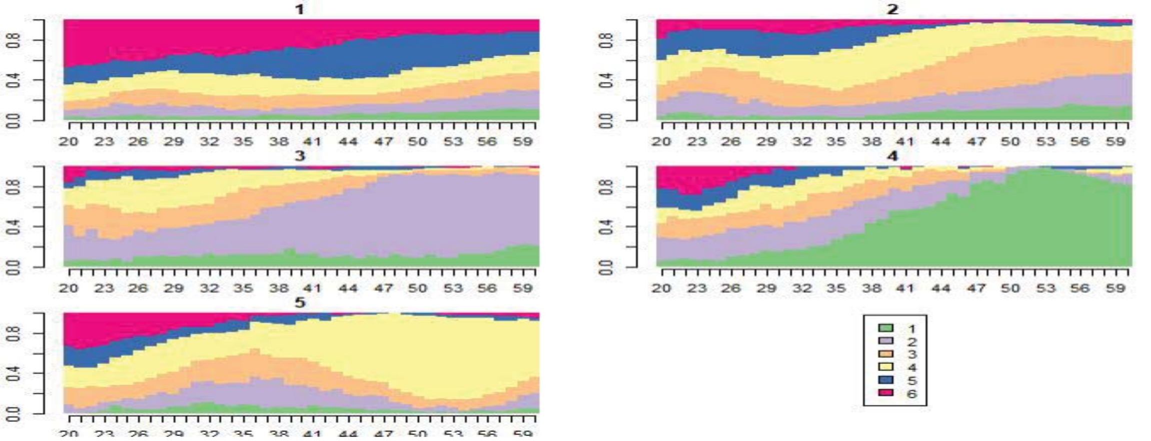

Figure 50: Sequential Distribution Plots from Number of Optimal Clusters for OM in *totalFUSmall* Dataset with *no missing values*

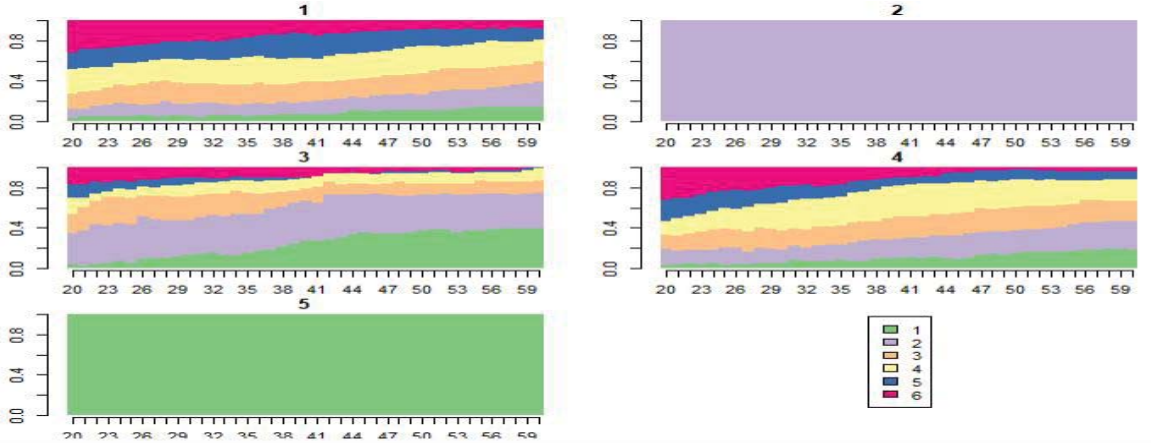

Figure 51: Sequential Distribution Plots from Number of Optimal Clusters for NMS in *totalFUSmall* Dataset with *no missing values*

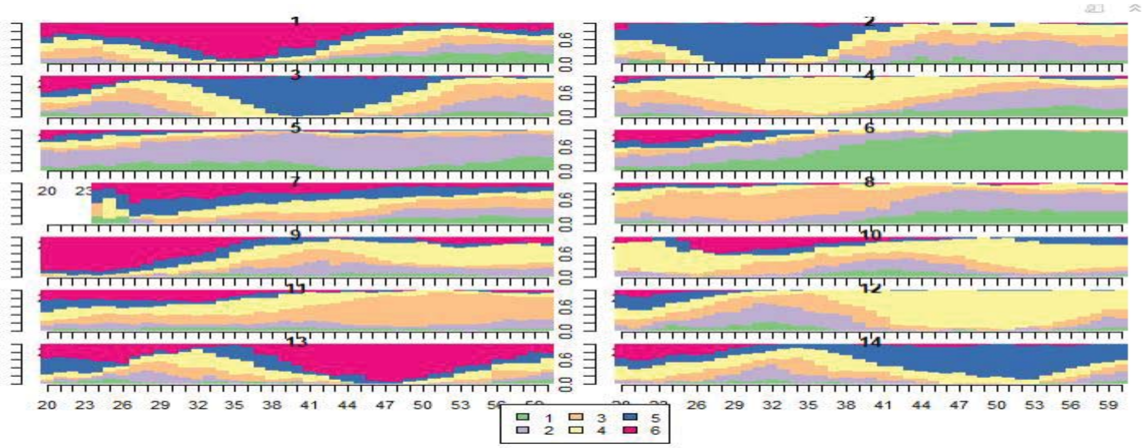

Figure 52: Sequential Distribution Plots from Number of Optimal Clusters for TWED in *totalFUSmall* Dataset with *no missing values*

Step 6: By taking the output from Step 5, Dimensionality Reduction techniques are implemented using Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA) and Rt-SNE shown in the following graphs 53 - 61.

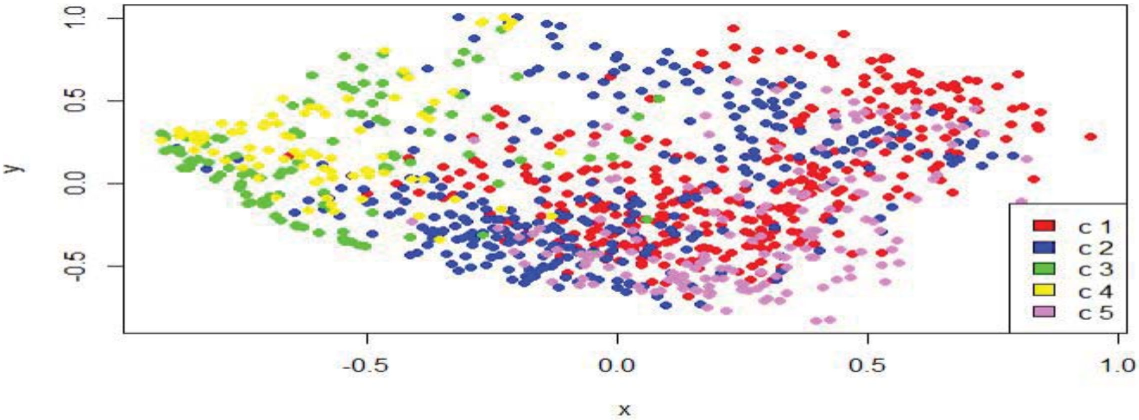

Figure 53: MDS Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with no missing values

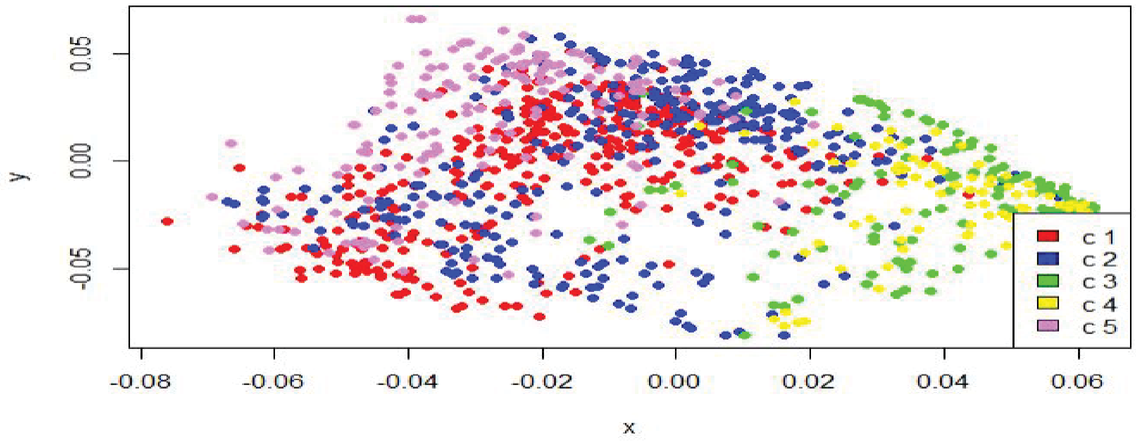

Figure 54: PCA Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with no missing values

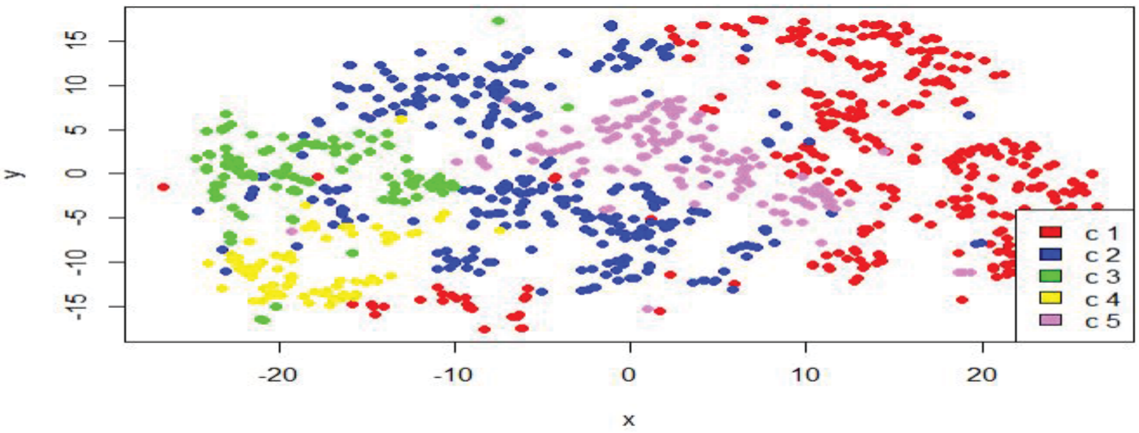

Figure 55: Rt-SNE Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with no missing values

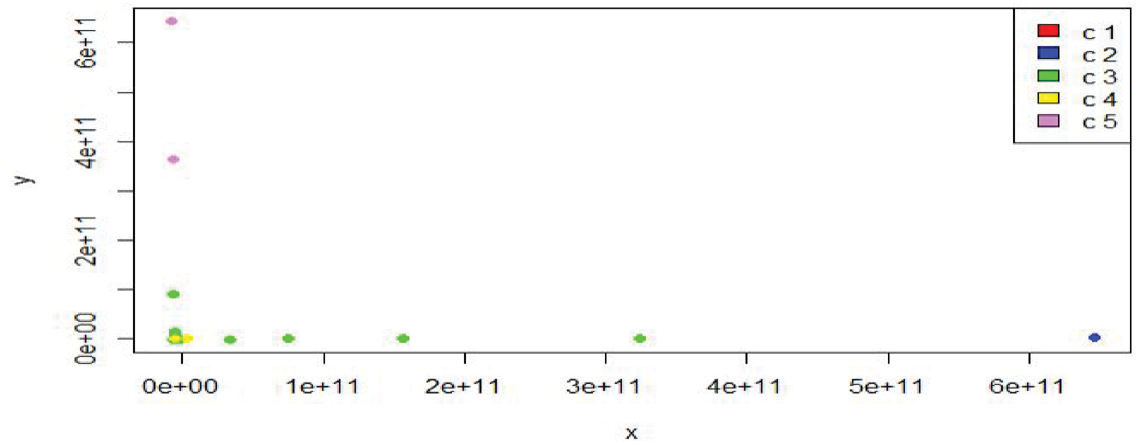

Figure 56: MDS Technique Using from Number of Optimal Clusters) on *totalFUSmall* Dataset with no missing values

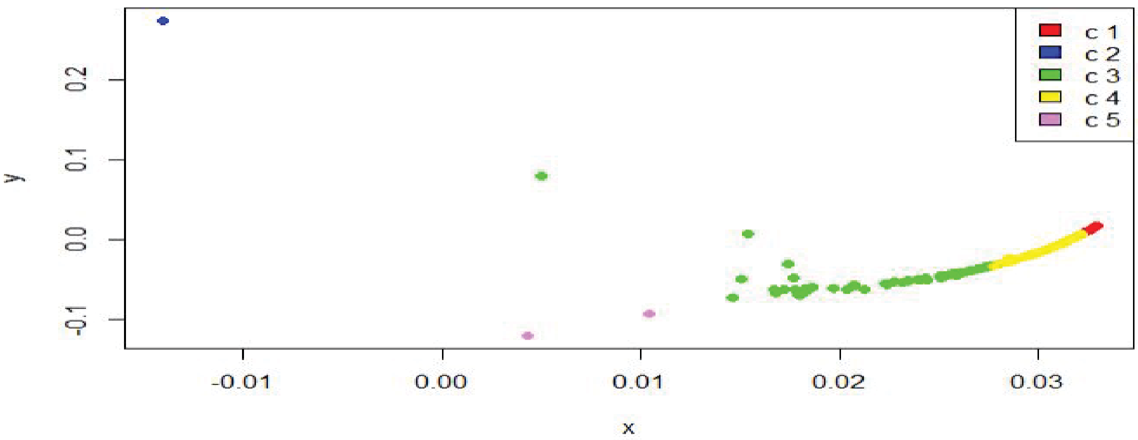

Figure 57: PCA Technique Using from Number of Optimal Clusters) on *totalFUSmall* Dataset with no missing values

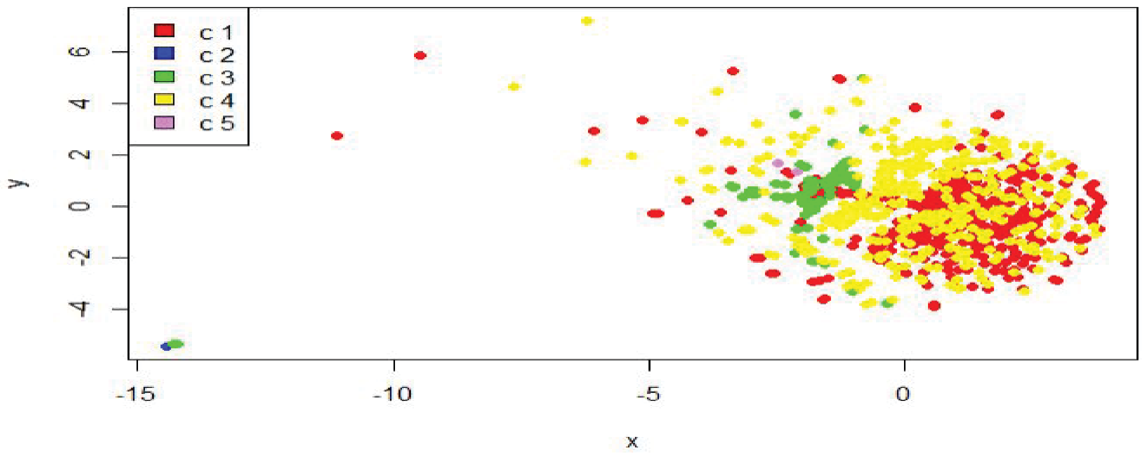

Figure 58: Rt-SNE Technique Using from Number of Optimal Clusters) on *totalFUSmall* Dataset with no missing values

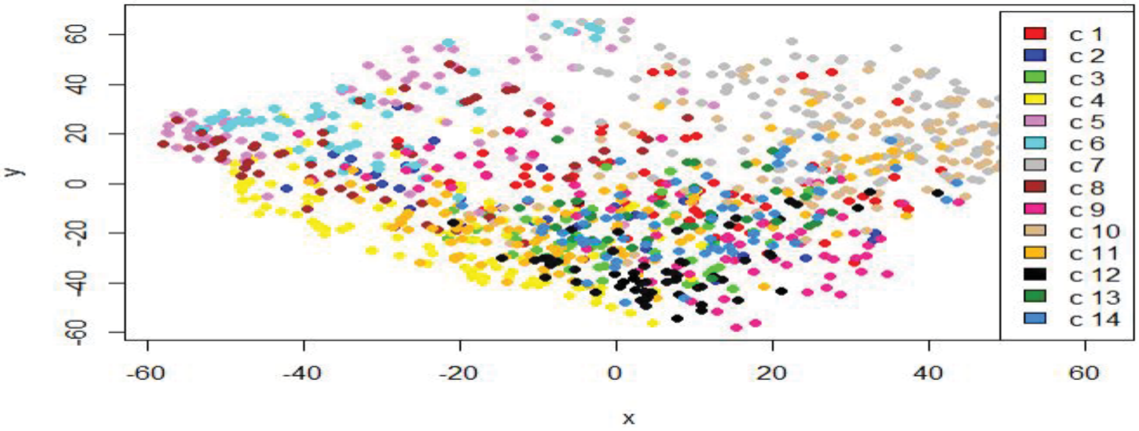

Figure 59: MDS Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with no missing values

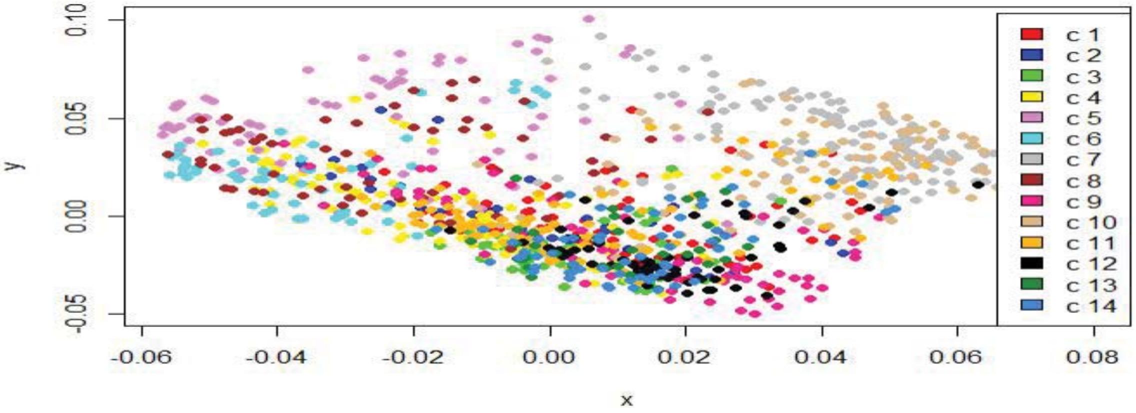

Figure 60: PCA Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with no missing values

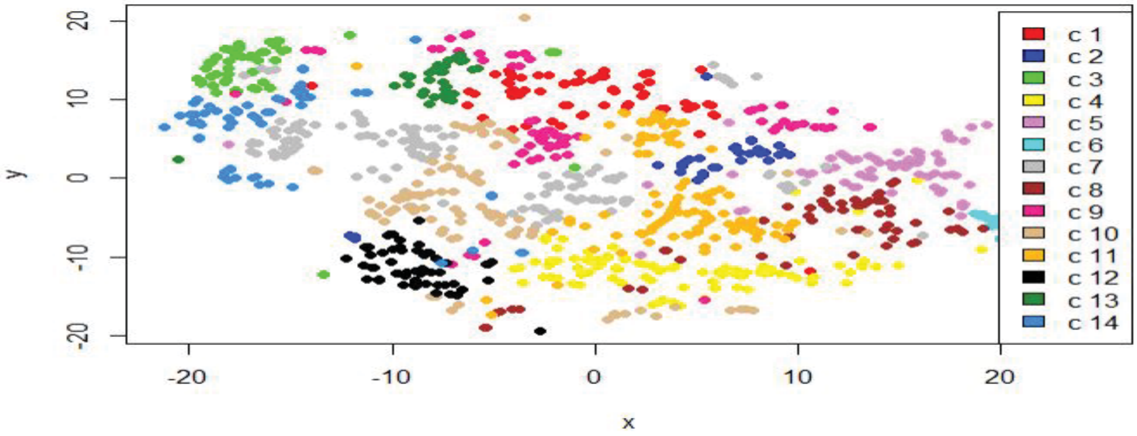

Figure 61: Rt-SNE Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with no missing values

6.4 Experiments on *totalFUSmall* dataset with missing values

Following steps are done to experiment the data.

Step 1: Initially *totalFUSmall* data with missing values is loaded and sequential data was built using four distance measures.

Step 2: Partitioning Clustering (PAM) is done on *totalFUSmall* with missing values, for first iteration, with a random selection of 20 clusters using the distance measures Optimal Matching (OM), Localized Optimal Matching (OMloc), Number of Matching Subsequences (NMS), Time Warp Edit Distance (TWED) separately.

Step 3: Internal clustering validity measures of *totalFUSmall* with missing values were graphically represented for ASWw, HG, PBC, HC using all four distance measures and are shown in the figures 62, 63, 64, and 65.

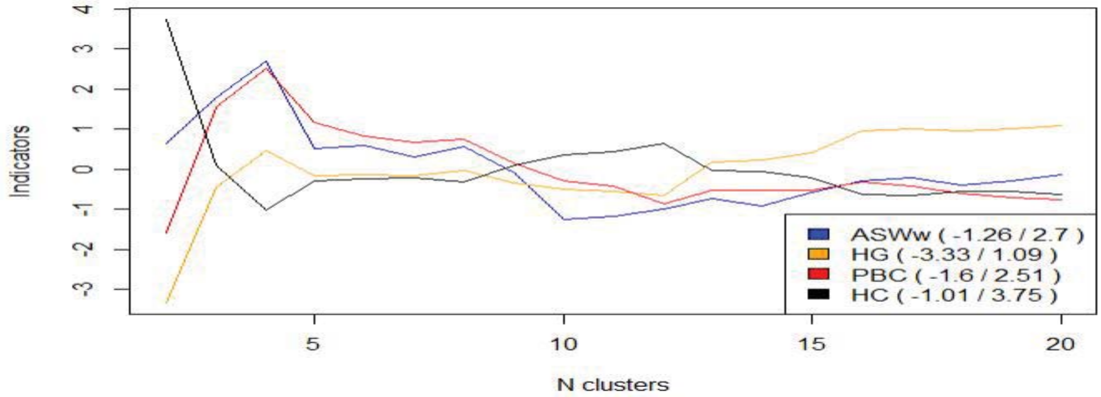

Figure 62: Internal Clustering validity measures for *totalFUSmall* dataset with missing using Optimal Matching (OM)

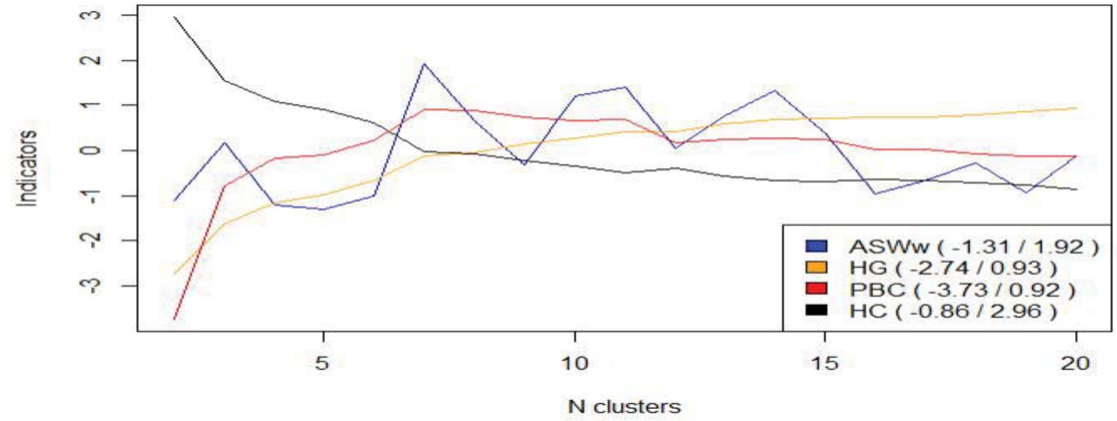

Figure 63: Internal Clustering validity measures for *totalFUSmall* dataset with missing values, using Localized Optimal Matching(OMloc)

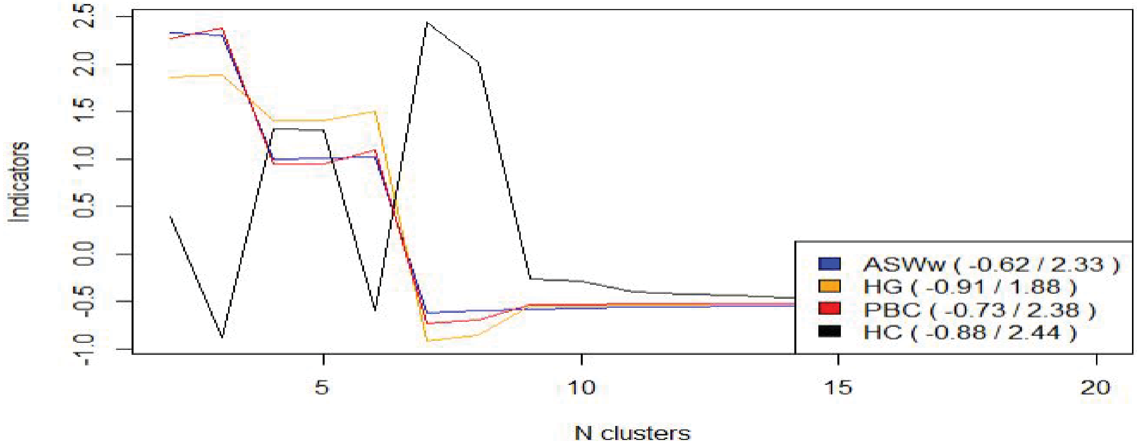

Figure 64: Internal Clustering validity measures for *totalFUSmall* dataset with missing values using from Number of Optimal Clusters)

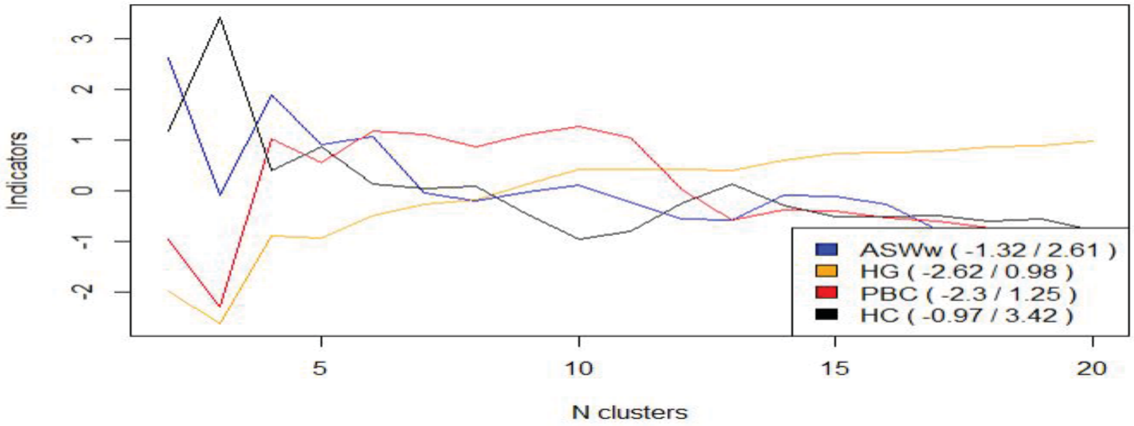

Figure 65: Internal Clustering validity measures for totalFUSmall dataset using Time Warp Edit Distance (TWED)

Step 4: From the figures 62, 63, 64, and 65, I have considered the highest peak value among ASWw, HG, PBC, HC (for all four distance masures) and took the corresponding number of clusters as *Optimum number of Clusters* for that particular distance measure.

Step 5: By taking the number of clusters from Step 4, PAM is ran for 2$^{\text{nd}}$ iteration on *totalFUSmall* data with missing values, for OM, NMS, TWED respectively and clusters are observed, shown in the following figures 66, 67, and 68.

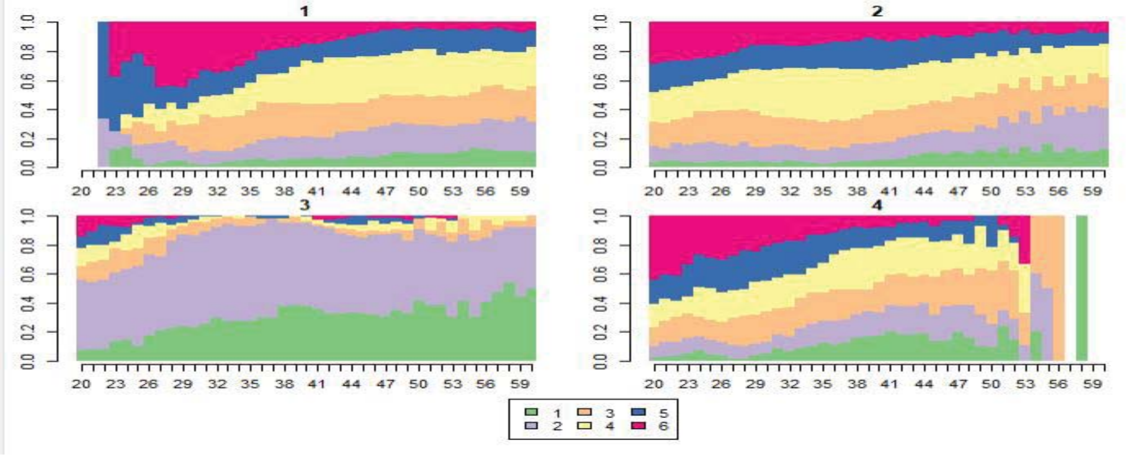

Figure 66: Sequential Distribution Plots from Number of Optimal Clusters for OM in *totalFUSmall* Dataset with *missing values*

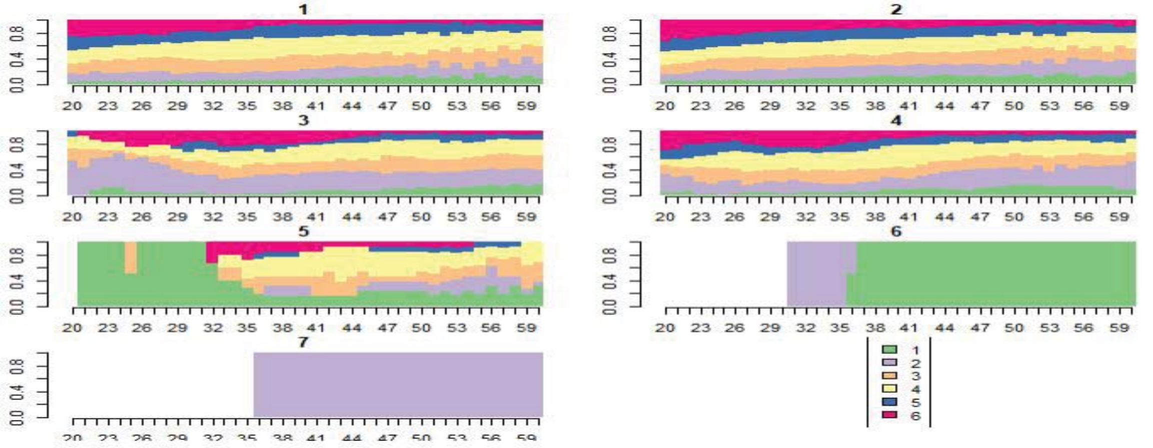

Figure 67: Sequential Distribution Plots from Number of Optimal Clusters for NMS in *totalFUSmall* Dataset with *missing values*

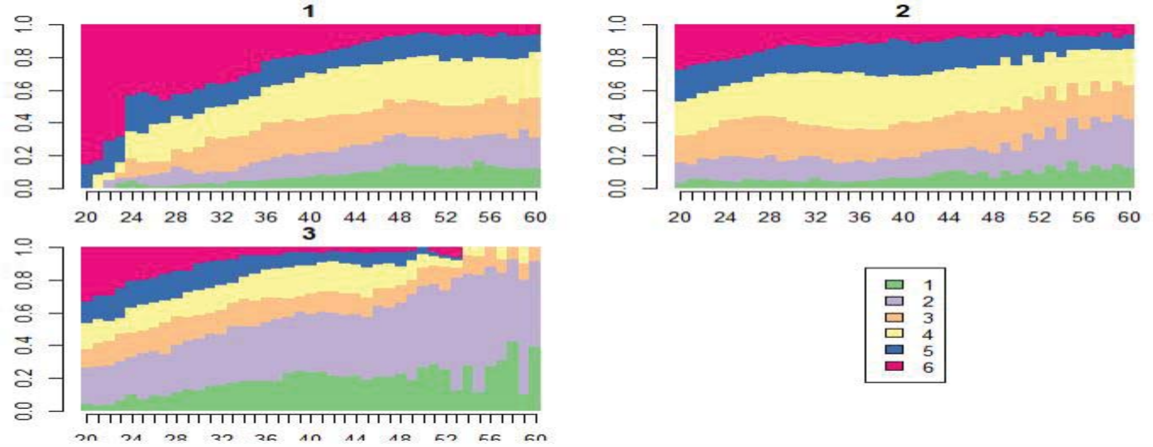

Figure 68: Sequential Distribution Plots from Number of Optimal Clusters for TWED in *totalFUSmall* Dataset with *missing values*

Step 6: By taking the output from Step 5, Dimensionality Reduction techniques are implemented using Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA) and Rt-SNE shown in the following graphs 69 - 77.

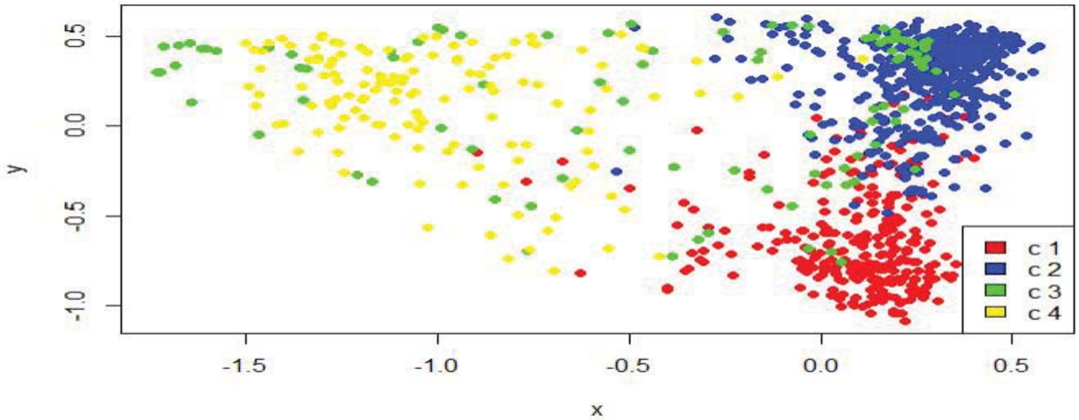

Figure 69: MDS Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with missing values

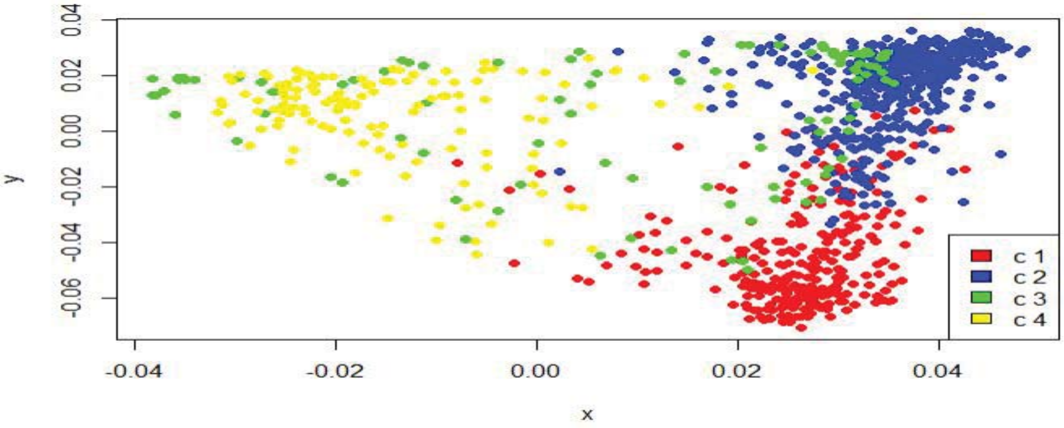

Figure 70: PCA Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with missing values

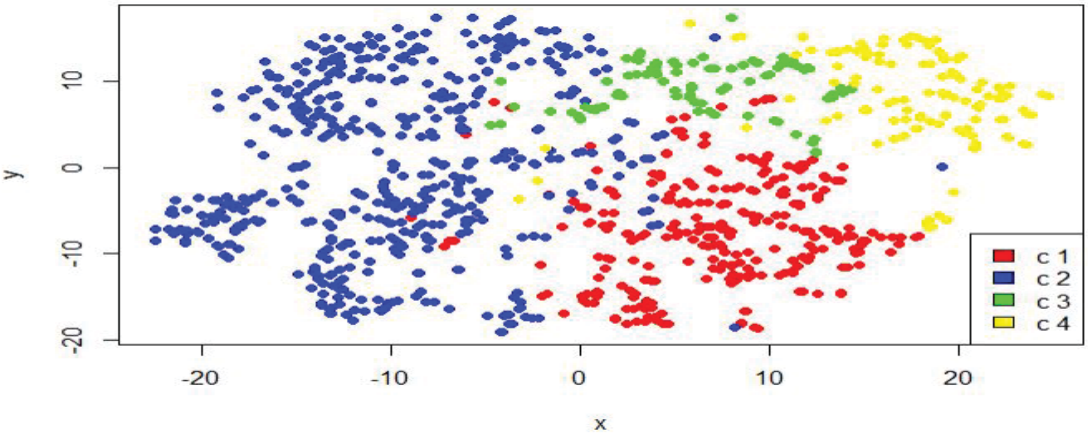

Figure 71: Rt-SNE Technique Using Optimal Matching (OM) on *totalFUSmall* Dataset with missing values

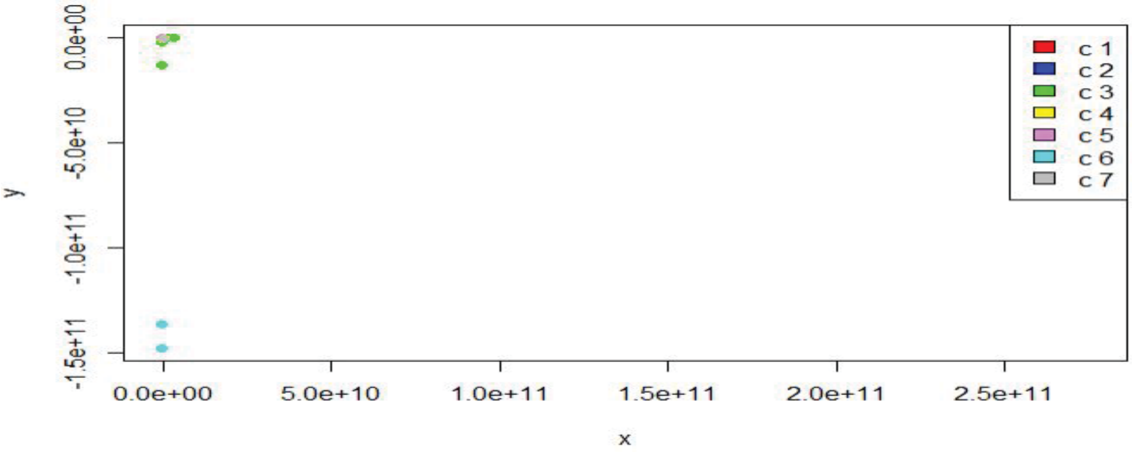

Figure 72: MDS Technique Using NMS technique on *totalFUSmall* Dataset with missing values

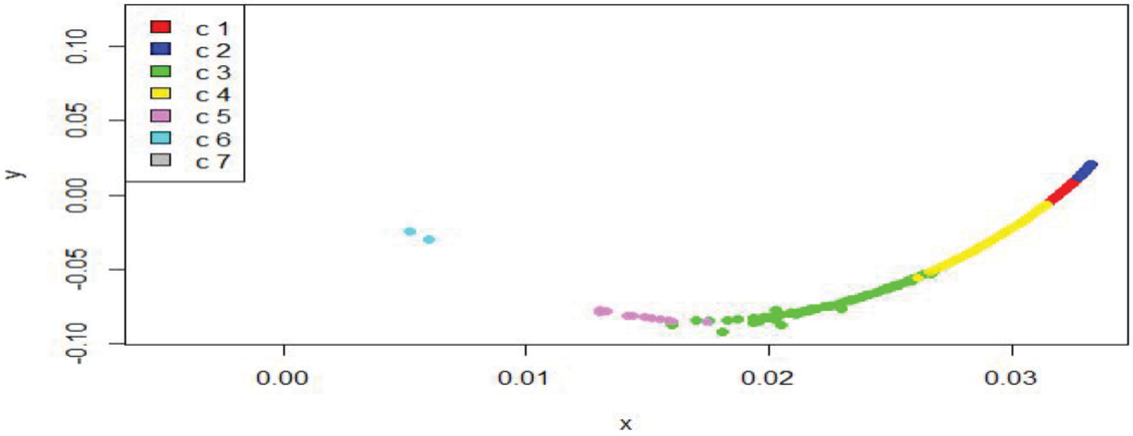

Figure 73: PCA Technique Using NMS technique on *totalFUSmall* Dataset with missing values

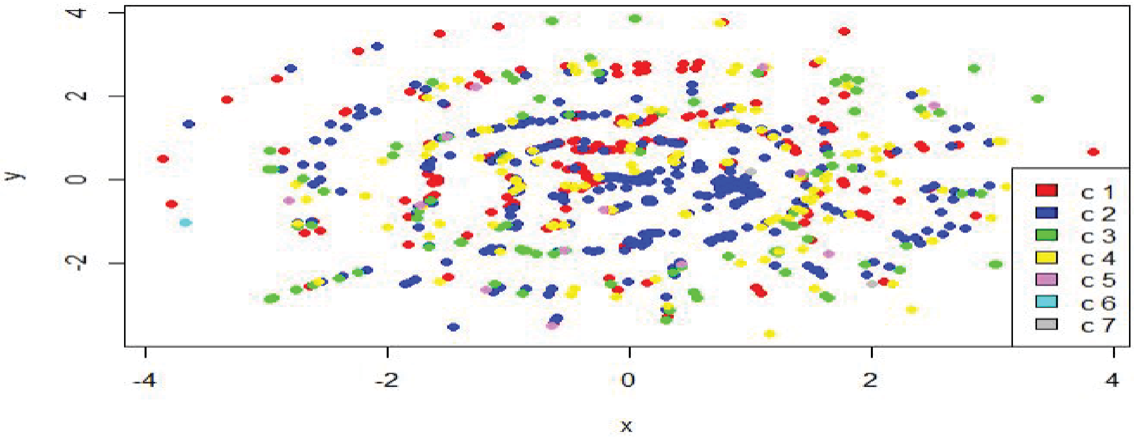

Figure 74: Rt-SNE Technique Using from Number of Optimal Clusters) on *totalFUSmall* Dataset with missing values

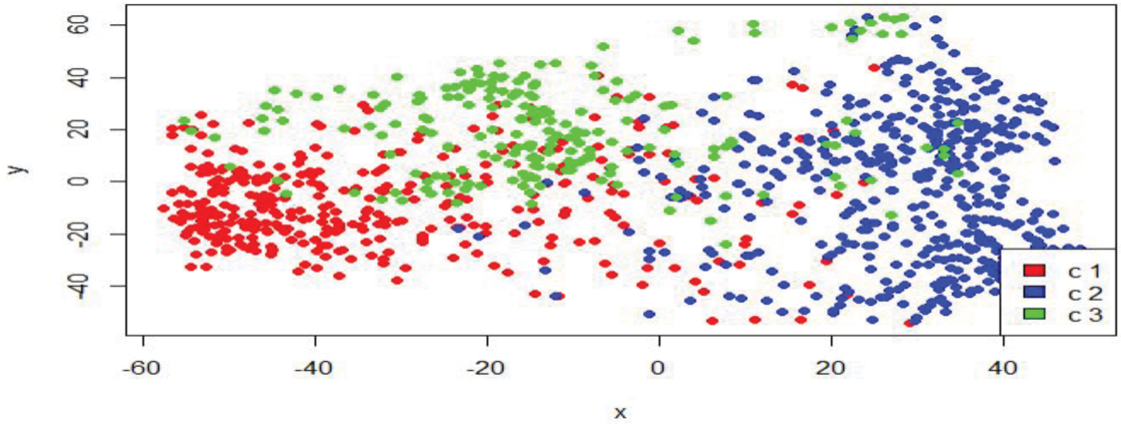

Figure 75: MDS Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with missing values

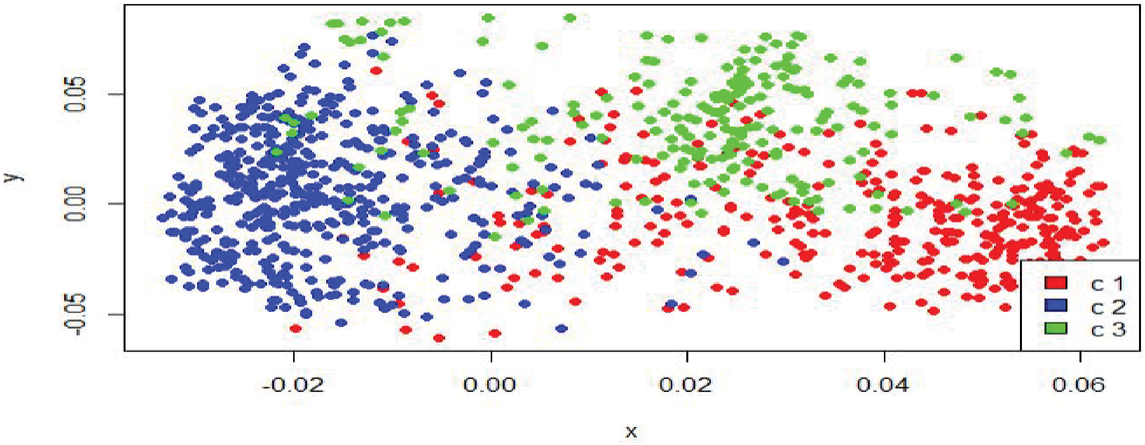

Figure 76: PCA Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with missing values

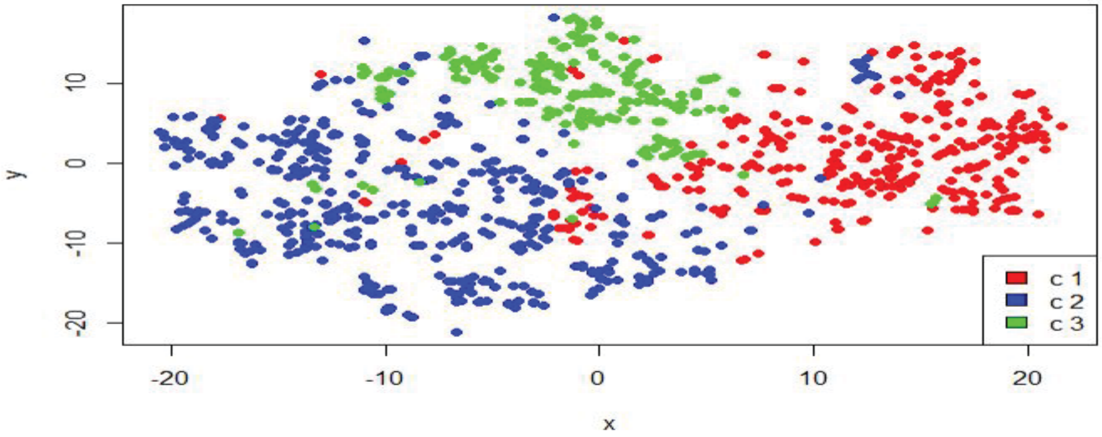

Figure 77: Rt-SNE Technique Using Time Warp Edit Distance (TWED) on *totalFUSmall* Dataset with missing values

7 Conclusion

Constant advancement in fields of science and technology is resulting in huge collection of data. To extract useful information from these data and proper visualize it, clustering methods along with number of distance measures and dimensionality reduction methods are implemented to get the most useful information out of the data. Clustering of life-course individual time series sequences helps in identifying the information more accurately.

The research question I would seek to conclude here is *"Can we Identify and extract representative sequences from categorical sequence datasets using clustering and dimensionality reduction techniques?"*. From the results we observe that optimal number of clusters is different for same dataset when different distance measures are used. This is because each distance measure implement separate number of groups that match the similarity between data points. It can be concluded that optimal number of clusters depends on the measures used for any dataset and is not constant for all the measures for the same dataset.

There is no perfect measure to capture the information in every dataset. For some datasets, tsne is able to show the good graphical representation of clusters in the data. That is harder in the case of totalFUSmall with no missing values. The future work that needs to be done is to identify representative sequences for each cluster and plot them using the t-SNE technique graphs.

8 References

[1] Andrew Abbott and John Forrest. 1986. Optimal Matching Methods for Historical Sequences. *The Journal of Interdisciplinary History* 16, 3 (1986), 471–494. https://doi.org/10.2307/204500

[2] Nicola Barban and Francesco C. Billari. 2012. Classifying life course trajectories: a comparison of latent class and sequence analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 61, 5 (2012), 765–784. http://eps.cc.ysu.edu:2059/stable/23361038

[3] P. Berkhin. 2006. A Survey of Clustering Data Mining Techniques. In *Grouping Multidimensional Data*. Springer, Berlin, Heidelberg, 25–71. https://doi.org/10.1007/3-540-28349-8_2

[4] V Umadevi CHEZHIAN, Jeya CELIN, and S GEETHA. 2011. Hierarchical Sequence Clustering Algorithm for Data Mining. 5, 1 (2011), 5.

[5] Stephan Dlugosz. 2011. Clustering Life Trajectories a New Divisive Hierarchical Clustering Algorithm for Discrete-Valued Discrete Time Series. *SSRN Electronic Journal* (2011). https://doi.org/10.2139/ssrn.1763815

[6] Daniel Engel, Lars Httenberger, and Bernd Hamann. 2012. *A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization.* Technical Report. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany. – pages. https://doi.org/10.4230/oasics.vluds.2011.135

[7] Ling Jin, Doris Lee, Alex Sim, Sam Borgeson, Kesheng Wu, C. Anna Spurlock, and Annika Todd. 2017. *Comparison of Clustering Techniques for Residential Energy Behavior using Smart Meter Data*. Technical Report. Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States). https://www.osti.gov/biblio/1398467

[8] A. Lazar, L. Jin, C. A. Spurlock, K. Wu, and A. Sim. 2017. Data quality challenges with missing values and mixed types in joint sequence analysis. In *2017 IEEE International Conference on Big Data (Big Data)*. 2620–2627. https://doi.org/10.1109/BigData.2017.8258222

[9] Laurent Lesnard. 2006. Optimal Matching and Social Sciences. (Jan. 2006). https://halshs.archives-ouvertes.fr/halshs-00008122/document

[10] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[11] Duncan McVicar and Michael Anyadike-Danes. 2002. Predicting successful and unsuccessful transitions from school to work by using sequence methods. *Journal of the Royal Statistical Society Series A* 165, 2 (2002), 317–334. https://ideas.repec.org/a/bla/jorssa/v165y2002i2p317-334.html

[12] Tadeusz Morzy, Marek Wojciechowski, and Maciej Zakrzewicz. [n. d.]. *Scalable Hierarchical Clustering Method for Sequences of Categorical Values*. https://doi.org/10.1007/3-540-45357-1_31

[13] Nicolas Severin Mueller, Matthias Studer, and Gilbert Ritschard. 2007. Clas-

sification de parcours de vie a de optimal matching. (2007), 157–160. https://archive-ouverte.unige.ch/unige:4532?gathStatIcon=true

[14] S R Pande, S S Sambare, and V M Thakre. 2012. Data Clustering Using Data Mining Techniques. 1, 8 (2012), 6. https://pdfs.semanticscholar.org/df96/a0d483bee2bd5b224698bdc7faa5f95d32db.pdf

[15] Matthias Studer. 2013. WeightedCluster Library Manual: A practical guide to creating typologies of trajectories in the social sciences with R. (2013). https://archive-ouverte.unige.ch/unige:78576

[16] Matthias Studer and Gilbert Ritschard. 2016. What matters in differences between life trajectories: a comparative review of sequence dissimilarity measures. *J. R. Stat. Soc. A* 179, 2 (Feb. 2016), 481–511. https://doi.org/10.1111/rssa.12125

[17] Laurens Van Der Maaten. 2014. Accelerating t-SNE Using Tree-based Algorithms. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 3221–3245. http://dl.acm.org/citation.cfm?id=2627435.2697068

[18] Juan Zuluaga. 2013. Optimal Matching Distances between Categorical Sequences: Distortion and Inferences by Permutation. *Culminating Projects in Applied Statistics* (Dec. 2013). http://repository.stcloudstate.edu/stat_etds/8