MULTI-MODAL CONSENSUS CLUSTERING TO IDENTIFY PHENOTYPES OF KIDNEY TRANSPLANT DONORS AND RECIPIENTS AND THEIR ASSOCIATION WITH SURVIVAL

by

Kranthi Kiran Jalakam

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
November 2022

To my mom and dad,

Thank you for everything. I wish you could've witnessed this.

Hope I've made you proud!

Table of Contents

List of Tables

List of Figures

Abstract

Kidney transplantation is an essential treatment option for individuals diagnosed with End-stage renal disease (ESRD). Being able to predict the survival of the transplant and the outcome of the recipient is an important decision point at the time of kidney allocation. Understanding the underlying characteristics of donors and recipients—referred to as phenotypes—can help in matching donors and recipients to improve patient and allograft survival. In this thesis, we are studying clustering methods to identify clusters of homogeneous donors and recipients with respect to their clinical characteristics, and using the generated phenotypes to study their relationship with kidney transplant outcomes. The dataset is a combination of both categorical and numerical data, consisting of 25824 records of donor and recipient features spanning 3 years (2009 - 2011). We investigated multi-modal clustering methods to handle the mixed data types. Two base clustering methods, KAMILA and Mixture Model, were applied resulting in 3 clusters. Consensus clustering was next applied using three consensus functions, k-modes, Majority Voting and Latent Class Analysis (LCA), to generate the final consensus-driven clusters. Latent Class Analysis (LCA) gave us the best clusters on the basis of internal evaluation indices and t-SNE visualizations. Self-Organizing Maps (SOMs) with hierarchical clustering was applied to validate the consensus clustering results. The generated clusters were evaluated by domain experts for clinical utility and each of the phenotypes. Importantly the clusters showed strong and differential associations with transplant outcomes. Some non-outcome attributes were also separately distributed across clusters.

List of Abbreviations and Symbols Used

ESRD	End-stage renal disease
ML	Machine Learning
MICE	Multivariate Imputation by Chained Equations
KS	Kolmogorov-Smirnov
SOMs	Self-Organizing Maps
LCA	Latent Class Analysis
KAMILA	KAy-means for Mixed Large Data
t-SNE	t-Stochastic Neighbor Embedding
BMI	Body-Mass Index
s.d	Standard Deviation
pmm	Predictive Mean Matching
BIC	Bayesian Information Criterion
ICL	Integrated Completed Likelihood

Acknowledgements

I would like to express my sincere gratitude to my brother, Srinivas Jalakam, and my partner in life, Urvashi Kishnani. I would not have been able to achieve this without their constant strength and support. I would like to thank my parents, Madhu Jalakam and Venkateswara Rao Jalakam, for getting me this far in life. I only wish they could be here to witness this. I owe everything to them.

I would also like to show my appreciation for my supervisor, Dr. Syed Sibte Raza Abidi, who was very supportive, understanding and accommodating throughout this journey. His valuable insights, suggestions and experience proved extremely helpful towards this work. If not for his assistance financially and personally, I would not have gotten to this point. A special thanks to Dr. Karthik Tennankore and Dr. Amanda Vinson for their expert opinion and assistance at all stages of my research. I thank all the members of the NICHE lab, particularly Syed Asil Ali Naqvi, for welcoming me and answering any questions/concerns that I had with regard to my work.

I am also really grateful to all of my friends for their everlasting support and confidence in me. They have stood by me the entire way it has taken me to accomplish this work. Specifically, Huzaifa, Arsalan, Arwa and Gaurav who have always had my back from halfway across the world!

All of their collective support, wishes and appreciation have allowed me to get here.

CHAPTER 1 INTRODUCTION

1.1 Motivation

Kidneys are an essential organ in human beings that are tasked with excretion, metabolism and endocrine functions [1]. When there is an irreversible decline in an individual's kidney function that is serious enough to be potentially fatal, that is termed End-stage renal disease (ESRD). There are a lot of factors that could increase the risks of ESRD, and ESRD in turn can lead to fluid retention, increased chances of cardiovascular disease and anaemia, among other issues [2].

For the individuals with ESRD, kidney transplantation is the most economical and preferred method of renal replacement therapy [3]. Kidney transplants are capable of providing high-quality life years to those suffering from ESRD. Transplantation is more beneficial for long-term survival in comparison with maintenance dialysis [4]. With regards to that comparison, kidney transplants could also lead to reductions in risk of mortality and cardiovascular events as well as quality of life improvements when compared to dialysis [5]. However, transplant recipients face a number of problems such as immunologic rejection and adverse effects of immunosuppressant agents [3]. Nonetheless, acknowledging that transplant is superior to dialysis in most scenarios, prolonging graft survival and reducing outcomes following transplantation have been key areas of focus for several studies up until this point. There are many donor and recipient characteristics that are known to play a role in the outcomes of kidney transplantation [6]. Being able to understand the complex relationships that may exist among these characteristics and the outcomes can play a crucial role towards advancing patient care, kidney allocations for transplants and survival prediction.

Machine Learning (ML) algorithms are an efficient tool to predict events as well as understand patterns and relationships leading to them. Cluster analysis (primarily unsupervised learning) has the potential to partition available data into groups based on their characteristic features. This provides us with a deeper understanding or interpretation of how certain outcomes are more likely to occur within a group and the potential associations between features of a group. Clustering algorithms have been employed

across various industries like healthcare, e-commerce and banking. In relation to kidney transplantation, cluster analysis can be used to recognize patterns within the data and thus potentially contribute toward improving outcomes and our understanding of the underlying features and their associations.

Clustering has been shown to be an effective approach towards phenotyping in several previous works. Soler at al. employ hierarchical clustering in their work for identifying phenotypic subgroups among patients with chronic rhinosinusitis (CRS). They mention inadequacies with the existing classification methodology and use clustering to produce a more appropriate classification for CRS patients [7]. Asthma phenotypes were identified using clustering analysis in a work by Moore et al. They used data from a research program cohort consisting of individuals with asthma and found five distinct clusters with differences in clinical, physiologic and inflammatory parameters [8]. Cluster analysis was used to identify phenotypes of chronic heart failure using patients' clinical data in the study by Ahmad et al. Their results also demonstrate the heterogeneity that exists among the patients that current measures might not recognize [9]. In an attempt to further the understanding of the pathophysiology of acute heart failure and contribute to decision-making, Horiuchi et al. used k-means clustering to identify phenotypes of acute heart failure. They obtained three clusters with differing clinical feature observations [10]. Bailly et al. identified eight phenotypes of obstructive sleep apnoea using data from over 20000 patients with the help of the Latent Class Analysis (LCA) clustering technique. Among the clusters obtained, four of them were gender-based [11].

1.2 Research Objectives

The primary objective of this study is to identify phenotypes among kidney transplant donors and recipients on the basis of their clinical characteristics. This can help provide a deeper understanding of possible associations existing among individuals' characteristics and outcomes to facilitate better event prediction and patient care. With the help of ML methods of unsupervised learning, we employ cluster analysis to generate phenotypes of kidney donors and recipients and then identify their association with patient/graft survival. An ensemble or consensus clustering approach, combining multiple clustering algorithms,

is investigated for handling mixed (numerical and categorical) data types. The research centers around obtaining meaningful and valuable groups or phenotypes rather than solely comparing performances among clustering algorithms.

This work is set to answer the following research questions:

a) Can we identify distinct phenotypes among kidney transplant donors and recipients based on their clinical characteristics with unsupervised cluster methods?
b) Can ensemble clustering facilitate better groupings among kidney transplant data than individual clustering methods?

1.3 Solution Approach

Our dataset comprises of mixed data types - i.e. categorical and numerical attributes - and there are no class labels to support cluster evaluation. To address these challenges, our solution approach is to (a) investigate clustering approaches that can simultaneously handle multiple data types, and (b) use expert validation to interpret the clusters in order to generate phenotypes. For clustering, we applied model-based clustering and cluster ensemble methods to generate consensus clusters. Additionally, we employed Self-Organizing Maps (SOMs) as an alternate approach to validate our consensus clustering results. This thesis details an approach to the task of performing cluster analysis when the data is of a mixed nature in the absence of class labels while attaining valuable results and insights in the domain of kidney transplantation.

1.4 Contributions

The contributions of this work are encapsulated as follows:

- Experimentation and application of clustering algorithms capable of handling mixed-type features to generate valuable results in the absence of a ground truth or labels.

- Employing cluster ensemble methods to produce meaningful clusters among kidney transplant donors and recipients based on their clinical characteristics. Additionally, supporting those results through the alternative method of Self-Organizing Maps.

This work can potentially assist nephrologists and kidney transplant researchers in their understanding of clinical characteristics and outcome associations to potentially improve optimal kidney allocation. On a broader scale, it could contribute to the area of mixed data type clustering which has not been as extensively researched as clustering typically involves only numeric or categorical data.

1.5 Organization of the Thesis

The thesis is structured into five primary chapters. The first is the introduction to the work, research questions and the motivation behind it. The second chapter consists of a detailed literature review in addition to the background behind the methods and procedures involved. This chapter also has a brief discussion of the prior work done in the area of kidney transplantation involving ML methods.

The third chapter is focused on the methodology used in the work for data preprocessing, imputation, clustering, consensus clustering, self-organizing maps and finally their visualization and evaluation.

The fourth chapter details the results obtained from the various clustering algorithms and approaches used and compares them using different metrics and visualizations. Similarity or agreement between clusters produced by the different methods is shown. Overall cluster descriptions and variable distributions based on the results obtained are presented here.

The fifth and final chapter summarizes the entire work involved, its contributions and the limitations encountered in the process. Future work that can be explored is highlighted in this chapter.

CHAPTER 2 BACKGROUND

In this section, we give a brief review of some of the prior work done in the area of ML for kidney transplantation. Next, we discuss data imputation methods as applied in our work. We provide a detailed description of the clustering algorithms, cluster evaluation metrics and cluster visualization methods used in our work.

2.1 ML in Kidney Transplantation

Several previous studies in kidney transplantation involving clustering methods are focused on either donors or recipients of a certain group or even a certain demographic. Joshi et al. discussed a method of clustering in order to better understand renal function loss following transplantation. The data they worked with consisted of the glomerular filtration rate (eGFR) of individuals taken over a two-year time period. Two clusters were generated in their work from that data [12]. In the work by Vaulet et al., histologic data of biopsies from kidney transplant recipients were used to perform semi-supervised clustering in order to attain phenotypes of kidney transplant rejection, using consensus clustering with the k-means method. Their goal was to develop a clinically relevant phenotypic reclassification of renal transplant rejection that also improved the prediction of subsequent graft failure above and beyond existing approaches. In their study the standard of care (the Banff pathological classification) was compared with their method to show its effectiveness [13]. In another study, consensus clustering techniques were used to identify phenotypes among Black kidney transplant recipients from a cohort covering five years. Thongprayoon et al. applied a consensus clustering approach based on multiple runs of the k-means clustering algorithm (with Euclidean distance) [14]. Consensus clustering was also used to identify clusters among a population of morbidly obese kidney transplant recipients and to analyze outcomes [15]. Gangopadhyay et al. employed spectral clustering algorithms on time-series data of 24 months following 111 patients' transplants in an effort to study renal function following kidney transplantation and identify intervention strategies [16]. In order to better understand changes in health-related quality of life (HRQOL) post-kidney transplantation, Villeneuve et al. used k-means clustering

specifically designed for longitudinal data and identified two clusters of time profiles. They used data from patients at different points in time spanning a 36-month period. Differences among clusters in terms of the HRQOL were described. Additionally, Random Forests were used to study the association between the covariates involved and the clusters obtained [17].

Beyond clustering, ML-based classification algorithms have been used to predict the risk of graft failure following transplant across three temporal cohorts in the work by Naqvi et al. [18]. Paquette et al. used various survival analysis models that also included artificial neural networks in an attempt to predict transplant outcomes for donor-recipient pairs [19]. Yoo et al. created new prediction models of graft survival in order to predict long-term graft survival in kidney transplant recipients using data from a Korean population [20]. Decruyenaere et al. employed multiple ML models to predict delayed graft function in recipients post-renal transplantation. They compared the performances of 9 different models and used a reduced form of a dataset consisting of features that were potential risk factors for delayed graft function and found that Linear Support Vector Machines (SVMs) had the best model performance [21]. An ensemble of Random Survival Forests and Cox proportional hazard model was used to predict kidney transplant survival in the study by Mark et al.. They split the data into two age-dependent cohorts and applied one of the two models to each of them and finally combined the predictions from both of them. Variable selection was done based on importance. The performance of the proposed method was compared with other existing kidney transplant survival models [22]. Four different ML models were applied to data from kidney transplant recipients in the work of Peng et al.. This was done in order to study the association between immune monitoring and pneumonia, said to be one of the major complications post-surgery. A comparison of the methods in their ability to evaluate pneumonia risk was reported and characteristic differences among pneumonia patients and those without penumonia were highlighted [23].

The work presented in this thesis involves data spanning a broad population involving both donor and recipient characteristics using consensus clustering built on model-based

clustering algorithms. To our knowledge, we have not seen previous work that has dealt with model-based clustering methods in the area of kidney transplantation.

2.2 Multivariate Imputation by Chained Equations

Multivariate Imputation by Chained Equations (MICE) is a popular data imputation approach. It can be used to impute both categorical and numerical data simultaneously, which other imputation methods such as KNNimputer [24][25], IterativeImputer [24][26] are unable to achieve.

MICE enables us to use regression methods appropriate for categorical data like logistic regression and linear regression for numerical data simultaneously [27].

Also known as fully conditional specification (FCS), MICE allows us to specify an imputation model on a variable-by-variable basis by a set of conditional densities, one for each of the existing variables with incomplete values [27]. Another well-known multiple imputation approach is joint modeling (JM); however, JM utilizes linear regression equations and does not work well for categorical data imputation due to its assumption of normality and linearity [28]–[30].

Yang et al. used fully conditional specification for multiple imputation for an epidemiologic study evaluating national blood utilization patterns in Namibia [30]. Their study specified the importance of using imputation in datasets with a substantial amount of missing information instead of working on complete cases only, which can introduce a loss of information and bias. Complete case analysis is a widely used approach to deal with missing data where any record with missing variables is excluded from the analysis, which could be appropriate when we have less than 5% missing data. Multiple imputation has the added advantage of working with more relaxed assumptions [30], [31]. Robert et al. applied multiple imputation by chained equations to impute missing values in their data associated with smoking cessation treatment [32]. MICE was employed in a study that compared initial growth, virological and immunological responses of HIV-infected children in the United Kingdom/Ireland and Kampala, Uganda to antiretroviral therapy

(ART) [33]. Thongprayoon et al. used MICE in their work on clustering to identify groupings among black kidney transplant recipients [14].

The procedure involved in MICE is briefly described below.

Let the hypothetically complete data Y be a partially observed random sample from the p-variate multivariate distribution $P(Y|\theta)$. θ is a vector of unknown parameters that is assumed to completely specify the multivariate distribution of Y. $t = 1,2,\ldots,m$ denote the variables in Y. Y_t is the t^{th} variable and Y_{-t} denotes all the other variables in Y besides Y_t. MICE attains the posterior distribution of θ by sampling iteratively from conditional distributions of the form

$$P(Y_1|Y_{-1},\theta_1) \ldots\ldots\ldots P(Y_m|Y_{-m},\theta_m)$$

$\theta_1 \cdot \cdots \cdots \theta_m$ are associated with the respective conditional densities. Beginning at a draw from the observed marginal distributions, the n^{th} iteration of chained equations is a Gibbs sampler that successively draws

$$\theta_1^{*(n)} \sim P(\theta_1|Y_1^{obs}, Y_2^{(n-1)}, \ldots., Y_m^{(n-1)})$$

$$Y_1^{*(n)} \sim P(Y_1|Y_1^{obs}, Y_2^{(n-1)}, \ldots., Y_m^{(n-1)}, \theta_1^{*(n)})$$

$$\ldots$$

$$\theta_m^{*(n)} \sim P(\theta_m|Y_t^{obs}, Y_1^{(n)}, \ldots., Y_{m-1}^{(n)})$$

$$Y_m^{*(n)} \sim P(Y_m|Y_m^{obs}, Y_1^{(n)}, \ldots., Y_m^{(n)}, \theta_m^{*(n)})$$

where $Y_t^{(n)} = \left(Y_t^{obs}, Y_t^{*(n)}\right)$ is the t^{th} imputed variable in the nth iteration [27].

In our work, we have used only one of five different imputations generated. This is further elaborated in section 3.2.3. By using only one imputation, we were aware that we were not taking the possible uncertainty associated with the imputation into consideration. However, most of the prior studies and documentation spoke to regression analysis and used standard error when analyzing the datasets since those datasets consist of only numerical variables. One study even mentioned a value 'CritCF' that would be applicable for cluster analysis, but that criterion only works with numerical data [34]. For that reason,

in our study, we employed a Kolmogorov-Smirnov Test to test the quality of imputation. An identical approach was previously described in the work by Liu and De and was a method we found to be useful to determine if the imputations were reasonable [30].

2.3 Clustering Algorithms

Clustering is a popular ML method that can help us break down datasets and understand the detailed, complex and sometimes unexpected interactions that may exist in them while also being able to group the data into clusters. The aim is to find clusters where data points are most similar within the cluster and different from points in other clusters. This approach has been widely popular in a wide range of fields and applications like medicine, market segmentation and data analysis. Most commonly, it is a form of unsupervised ML since we are trying to find clusters among data not knowing their labels or groups that may already exist. In this work, clustering enables us to identify phenotypes among kidney transplant donors and recipients based on their clinical characteristics.

One of the most common clustering algorithms is k-means which is a simple yet effective algorithm to generate clusters [35]. However, k-means generally requires the data to be numerical. k-means is a distance-based partitional clustering algorithm [36]. Huang developed the k-prototype clustering algorithm, like the k-means algorithm, but it could work with mixed data type variables [37], [38]. However, it had a few shortcomings like several other partitional clustering algorithms or other algorithms that relied on the distance calculation/matrices for their cluster generation [39]. For one, the computational power and large memory required for the calculation of some of the distance matrices are a limitation. Some of these distance matrices have a complexity of $O(n^2)$ that are severely impacted by the size of the datasets. In our work, we decided to employ model-based clustering methods that can handle the data in their original state without any encoding or additional processing. Several studies have shown the promising results yielded when using model-based clustering algorithms, even with mixed datasets. These algorithms work off the assumption that the data points match a statistical distribution [39], [40].

Preud'homme et al. did a comparison of the results obtained using various distance-based and model-based clustering algorithms on a heterogenous clinical trial data and found the model-based methods to generally perform better [41]. Hunt and Jorgensen used mixed model clustering on a large mixed variable medical dataset and mentioned some of the issues with traditional clustering algorithms. One of them is that any randomness in the sample is not reflected and small fluctuations in the data can lead to considerably varying clusters being generated [42]. Storlie et al. performed clustering with a model-based method called the Dirichlet process model, on a group of individuals with potential autism spectrum disorder [43].

We have used two model-based clustering algorithms in our work, mixture model and KAMILA which are discussed below in Section 2.3.1 and Section 2.3.2 respectively.

2.3.1 Mixture Model

Mixture models are an effective way for us to cluster data consisting of mixed data type variables as it allows us to use gaussian mixtures for numerical data and multivariate latent class models for categorical data [44]. Depending on the problem, various algorithms are available to estimate the mixture parameters, particularly Expectation Maximization (EM), Stochastic Expectation Maximization and Classification Expectation Maximization [45].

In the case of gaussian models, each point x_i is assumed to arise independently from a mixture of d-dimensional Gaussian density having mean μ_k and variance matrix Σ_k.

The eigen value decomposition of Σ_k yields

$$\Sigma_k = \lambda_k D_k A_k D_k'$$

$\lambda_k = |\Sigma_k|^{1/d}$ and D_k is the matrix of eigenvectors of Σ_k. A_k is the diagonal matrix with $|A_k| = 1$. $\lambda_k, D_k \; and \; A_k$ define the volume, orientation and shape respectively of the cluster k. By varying these three quantities $\lambda_k, D_k \; and \; A_k$, different gaussian models are generated. The Bayesian information criterion (BIC), integrated completed likelihood (ICL) and normalized entropy criterion (NEC) can be used to determine which gaussian model is suitable for the task.

For categorical variables, each point x_i arises independently from a mixture of multivariate multinomial distributions. This latent class model assumes that the categorical variables are independent given the latent variable. Similar to gaussian mixture models, the multinomial distribution associated with the t^{th} variable of the k^{th} component is reparametrized by a center a_k^t and dispersion ε_k^t around this center. Different models are obtained depending on whether ε_k^t is independent of variable t or component k or both [44], [45].

The idea behind mixture models is to attain a statistical formulation of the data in the form of a model whose parameters are estimated using a maximum likelihood algorithm. This is used to identify group membership conditional probabilities of the samples involved. On convergence, clusters can be identified using those probabilities. Hunt and Jorgensen discuss how finite mixture models could be useful even in the absence of a natural cluster structure among the data. In their study, they demonstrated the use of mixture model clustering on two mixed datasets, a heart disease dataset and a class examination dataset [46].

Mixture model-based clustering was used to cluster prostate cancer patients who were diagnosed with stage 3 or 4 prostate cancer by McParland and Gormley. The dataset consisted of both numerical and categorical variables. They used the Bayesian information criterion (BIC) to determine the model to be used for their data and found 3 clusters to be ideal. They also applied their method to a mixed data type simulated dataset and attained good results [47]. Hunt and Jorgensen applied mixture models to identify clusters that are closely related to the pre-existing structure within the same prostate cancer dataset [48], [49]. However, their method involved clustering in the presence of incomplete data [50]. A Gaussian copulas-based mixture model is proposed in the work by Marbac et al.. It is capable of working with data that are of continuous, integer or ordinal types. The proposed method's effectiveness is evaluated on two real datasets and the BIC and ICL criteria are used for model selection in the process [51]. A multivariate Gaussian mixture model is presented in a work by Morlini that is capable of clustering datasets consisting of binary and numerical variables. Its effectiveness is shown with multiple datasets (simulated and

real) while the models are compared using the Akaike information criterion (AIC) and Bayesian information criterion (BIC) [52].

2.3.2 KAMILA

Kay-means for Mixed Large data (KAMILA) is a clustering method developed by Foss et al. to be able to cluster data containing quantitative and qualitative variables without having to make strong parametric assumptions and is also efficient with larger datasets. They identified that current clustering methods tended to make strong parametric assumptions to properly weigh the contribution from quantitative and qualitative variables and wanted to counter this problem. KAMILA is a semi-parametric generalization of k-means that is able to balance the contributions from both types of variables without having to additionally specify any weights for them [53]. Quantitative variables are modeled with a general class of elliptical distributions while qualitative variables are modeled with mixtures of multinomial random variables [54]. Similar to the mixture models mentioned above, parameters here are estimated by a process similar to expectation maximization [53], [54]. KAMILA showed great results when compared with several other clustering algorithms that were applied to a clinical trial dataset in the work by Preud'homme et al. [41]. In the benchmark study of Jimeno et al., they evaluated the performance of KAMILA along with other algorithms in 27 different simulated scenarios consisting of different cluster numbers, overlaps and number of numerical variables and found KAMILA to consistently perform well [55].

2.4 Cluster Ensemble

Cluster ensemble is a method of combining solutions from various clusterings of the same dataset in order to produce a better quality and more robust solution than the individual or base clusterings. The consensus clusters that result from these cluster ensembles are obtained based on different generation mechanisms and consensus functions [56].

Performances of algorithms may vary widely depending on the dataset being used. For these ensembles, we first need to decide on the method of cluster generation. This can come from varying the number of runs of the algorithm, differing initialization parameters (like in k-means) or as in our case, algorithms that partition the data by different approaches. Our method of cluster generation allows us to use the results from diverse clustering algorithms that vary in their approach to clustering the data and integrate it into a single result. Following the generation phase, we can obtain a consensus among these "base clusterings" through several different available strategies [57], [58].

Shen et al. generated a cluster ensemble using k-means, a hierarchical clustering method and an expectation maximization (EM) based method. Stability and fitness were used to validate the best solutions among the base clustering methods since they were generated by varying the number of clusters. Six best solutions were obtained after evaluation. They proceeded to use k-modes as the consensus function to generate a single consensus result from these initial clusterings. This method was applied to a dataset consisting of patients with Pervasive Development Disorders (PDD) [59]. Alternatively, in the study by Iam-On et al., a cluster ensemble was generated by using different initialization parameters on the same k-prototypes clustering algorithm. They have presented a link-based approach to aggregate the clustering results and produce the final consensus solution. Results from the use of this ensemble approach on two different biological datasets are discussed [60].

Greene et al. employ k-means, k-medoids and a fast "weak clustering" algorithm to generate the clusterings for the ensemble. They obtain a consensus from these clusterings using hierarchical clustering algorithms with different linkage options. Results from consensus clustering on benchmark medical datasets as well as synthetic datasets are presented [57], [61].

In our work, the base clusters are produced by the two clustering algorithms - i.e. mixture model and KAMILA. Three different consensus functions are tested and applied to the base clustering outcomes to generate the final consensus clusters - i.e. k-modes, majority voting and Latent Class Analysis (LCA).

2.4.1 k-modes

k-means is an effective clustering algorithm traditionally used for numerical data. k-modes is a method of clustering datasets that consists of categorical variables and was developed to be an extension of k-means by Huang. As a result, k-modes also works as a suitable method of consensus clustering using the resulting clusters (with their labels) produced by the two individual model-based clustering algorithms. It primarily differs from k-means in the type of dissimilarity being used, using modes instead of means and following a frequency-based approach in order to update those modes [62].

The dissimilarity is given by

$$d(X,Y) = \sum_{t=1}^{m} \delta(x_t, y_t)$$

where X and Y are two objects between which we are trying to find the dissimilarity and x_t and y_t are their respective categorical variable values for the variable t.

$$\delta(x_t, y_t) = \begin{cases} 0 & (x_t = y_t) \\ 1 & (x_t \neq y_t) \end{cases}$$

The mode of X where X is a set of categorical objects having attributes $(A_1, A_2, \ldots, A_m)$ is given by $Q = [q_1, q_2, \ldots, q_m] \in \Omega$ that minimizes

$$D(Q,X) = \sum_{i=1}^{n} d(X_i, Q)$$

Ω is the overall space consisting of $A_1, A_2, \ldots, A_m$ attributes.

As with k-means, we first select initial modes for each cluster, assign samples to those clusters depending on the shortest d from the mode, and constantly update the mode of the cluster after each assignment. We then check the dissimilarity of the samples again with the new modes and reassign any samples that might be closer to another mode. We then calculate the new modes and repeat the process of assignment of samples and dissimilarity

calculation to newly generated modes until there are no more changes occurring in the assignment [38], [62].

The success of this algorithm on a popular soybean disease dataset is shown as well. This method also has the added advantage of working with very large datasets, just like with k-means [38], [62]. As mentioned in Section 2.4, k-modes was used as the consensus function in the study by Shen et al. [59]. Luo et al. demonstrate how k-modes can be used as a consensus function for obtaining final partitions based on labels from multiple runs of the k-means algorithm. Comparisons with five other consensus functions using a popular benchmark dataset are presented in their work [63].

2.4.2 Majority Voting

Ayad and Kamel developed a voting-based consensus clustering method that directly dealt with the voting problem. The voting problem is that of estimating sample assignments to the reference cluster partitions on the basis of their assignments to the base cluster partitions that make up the ensemble, so as to minimize the estimation errors with the representative ensemble partition. They deal with the voting problem using a multiple regression approach where they treat the representative cluster partitions as the outcome variables and the base clustering partitions as the input variables. The final consensus partition is regarded as a soft partition which means that it is generated by averaging the probabilities of cluster-label assignment. The objective is to find the optimal compression of the statistical distribution to retain the maximum amount of information [64]. Two algorithms, one that follows a bipartite matching scheme or bVote and an adaptive cumulative voting scheme or Ada-cVote are presented by Ayad and Kamel. The results of their consensus clustering approach on two artificial datasets and three real-world datasets are presented and compared with other existing algorithms [64]. Vaulet et al. use majority voting to obtain a consensus solution with multiple k-means algorithm runs (with different subsamples and initializations) in their work to identify phenotypes for acute renal transplant rejection [13].

2.4.3 Latent Class Analysis (LCA)

This is a technique useful for multivariate categorical data and hence works as a method of consensus clustering based on clusters produced by individual clustering algorithms. “The latent class model aims to stratify the cross-classification table of observed (or “manifest”) variables by an unobserved (“latent”) unordered categorical variable that eliminates all confounding between the manifest variables” [65]. Essentially, the model tries to group each sample into a latent class in a probabilistic manner. The observed variables are sampled from a mixture of multinomial distributions, which are associated with the cluster that the sample is a part of [41]. It works similarly to the mixture model mentioned previously. Ferreira et al. use latent class analysis to identify groupings in a study among patients with heart failures in order to be able to understand the outcomes and responses to different treatments [66]. As with the other model-based clustering methods, this is probabilistic in nature which implies that a sample is believed to be part of one cluster but the uncertainty about an object’s class membership is taken into consideration as well. This method does share some similarities with fuzzy clustering approaches. Vermunt and Magdison show the clustering performed by Latent Class on a diabetes dataset of numerical variables and a Prostate cancer dataset with mixed variables. In the case of the prostate cancer dataset, they wanted to identify clusters that differed based on the likelihood of success from treatment [67]. LCA was used as a clustering method in the work by Bailly et al. towards phenotyping sleep apnoea using a large dataset of over 20,000 patients [11]. As shown later in our results, this consensus method provided us with the best solution.

2.5 Self-Organizing Maps (SOMs)

Kohonen developed an Artificial Neural Network (ANN) method for the visualization of high-dimension data that presents the information on a low-dimensional grid. This implicitly groups samples together in the process which can be further observed in a data clustering manner. SOMs manage to retain the topological and metric relationships that may exist between the observations when obtaining that lower-dimensional representation.

In its generation, SOMs produce a two-dimensional grid of nodes where each node is linked to a model of some sample. These models are grouped closer together when they are more similar than the ones that are not. This is done in a nonparametric, recursive regression method [68]. Even though SOMs are able to group similar observations closer together, by applying a clustering method like partitioning around medoids (pam) or a hierarchical clustering algorithm on the output from the SOM, we can obtain clearly distinguishable clusters. Originally, SOMs generate prototype vectors that represent the main dataset which are then combined with some clustering algorithm such as hierarchical clustering as used in our work. Every sample of the dataset is a part of the same cluster as its nearest prototype. Vesanto and Alhoneimi state how this this method of clustering allows for efficient utilization of clustering algorithms due to the algorithms working with the prototype vectors which are typically smaller than the dataset and could also result in identifying clusters of arbitrary shapes and sizes [69].

Ong and Abidi used SOM on a dataset from the World Bank involving social indicators as the input variables. They were able to obtain a good visualization and clustering of the observations by applying the a-dK means method to the trained SOM. In their analysis, they observed that countries with a similar geographical location were grouped closer together in the SOM representation even though the SOM was trained in an unsupervised manner where no direct indication of similar countries was provided [70]. Vesanto and Alhoneimi explore the use of an agglomerative method and k-means clustering method on the output of the Self-Organizing Map. They apply their methodology to three diverse datasets, one real-world and two artificial, to yield some interesting results and also slightly emphasized the computational efficiency of this method [69]. In the study by Kiang, they applied a contiguity-constrained clustering method to the output of the SOM that was generated. Their clustering method relied on a minimal variance criterion rather than a minimal distance criterion. Comparisons with other clustering approaches were drawn as well. They applied this to two popular datasets, the Iris and wine datasets from UCI [61], [71]. García and González describe a two-level approach to using the prototype vectors obtained from SOM and apply k-means to obtain proper clusters in their work involving wastewater treatment monitoring. They detail the SOM algorithm and discuss the various ways to represent the information from the SOM generation [72].

In our work, we have used SOM as a way to support our clustering results. This is because there are a lack of existing metrics and evaluation indices when it comes to clustering mixed data having no "ground truth" or labels [39]. The results from our work also indicate how SOMs could be used as an independent clustering approach for mixed-type data clustering and not just as a supporting method. Additionally, SOMs present a visualization method for our dataset.

2.6 Cluster Evaluation

The clustering algorithms used in our study work with mixed-type data without relying on a distance measure. However, to evaluate the algorithms using the internal evaluation indices, a distance measure needs to be computed for our dataset. The Gower's distance measure is employed to facilitate that. This distance measure is also used to generate the t-SNE visualizations of our clustering results.

2.6.1 Gower's Distance

Calculating the distance between points in a dataset is an integral part of several clustering algorithms. A great portion of cluster analysis studies tends to deal with numerical variables only, for which multiple different distance measures are available to calculate the distance between two data points. In that case, Euclidean distance, Manhattan distance, and Minkowski distance are some of the commonly used distance measures. However, it is a little more complex when dealing with a mixture of numerical and categorical data. One of the most popular distance measures for mixed data (categorical and numerical) is the Gower distance. Gower introduced the concept in 1971 as a coefficient of similarity between sampling units [73]. It is a relatively simple distance calculation involved compared to some other mixed data distance measures. Weatherall et al. use this distance measure in their cluster analysis study that identified phenotypes among airway diseases in a population. Diana and agnes were the two hierarchical clustering algorithms that used the Gower distance in their cluster formation [74]. Özlem et al. performed clustering on a mixed panel dataset using agglomerative hierarchical clustering with Gower distance since

they had to deal with numerical and categorical variables [75]. Ebbert et al. analysed students' lecture recordings to identify patterns by clustering, using the partitioning around medoids algorithm with Gower distance [76].

Let $t = 1,2,\dots,m$ be the variables in the dataset. The Gower's distance is based on the following calculation:

$$S_{ij} = \frac{\sum_{t=1}^{m} s_{ijt}\delta_{ijt}}{\sum_{t=1}^{m} \delta_{ijt}}$$

Where S_{ij} is the similarity between the two points i and j. s_{ijt} and δ_{ijt} are calculated based on whether the variable is numerical, categorical or dichotomous.

If the variable is numerical:

$s_{ijt} = 1 - \left|x_{it} - x_{jt}\right| / R_t$, where x_{it} and x_{jt} are the respective values of the points i and j for the variable t and R_t is the range of the variable t.

If the variable is categorical:

$s_{ijt} = 1$ if both i and j have the same value for the variable t.

If the variable is dichotomous:

Only when both i and j have the same value of the variable t (Presence or yes or positive etc.), the $s_{ijt} = 1$. In every other case, including when both i and j don't have a value for t (absence or negative), the $s_{ijt} = 0$. δ_{ijt} is always 1 unless there is no value for variable t for i and j.

δ_{ijt} is always 1 whenever the variable is categorical or numerical.

The Gower distance between i and j is given by $\sqrt{1 - S_{ij}}$ [73].

2.6.2 Internal Evaluation Indices

To evaluate the performances of the clustering algorithms and identify the right number of clusters for our problem, internal evaluation indices need to be utilized. Since the data is of a mixed type and has no 'ground truth' or labels, options to evaluate the different approaches' and algorithms' performances are limited. This could be due to a lack of available implementations as well as not being applicable to data of this nature. Several previous works involving mixed-type data have worked with datasets having a ground truth or labels for which performance metrics like clustering accuracy, rand index and normalized mutual information are used [39]. We have identified three indices, the Silhouette index, Dunn's index and Calinski-Harabasz scores for our work. The indices are usually used on numerical variable datasets with Euclidean distance measures. In this work, they are applied to the Gower's distance matrix that is computed for mixed-type data.

2.6.2.1 Silhouette index

The Silhouette index is a cluster validity measure that is representative of the tightness and separation of clusters. It is dependent on the actual cluster partitions rather than the algorithms that are used. It can also aid us in determining the right number of clusters (or k) [77]. In a dataset of m objects and c clusters, the Silhouette width for an object $s(i)(i = 1, \dots, m)$ is given by

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance between the i^{th} object and all of the objects in a cluster X_j $(j = 1, \dots, c)$ and $b(i)$ is the minimum average distance between the i^{th} object and all of the objects in cluster X_k $(k = 1, \dots, c; k \neq j)$. The silhouette scores shown in our work are the overall average silhouette widths that are obtained by averaging the $s(i)$ values for all the objects. The scores fall between a value of -1 and $+1$ where a value closer to the latter is representative of a good clustering and a score closer to the former is indicative of a misclassification [77], [78].

2.6.2.2 Dunn's index

Dunn's index could be used to detect compact and well-separated clusters. In a dataset with c clusters and a partition U, the index is calculated by

$$D(U) = \min_{1 \le i \le c} \{ \min_{1 \le j \le c} \left\{ \frac{\delta(X_i, X_j)}{\max_{1 \le k \le c} \{ \Delta(X_k) \}} \right\} \}$$

where $\delta(X_i, X_j)$ is the intercluster distance between the two clusters X_i and X_j; $\Delta(X_k)$ is the intracluster distance of cluster X_k and U is the overall partition that the clusters are a part of. For Dunn's index, the value lies between 0 and ∞ where a higher score represents good clusters [78].

2.6.2.3 Calinski - Harabasz index

Initially proposed by Calinski and Harabasz, this cluster validation metric was shown to be successful in the work by Milligan and Cooper [79], [80]. Hennig and Liao extended this metric with mixed-type variables using dissimilarities [81]. The Calinski-Harabasz index is given by

$$CH = \frac{\Sigma_k d^2(c_k, g)/(NC - 1)}{\Sigma_k \Sigma_{j \epsilon X_k} d^2(j, c_k)/(m - NC)}$$

where $d^2(c_k, g)$ is the squared distance between the center of the cluster k and the center of the dataset g. $d^2(j, c_k)$ is the squared distance between the point j and the center of the cluster k. m is the total number of points in the dataset and NC is the number of clusters. This metric measures both compactness and separation simultaneously. A higher score represents a better result [79], [82].

2.6.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a visualization technique for high-dimensional data that allows each sample to have a location in a two or three-dimensional map. This method manages to capture most

of the local structure present in the data while simultaneously being able to represent the global structure of the presence of clusters. Maaten and Hinton show the superiority of this method compared to other methods like Sammon mapping and Isomaps on a number of data sets. The drawbacks of Stochastic Neighbor Embedding that are resolved by t-SNE are stated in their work [83]. The t-SNE visualization is computed on the Gower distance matrix generated.

2.6.4 Statistical Tests

Statistical tests can be used to compare variable distributions among the clusters and understand significant differences and associations among clusters. We have used the non-parametric Kruskal-Wallis test for numerical variables and the chi-square test for categorical variables. Statistical tests have been previously used in phenotyping studies to compare cluster variables. Thongprayoon et al. used the ANOVA and Kruskal-Wallis tests for numerical and chi-square test for categorical variables in their study involving clustering in kidney transplantation [14]. Soler et al. used t-tests for numerical and chi-square for categorical variables in their work to identify phenotypes among patients with chronic rhinosinusitis [7]. In their study involving phenotyping in sleep apnoea patients, Bailly et al. used the Kruskal-Wallis tests for numeric variables and chi-square or Fisher's exact test for categorical variables [11].

2.7 Summary

We have performed a significantly detailed review of existing literature and methods surrounding our work in this section. Initially, existing studies involving ML in kidney transplantation are presented. We discuss Multivariate Imputation by Chained Equations (MICE) which is an effective data imputation method that is versatile in its application. This is a popular imputation approach taken by researchers in this field. It works well with datasets that consist of both categorical and numerical variables. We then talk about the concept of clustering and the methods involved in our work. Cluster analysis is a data mining task that is popularly employed across industries and applications. A lot of existing

work has involved distance-based clustering algorithms or clustering with quantitative data. Clustering with mixed-type data is not as straightforward or as popular and is a field that needs to be explored further. There are a lot of challenges that come with a mixture of variable types like imputation, integration of distance calculations for the respective types and visualizing the data in the same space. Model-based clustering methods are effective in dealing with some mixed data clustering issues and can lead to significant insights from data of such mixed nature, which a lot of real-world medical datasets are. Two clustering algorithms for the task at hand are presented. There are benefits that these algorithms provide over a more traditional approach like k-means. Obtaining a consensus among clustering results is discussed. It is an effective method to improve our results by providing better quality solutions. This consensus can be obtained in different ways as has been done in other work. The k-modes, Majority Voting and Latent Class Analysis (LCA) functions to obtain a consensus among clustering solutions are detailed in this chapter.

We have described Self-Organizing Maps (SOM) as a method of clustering to reinforce the results. This also works as a powerful visualization tool in our analysis. Previous works that used SOM to cluster observations are reviewed and reported.

We proceed to talk about the distance measure of Gower distance, which was used in our work to select algorithms that performed well for our task and implemented with cluster evaluation indices. Gower distance is a popular mixed data distance measure and is relatively simple in calculation. This distance measure has widely been used in previous works involving distance-based clustering methods. Internal evaluation metrics to evaluate clustering model performances in the absence of labels are presented, even when the data is of a mixed nature.

Finally, we have reviewed t-Distributed Stochastic Neighbor Embedding (t-SNE) which is a powerful visualization technique. It allows us to view high-dimensional data, in a two or three-dimensional form. Overall cluster structure from the various clustering methods and algorithms can be observed in this lower dimensional form.

CHAPTER 3 CLUSTERING METHODOLOGY AND EXPERIMENTS

Our clustering methodology is designed to handle mixed data types and the lack of class labels that are used for cluster evaluation (as is the case with past clustering studies). The unlabelled dataset used in this study comprises both numerical and categorical features.

The methodology that has been developed can be segmented into four phases as shown in Figure 1**.** The first step of our methodology is pre-processing the dataset which includes an elaborate imputation approach suitable for our dataset. Next, model-based clustering algorithms that work with mixed-type data are investigated to generate base clustering results. Consensus methods are applied to these base clusters to produce the final consensus clusters. Our methodology also involves the evaluation of the clustering results in the absence of ground truth or labels. Finally, based on expert validation of the clusters, kidney donor and recipient phenotypes are generated.

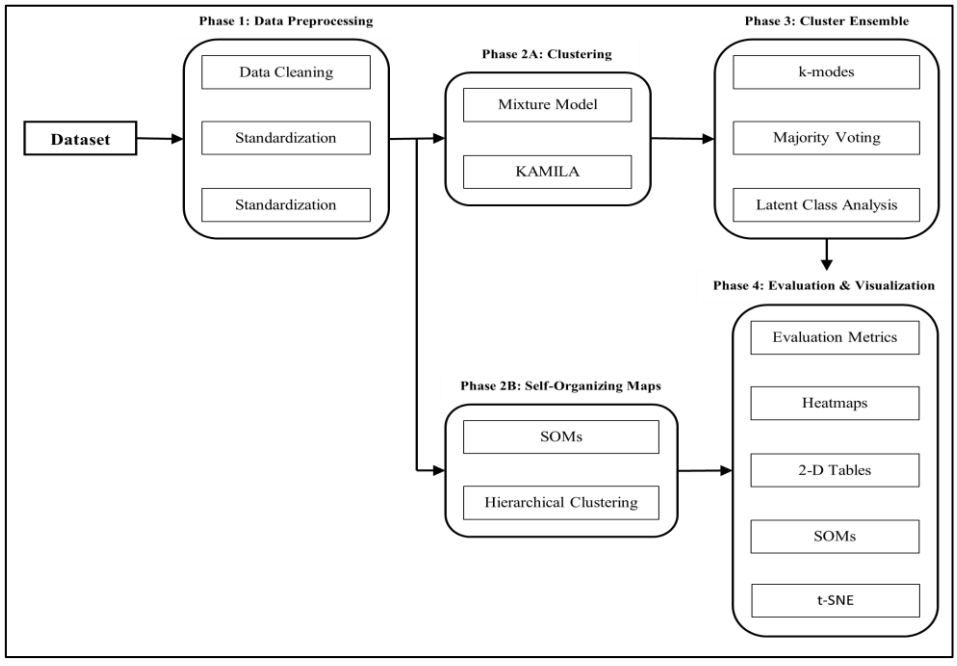

Figure 1: Overview of research methodology

The four phases involved are briefly described below and the methods involved are elaborated further in the next sections.

Phase 1 - The first phase involves the preprocessing of the dataset, including imputation, null value deletion, value deletion by criteria, and data standardization among other tasks. Recalculation / Creation of new features is a part of this phase as well.

Phase 2A and 2B - In the second phase, clustering is performed on the features obtained after preprocessing the dataset (and cohort extraction) using two different clustering algorithms to generate the base clusters. Additionally, in this phase, Self-Organizing Maps with clustering is applied to substantiate the final ensemble clustering results.

Phase 3 - In the third phase, we generate a consensus cluster from the base clusters using three popular ensemble methods: k-modes, Majority Voting and Latent Class Analysis (LCA). This allows us to obtain the final cluster partitions.

Phase 4 - In the fourth and final phase, we evaluate and visualize generated clusters. Heatmaps are used to represent the categorical variable distributions among the generated clusters. Simple tables to show differences between numerical variables among the clusters are presented. Internal evaluation indices of Silhouette index, Dunn index and Calinski- Harabasz scores for the various methods are presented. Self-Organizing Maps and t-SNE are used to represent the clusters obtained in different forms. Final cluster descriptions obtained from the solutions are described and detailed. Brief as well as elaborate summaries are presented in Section 4.5.

Our methodology is applied to a kidney transplantation dataset involving donor and recipient characteristics. It allows us to obtain clusters in that data to identify phenotypes that exist among these individuals. This provides us with a better understanding of underlying associations and outcome predictions. As mentioned previously, this could contribute to better diagnostic and treatment plans for the individuals involved.

3.1 Dataset

The dataset that is employed in this study is from the Scientific Registry of Transplant Recipients (SRTR). The SRTR data system includes data on all donors, wait-listed candidates, and transplant recipients in the United States, submitted by the members of the Organ Procurement and Transplantation Network. The Health Resources and Services Administration and the US Department of Health and Human Services provided an overview of the activities of the Organ Procurement and Transplantation Network and SRTR contractors. The dataset consists of kidney transplant donor and recipient clinical and outcome features originally spanning a period of 18 years from 2000 to 2017. There are a total of 165,090 records and 52 features. Each record is associated with features of one donor and one recipient individual. A description of all the features along with their feature name is provided in Table 1. The features include a wide range of characteristics of the individuals like height, weight and BMI as well as more specific illness-related features like a history of hypertension, diabetes and time on dialysis. Pandas dataframes were used to handle the datasets in python [84], [85].

Table 1: Description of variables in the dataset

Feature Name	**Feature Description**	**Feature Type**
dtype	Donor type	Categorical
TRR_ID	Individual transplant ID number	Categorical
rwt2	Recipient weight (kg)	Numerical
rht100	Recipient height (cm)	Numerical
dwt	Donor weight (kg)	Numerical
dht100	Donor height (cm)	Numerical
rbmisimp	Recipient BMI (kg/m^2) category	Categorical
dbmisimp	Donor BMI (kg/m^2) category	Categorical
functstat4	Functional status	Categorical
pkpragroup	Peak PRA group	Categorical
rsex	Recipient sex	Categorical
rracesimp	Recipient race	Categorical

Feature Name	Feature Description	Feature Type
esrddxsimp	Simplified End Stage Renal Disease diagnosis	Categorical
dsex	Donor sex	Categorical
dracesimp	Donor race	Categorical
dgf	Delayed Graft Function	Categorical
txtype	Transplant type	Categorical
rdm2	Recipient Diabetes	Categorical
hlamm	HLA mismatch	Categorical
rprvki	Previous kidney transplant	Categorical
cit	Cold ischemia time (hours)	Numerical
dage	Donor age at transplant (years)	Numerical
ragetx	Recipient age at transplant (years)	Numerical
dcmv	Donor CMV	Categorical
ddm	Donor Diabetes	Categorical
dhcv	Donor Hepatitis C virus	Categorical
dcd	Donation after cardiac death	Categorical
drcmv	Donor and recipient CMV	Categorical
drsex	Donor and recipient sex	Categorical
drrace	Donor and recipient race	Categorical
abshtdif	Absolute height difference between donor and recipient	Numerical
drwtdif	Donor and recipient weight difference	Numerical
dragedif	Donor and recipient age difference	Numerical
eventdt3	Minimum date of death, graft loss or censor	Categorical
event	Event	Categorical
survtime3	Survival time of recipient (or last follow-up)	Categorical
txdate	Transplant date	Categorical
graftfailure	Graft failure or death	Categorical
death	Death	Categorical
txfailuredt	Graft failure date	Categorical

Feature Name	Feature Description	Feature Type
txdeathdt	Recipient death date	Categorical
ecd	Expanded criteria donor	Categorical
rhtn	Recipient Hypertension	Categorical
rcvd	Recipient Cardiovascular disease	Categorical
rpvd	Recipient Peripheral Vascular disease	Categorical
rmalig	Recipient Malignancy	Categorical
rcmv	Recipient CMV	Categorical
dhtn2	Donor Hypertension	Categorical
preemptive	Pre-emptive transplant	Categorical
rcad	Recipient Coronary Artery disease	Categorical
vintage	Years on dialysis pre-transplant	Numerical
wit	Warm ischemia time (hours)	Numerical

3.2 PHASE 1: Data Preprocessing

3.2.1 Data Cleaning and Feature Engineering

Based on our analysis and experts' opinion, features have been deleted, modified and generated to resolve any inconsistencies that may exist in the dataset. Most of the preprocessing and cleaning were done in python using the pandas, numpy and datetime (for date-related operations) libraries [84]–[87].

There were some features that were unimportant for our analysis and these were removed. The features dtype and TRR_ID were removed since dtype only had one value and TRR_ID is simply an ID number that did not have any significance in the study. Any individuals with previous kidney transplants (rprvki) were eliminated followed by removing the feature since it had only one value. We wanted to observe the effects of using unpaired variables like sex, cmv, height, etc., of the donors and recipients instead of the paired variables for them. Hence, the drcmv, drsex, drrace, abshtdif and drwtdif variables were removed. Delayed graft function (dgf) was an outcome variable that we did not want to include in our analysis and it was removed. Warm ischemia time (wit) was deleted due

to having far too many missing values (>40%). The event date (eventdt3) was recalculated to reflect the earliest of the events (graft loss or death or censored) since in some cases it did not consist of the earliest event occurrence date. A new feature called txtoevent which contained the number of days between the transplant and the event date had been calculated. This was used to recalculate the survival time (survtime3) field since it had some incorrect values. The event date (eventdt3) column was dropped as it did not have any information now that would help with our clustering objective. Similarly, the graft failure date (txfailuredt) and recipient death date (txdeathdt) were removed. The empty values in the event, graftfailure and death variables were replaced with 0 for easier interpretability and processing in the future stages. Records with negative vintage values were removed since that is not viable. We also got rid of the recipient and donor BMI category features (rbmisimp and dbmisimp) because some of the values were missing. We recalculated these features after the imputation tasks. The set of attributes obtained at this stage is shown in Table 2. There are a total of 143,297 records at this stage.

Table 2: Variable type and status.

Variable Name	**Type - Status**
rwt2	Recipient - Incomplete
rht100	Recipient - Incomplete
dwt	Donor - Complete
dht100	Donor - Incomplete
functstat4	Recipient - Incomplete
pkpragroup	Recipient - Complete
rsex	Recipient - Incomplete
rracesimp	Recipient - Complete
esrddxsimp	Donor - Incomplete
dsex	Donor - Complete
dracesimp	Donor - Incomplete
txtype	Recipient - Complete
rdm2	Recipient - Complete
hlamm	Recipient - Complete

Variable Name	Type - Status
cit	Donor - Incomplete
dage	Donor - Complete
ragetx	Recipient - Complete
dcmv	Donor - Incomplete
ddm	Donor - Incomplete
dhcv	Donor - Incomplete
dcd	Donor - Incomplete
event	Recipient - Complete
survtime3	Recipient - Complete
graftfailure	Recipient - Complete
death	Recipient - Complete
ecd	Donor - Complete
rhtn	Recipient - Incomplete
rcvd	Recipient - Incomplete
rpvd	Recipient - Incomplete
rmalig	Recipient - Incomplete
rcmv	Recipient - Incomplete
dhtn2	Donor - Incomplete
preemptive	Recipient - Incomplete
rcad	Recipient - Incomplete
vintage	Recipient - Complete

3.2.2 Data Standardization

In order to impute the numerical variables in the next stage, the variables were first scaled using a method that did not forcefully center the data. This in turn avoids getting rid of any sparsity that may exist. For that purpose, scikit - learn's absolute maximum scaler (MaxAbsScaler) in python was used. This allows the variables to be scaled in a way that each of their respective maximum absolute values would be 1.0 [24].

3.2.3 Data Imputation

For the imputation step, there were 35 variables in our dataset after the cleaning and removal of some features as mentioned above. They are specified above in Table 2. Their status, whether complete or incomplete and if they are a recipient or donor variable is also shown. Only variables with lower than ~20% missing data are imputed which is approximately the maximum missingness in a variable at this stage.

Two imputation tasks were performed with different groups of attributes.

3.2.3.1 First Imputation Task

In this task, all the donor variables are involved in addition to the recipient age (ragetx) variable based on the expert's suggestion, given the completeness of data for donor parameters and recipient age in the SRTR. These 13 variables and the model used for the imputation of incomplete variables are shown in Table 3.

Table 3: First imputation task feature data types and imputation methods

Feature Name	Type	Model
dwt	Numerical	
dht100	Numerical	Predictive mean matching
dsex	Categorical - 2 levels	
dracesimp	Categorical - 3 levels	Polytomous logistic regression
cit	Numerical	Predictive mean matching
dage	Numerical	
ragetx	Numerical	
dcmv	Categorical - 2 levels	Logistic regression
ddm	Categorical - 2 levels	Logistic regression
dhcv	Categorical - 2 levels	Logistic regression
dcd	Categorical - 2 levels	Logistic regression
ecd	Categorical - 2 levels	

Feature Name	Type	Model
dhtn2	Categorical - 2 levels	Logistic regression

3.2.3.2 Second Imputation Task

For the second imputation task, all the recipient variables in addition to donor variables that were complete were included. These 27 variables and the corresponding imputation method for incomplete variables are shown in Table 4.

Table 4: Second imputation task feature data types and imputation methods

Feature Name	Type	Model
rwt2	Numerical	Predictive mean matching
rht100	Numerical	Predictive mean matching
dwt	Numerical	
functstat4	Categorical - 10 levels	Polytomous logistic regression
pkpragroup	Categorical - 3 levels	Polytomous logistic regression
rsex	Categorical - 2 levels	
rracesimp	Categorical - 3 levels	
esrddxsimp	Categorical - 5 levels	Polytomous logistic regression
dsex	Categorical - 2 levels	
txtype	Categorical - 2 levels	
rdm2	Categorical - 2 levels	Logistic regression
hlamm	Categorical - 7 levels	Polytomous logistic regression
dage	Numerical	
ragetx	Numerical	
event	Categorical - 3 levels	
survtime3	Numerical	
graftfailure	Categorical - 2 levels	
death	Categorical - 2 levels	
ecd	Categorical - 2 levels	

Feature Name	Type	Model
rhtn	Categorical - 2 levels	Logistic regression
rcvd	Categorical - 2 levels	Logistic regression
rpvd	Categorical - 2 levels	Logistic regression
rmalig	Categorical - 2 levels	Logistic regression
rcmv	Categorical - 2 levels	Logistic regression
preemptive	Categorical - 2 levels	Logistic regression
rcad	Categorical - 2 levels	Logistic regression
vintage	Numerical	

For both imputation tasks, we performed independent imputations using the 'mice' function from the mice R package [27]. The parameters used for both imputation tasks are the same besides the features involved and the models specified for those features. The parameters we specified for the 'mice' function are given in Table 5.

Table 5: Parameters for both imputation tasks

Parameter	Value
m (number of imputations generated)	5
maxit (number of iterations)	10
seed	2105

There is another parameter predictorMatrix which is used to set the predictors that will be used for the imputation tasks. For the first imputation task, we created a partition of the dataset with only donor variables and recipient age variables and so did not have to specify the predictorMatrix parameter since it uses all the variables present by default for the imputation process. Whereas in the case of recipient variable imputation, almost the entire dataset was imported and we set the predictorMatrix parameter entries to '0' (feature is not used for imputation) for the donor variables that had null values prior to imputation and for the txtoevent feature we created. The txtoevent variable is simply another representation of the recalculated survtime3 field and is redundant going further. In order

to set these entries of the predictorMatrix parameter, we had to do a dry run of the 'mice' function to generate that predictor matrix and then modify the values in it as suggested by Buuren and Groothius-Oudshoorn [27]. The reason for setting these parameter entries to '0' in the predictorMatrix rather than simply removing the variables in this step is to avoid confusion and further unnecessary processing going further.

The logistic regression model is used for the imputation of categorical values with 2 categories/levels while the polytomous logistic regression model is used when there are $>$ 2 levels. Predictive mean matching is a semi-parametric imputation model used for numerical variable imputation that ensures that the imputations respect the upper and lower bounds of the variable [27], [88]. This method and its benefits are further elaborated in the book by Buuren [89]. Imputations with linear regression (with predicted values), linear regression ignoring model error and linear regression with bootstrap were briefly experimented and the results are presented in section 4.1.

The default number of imputations produced by this method is 5. Liu and De discuss how generating multiple complete imputations help include the uncertainty from the imputation method and the variability of values in their analysis. However, that work revolves around combining estimates from the different methods in a setting such as regression analysis as also described in the documentation of the mice package [27], [30]. In the work by Liu and De, they also described picking a dataset at random, applying diagnostics and also seeing if similar results are obtained with other imputations [30]. However, for our work, we require only one dataset. As mentioned earlier in the background section, we are aware that doing so does not account for uncertainty from the imputation process. For that reason, we picked one dataset at random from the 5 different datasets produced for both imputation tasks (5 is the default number of imputations). We applied a non - parametric Kolmogorov-Smirnov test to compare the distributions of the imputed numerical variables with the original distributions of those variables. If the $p - value$ was less than 0.05, we would say that the distributions are different and hence an undesirable outcome. In the case of all the imputed variables (donor and recipient) from the imputation method we picked, the $p - value$ was $>$ 0.05 hence signifying that the distributions are similar. Also, a visual inspection of the density plots of the pre and post imputation numerical variables showed

extremely identical and nearly indistinguishable plots with our method. Some of the other models that we tested for imputing numerical variables led to visually different density plots as well as $p - value < 0.05$ for the Kolmogorov-Smirnov test. Liu and De suggested a similar approach to determining the quality of the imputation generated and has shown how different models can lead to different diagnostic results [30]. These results are further elaborated in section 4.1.

For reference, we randomly selected the 2nd imputed dataset generated from the first imputation task and the 4th imputed dataset generated from the second imputation task.

3.2.4 Additional Dataset Processing

After the imputation process, all the donor and recipient variables from both imputation tasks are combined into one complete dataset for further processing and analysis. We also apply an inverse transform to the numerical variables back to their unscaled form, using the same absolute maximum scaler (scikit-learn's MaxAbsScaler) [24].

The recipient and donor BMIs are calculated based on the now completed dataset. This is attained by dividing the weight (in kg) of the donor or recipient by the square value of the height (in cm) of the donor or recipient. Based on these calculations, the missing recipient and donor BMI category features (rbmisimp and dbmisimp) from our original dataset are calculated. These features are now made a part of the completed dataset for further analysis.

The transplant date is also added to the complete dataset at this point for the purpose of extracting the required cohort in further stages.

Based on experts' suggestions, we deleted records from the dataset where the recipient or donor BMIs were below 10 or above 100. The recipient and donor heights, weights and BMI calculations are removed from the dataset since they are no longer required. However, the categorical variables consisting of the BMI categories that were filled in as mentioned above (rbmisimp and dbmisimp), remain in the dataset.

3.2.5 Cohort Preparation

The entire dataset (140,000+ records) is very large to efficiently run experiments with the various methods and analyze the results due to the computational expense. So, a cohort was prepared at this stage for further experiments.

On the basis of the transplant date, we prepared a cohort consisting of transplants that took place between 2009 and the end of 2011. This produced 25,824 records. Selecting a cohort that is in the more contemporary time frame of the dataset would be expected to result in fewer outcomes being observed (due to the short observation period in a lot of cases) and selecting a cohort that is much earlier in the dataset might be outdated and not generalizable to the management of transplantation in the most current era. After picking this cohort, the transplant date (txdate) is eliminated from the dataset since it serves no further purpose.

We perform one final transformation of scaling the numerical variables in this dataset using the scikit-learn's MaxAbsScaler for our clustering work going ahead [24].

The outcome variables event, graftfailure, death, survtime3, txtoevent as well as ecd are excluded in the dataset we used for the clustering and only reported after the clusters were generated for analysis. This was because we did not want the outcome variables to influence the clustering methods. A breakdown of the records from each year in the cohort is given in Table 6.

Table 6: Number of records by year in the selected cohort.

Year	**Number of records**
2009	8,294
2010	8,557
2011	8,973

3.3 PHASE 2A: Clustering

There was a total of 28 variables obtained after the previous pre-processing and cohort preparation tasks in Phase 1 that contribute towards the clustering of the dataset as shown

in Table 7. In our work, we decided to go with less traditional and popular clustering methods. Several previous studies involving clustering algorithms relied on distance measures and numerical datasets. There are distance measures that can work with mixed datasets having categorical and numerical data. However, a lot of these measures and algorithms that rely on them, have very expensive computation requirements and suffer from various drawbacks that are discussed in the background section. We wanted to explore model-based clustering methods for the purpose of mixed data clustering. Also, according to the silhouette index, some of the algorithms that are used in this work performed significantly better than distance-based methods like k-prototype, partitioning around medoids and hierarchical clustering that are built to handle mixed datasets [37], [38], [90], [91]. k-medoids and hierarchical clustering used the Gower distance matrix for computation while k-prototypes uses its own measure. k-prototypes was employed with the clustMixType R package while portioning around medoids was from the cluster R package and hierarchical clustering, from the stats R package [91]–[93]. The two clustering algorithms selected to produce the *base clusterings* are Mixture Model and KAMILA. For both algorithms, we generated 3 clusters since it produced a better silhouette score than anything above 3 clusters and 2 clusters (binary problem) were not of interest in our research. The silhouette plot showing the decrease in scores when using the KAMILA clustering algorithm is shown in Figure 7 in Section 4.2.1. A similar plot with the SOM method is presented in Section 4.3.

Table 7: List of variables used for clustering.

functstat4	pkpragroup	rsex	rracesimp
esrddxsimp	dsex	dracesimp	txtype
rdm2	hlamm	cit	dage
ragetx	dcmv	ddm	dhcv
dcd	rhtn	rcvd	rpvd
rmalig	rcmv	dhtn2	preemptive
rcad	vintage	rbmisimp	dbmisimp

Outcome variables are not a part of the clustering process since we wanted to analyze how clusters were formed in their absence. However, when analyzing cluster results, we look at the presence of those outcome variables among generated clusters.

3.3.1 Mixture model

The first clustering algorithm that was utilized was a mixture model-based clustering method obtained from the Rmixmod package in R [45], [91]. The function used is called mixmodCluster and the parameters used are specified in Table 8. The method is able to differentiate between quantitative and qualitative attributes automatically and no discrete specification of data types or variable discretization is required [45].

Table 8: Parameters for mixture model clustering

Parameter	**Values**	**Selected Value**
nbCluster (number of clusters required)	3 - 8	3
criterion (criterion for picking the best model)	BIC, ICL, NEC	BIC
seed	-	5

The model used is called “Heterogeneous_pk_Ekjh_Lk_Bk” which indicates free and not necessarily equal proportions. Other models that were tested were “Heterogeneous_pk_Ekjh_Lk_B”, “Heterogeneous_pk_Ekjh_L_Bk” and “Heterogeneous_pk_Ekj_Lk_Bk” which produced worse results (silhouette score wise). There are 40 composite models available and when running a search based on BIC, ICL and NEC criteria for the best model, they all selected models that produced lower silhouette scores than our selected “Heterogeneous_pk_Ekjh_Lk_Bk” model. The various models differ in their parameters used in identifying clusters. When we tried to identify clusters greater than 3 with this method and the selected model, the model failed and produced errors. That could also indicate that a higher number of clusters are not suitable

for our dataset (with this model) because this algorithm fails to fit the components when $k > 3$ [45]. A similar issue was seen when we tried using another package that implemented a similar method. This method generates the first set of base clusters for our work.

3.3.2 KAMILA

KAy-means for Mixed Large data (KAMILA) is a semi-parametric clustering approach that we implement using the kamila package in R, developed by Foss and Markatou. The kamila function in this package is used to perform clustering [54]. Unlike the mixture model function we used, we are required to explicitly specify a dataframe of quantitative variables and a dataframe of qualitative variables. The parameters used for this clustering method are mentioned in Table 9.

Table 9: Parameters for KAMILA clustering

Parameter	**Values**	**Selected**
numClust (number of clusters required)	2-8	3
numInit (number of initializations)	30,40,100,1000	100
maxIter (maximum number of iterations in each run)	25,100,1000	1000

The selected parameter values obtained the best silhouette scores for the randomly set seed of 3 (for reproducibility). The decrease in Silhouette scores with the increase in clusters is shown in the results section. We obtain the 2nd set of base clustering assignments from this method.

3.4 PHASE 3: Cluster Ensemble Generation

The consensus among base clusterings is obtained with the help of three methods from the diceR package in R. The k-modes, Majority Voting and LCA consensus methods from this package are employed to generate the consensus clustering results [94].

3.4.1 k-modes

To generate a consensus using the k-modes algorithm, the 'k_modes' method from the diceR package is utilized [94]. k-modes is a relatively simple algorithm that was originally meant for clustering categorical variables and hence works well in order to obtain a consensus using base clustering assignment labels. It is similar in its operation to the very popular k-means clustering algorithm but is meant for categorical features rather than the numerical features that k-means is suited for. The parameters employed with this consensus method are mentioned in Table 10 below.

Table 10: Parameters for k-modes consensus function

Parameter	**Value**
is.relabelled	TRUE
seed	1

The parameter is.relabelled is set to TRUE just to allow the k_modes function to generate its own consensus labels instead of using the first clustering as a reference. Some other random seeds were experimented with but produced worse silhouette scores [94].

Shen et al. used k-modes as a consensus function on results from three different clustering algorithms of k-means, hierarchical and Expectation-Maximization clustering methods. The goal of their study was to identify subtypes of Pervasive Development Disorders (PDD) using a dataset of patients with PDD [59].

3.4.2 Majority Voting

Majority voting is a consensus method that is able to deal with the voting problem through a multiple regression approach and produces a final consensus partition in the form of a soft partition attained by cluster-label assignment probabilities being averaged [64].

The 'majority_voting' function from the diceR package is implemented to facilitate this method. The only parameter specified for this package is the 'is.relabelled = TRUE' parameter, identical to what is mentioned for the kmodes method ('k_modes' function) [94].

3.4.3 Latent Class Analysis (LCA)

The 'LCA' method from the diceR package is used to implement the LCA consensus algorithm in our work. There are only two parameters for this method, one of which is 'is.relabelled = TRUE' (same as in the other two consensus methods above) and seed which is set to 4 which gave the best silhouette score [94].

3.5 PHASE 2B: Self-Organizing Maps (SOMs)

Self-Organizing Maps are an effective approach to clustering as well as data visualizations as discussed previously. On their own, SOMs group samples closer together based on how similar they are [68]. With the help of a clustering algorithm being employed on the results generated by the SOM, we are allowed to observe visually distinguishable cluster partitions [70]. In our work, these SOMs are used to substantiate the clustering consensus results we obtain and allow us to make a stronger case for our results. This also further strengthens the case for using SOMs as an independent clustering approach, even in the presence of mixed-type attributes. Self-Organizing Maps also provide an effective means for visualization, especially considering how challenging the task of visualization tends to become in the case of mixed-type datasets. First, the SOM is generated using the scaled numerical variables and categorical variables which were also used in the base clustering tasks. We have implemented SOMs with the help of the aweSOM package in R [95]. We

apply a clustering algorithm to the SOM representation produced to obtain proper cluster assignments for all the samples in the dataset. aweSOM has an interactive interface that provides several tools and customization options. The parameters used to generate the SOM in our work are provided in Table 11.

Table 11: Parameters for Self-Organizing Maps method

Parameter	**Values**
Rows, Cols	20, 20
Topology	rectangular
Random seed	81016
Initialization	Random Obs
maxNA.fraction	0.25
rlen	100
Alpha (start, stop)	0.05, 0.01
Radius (start, stop)	11, -11

'rlen' represents the number of times the complete dataset is presented to the network; alpha is the learning rate and radius is the neighborhood radius.

For the hierarchical clustering of the SOM output, we applied the 'ward.D' linkage to generate 3 clusters. This linkage presented the best results based on silhouette score and representation (appropriately distinguishable clusters) among ward.D, single, complete, average and mcquitty linkage methods. We also tried the partitioning around medoids clustering on the SOM output and got lower silhouette scores than the selected method [95]. The visualizations from this method are presented in Section 4.3. A silhouette plot indicating the decrease in scores when $k > 3$ clusters are used is also presented in that section.

3.6 PHASE 4: Evaluation and Visualization

3.6.1 Internal Evaluation Indices

The Silhouette scores, Dunn's index and Calinski-Harabasz scores are used to briefly evaluate the clusters obtained from the various clustering methods. Since the data is of a mixed type, they are applied to the Gower's distance matrix calculated by the daisy method from the cluster package in R ("gower" metric). The Silhouette score is calculated with that same package as the Gower's distance matrix [93]. Dunn's index value calculation is obtained with the help of the clValid package in R [96]. The Calinski-Harabasz score is calculated using the fpc package in R [97].

3.6.2 Visualizations

t-SNE is used to represent the overall cluster structure produced by various methods. Heatmaps are used to understand cluster variable distributions as well as in identifying cluster agreements with the Self-Organizing Maps method.

3.6.2.1 t-Stochastic Neighbor Embedding (t-SNE)

In addition to the visualizations produced by SOMs from above, t-SNE is also used to show clusters obtained from the results of the clustering and consensus functions. A matrix constructed from the Gower's distance calculation is used to generate the 2-d representation using t-SNE. The 'Rtsne' function from the Rtsne package in R allows us to create this visualization by taking the Gower's distance matrix as input. The seed is randomly set to 2 and the parameter 'is_distance' is set to 'TRUE' [98]. The output of that function creates an object that we plot using the ggplot function from the tidyverse package in R [99]. The cluster memberships are also plotted with the points in the t-SNE plot to view the different clusters.

3.6.2.2 Heatmaps

Heatmaps are used to show the difference in the distribution of categorical variables among the clusters generated. This is done with the help of the heatmap method from the seaborn library as well as the matplotlib library in python [100], [101]. The pandas crosstab function is responsible for computing the categorical variable distributions being shown in the heatmaps [84], [85]. Each cell in the heatmap shows the presence of the corresponding level of that attribute in that cluster (1-100%). We have also used heatmaps to show the similarity between clusters produced by different methods.

3.6.3 2-D Tables

Means and standard deviations of the numerical variables among clusters are represented in simple 2-d tables. These tables help us get a better understanding of the numerical features present among the different clustering results. It is also used to present a detailed overview of final cluster compositions in Section 4.5.1.

3.6.4 Statistical Tests

The Kruskal-Wallis test for numerical variables and chi-square tests for categorical variables were performed using the 'kruskal.test' and 'chisq.test' functions respectively, from the stats package in R [91]. The parameter 'correct' is set to 'FALSE' for the chisq.test since we are not using the test on a 2×2 table. We have used a significance level of 0.05 in our work for both tests which is a common standard used in the field.

3.7 System and Packages Used

For our work, we constantly switched between python on Jupyter Notebook and R on RStudio [102], [103]. Since the dataset was of mixed type and imputation was required, acquiring packages and methods in any one environment or language proved challenging, hence the constant transition. Dataset handling, preprocessing and modification such as

standardization and recalculation took place primarily in python using Numpy and pandas packages [84]–[86]. Standardization was performed with the scikit-learn library in python [24]. Imputation was done in R using the mice package [27]. The base clustering and consensus clustering was attained using the Rmixmod, kamila and diceR packages in R [45], [54], [94], [104]. The Self-Organizing Maps were employed with the aweSOM package in R [95]. Heatmaps for visualizations were produced using the seaborn and Matplotlib libraries in python [100], [101]. t-SNE representations were obtained in R using the Rtsne package [98]. Additionally, some of the other packages that were used in R to facilitate data manipulation, processing, testing and method support are cluster, dplyr, tidyverse, tidymodels, ggplot2 and Rcpp [91], [93], [99], [105]–[110].

All the experiments and analyses were carried out on a machine with an Intel i7 - 9750H CPU (Base Clock: 2.6 GHz, Turbo Clock: 4.5 GHz) paired with 32 GB of RAM and a Windows 10 64-bit Operating system.

CHAPTER 4 RESULTS

Clustering analysis was performed on a cohort of 25,824 records consisting of clinical features of donors and recipients between the years of 2009 – 2011 (3 years). Following an elaborate pre-processing and imputation process, cluster algorithms and consensus clustering are employed to provide us with a set of results. Simultaneously, Self-Organizing Maps provides us with an alternate solution to support our consensus clustering results.

In the following section, Section 4.1, we examine the results from the imputation procedures following which we go discuss the individual clustering and consensus clustering results in Section 4.2. In Section 4.3, Self-Organizing Map results are presented and compared with ensemble clustering results. t-SNE visualizations are provided in Section 4.4. Finally, the cluster descriptions and distributions are detailed in Section 4.5.

4.1 PHASE 1 Results: Data Imputation

In the data imputation process, only variables with lower than 20% missingness are imputed. Four of the variables imputed are numerical - i.e. donor height (dht100), cold ischemia time (cit), recipient weight (rwt2) and recipient height (rht100). Eighteen of the imputed variables are categorical (binary and nominal). As mentioned in the methodology, multiple imputations are generated in both imputation tasks and a dataset is selected from each task. In addition to the statistical test for quantitative variables, we also compared the means and standard deviations of those variables pre and post-imputation to ensure the quality of imputation. Density plots are also used to support the imputations produced.

For the statistical test of quantitative variables, we used the nonparametric Kolmogorov-Smirnov (KS) test as suggested in the work by Liu and De [30]. The p-value is used to identify whether there exists a statistically significant difference in the distributions of the variable before and after imputation. When the $p - value < 0.05$, this is indicative of a significant difference and that would be undesirable since we would prefer that the variable distributions to not be significantly affected by the imputation process. Different imputation methods led to different $p - values$ being obtained as well as different density

plots. Predictive mean matching (pmm) is the method selected for quantitative variable imputation. Another method that was previously tried and tested from this package is the linear regression with predicted values [30]. Following statistical testing, the method was changed to pmm which obeyed all the bounds of the variable being imputed as well as consistently satisfied the statistical test for all the numerical variables being imputed. Comparisons of the $p - values$ from the two different methods from the first and second imputation tasks are presented in Table 12. When using the linear regression (with predicted values) method for numerical variable imputation, the cold ischemia time (cit) and recipient height (rht100) imputations did not satisfy our requirement for the KS statistical test and produced $p - values < 0.05$.

Table 12: Kolmogorov-Smirnov Test results p-value comparison between pmm and linear regression (with predicted values).

Method / Variable	Pmm	Linear Regression (predicted values)
cit	1	< 2.2e-16
dht100	1	1
rwt2	1	1
rht100	1	0.002

Predictive mean matching always produced consistent results with imputations within bounds (existing upper and lower bounds of the variable) and satisfied the statistical test. In most cases, it produced an almost visually indistinguishable density plot that compared the distributions of the variable pre and post-imputation.

We have primarily shown the cold ischemia time (cit) variable in the figures in this section since it has the largest number of missing values to be imputed for a numerical variable ($\sim 6\%$) and for easier comparison among imputation methods. More visually distinguishable plots can be seen for this variable when comparing methods.

Figure 2 and Figure 3 show the density plots representing pre and post-imputation values for the cold ischemia time (cit) variable using predictive mean matching and linear regression with predicted values respectively.

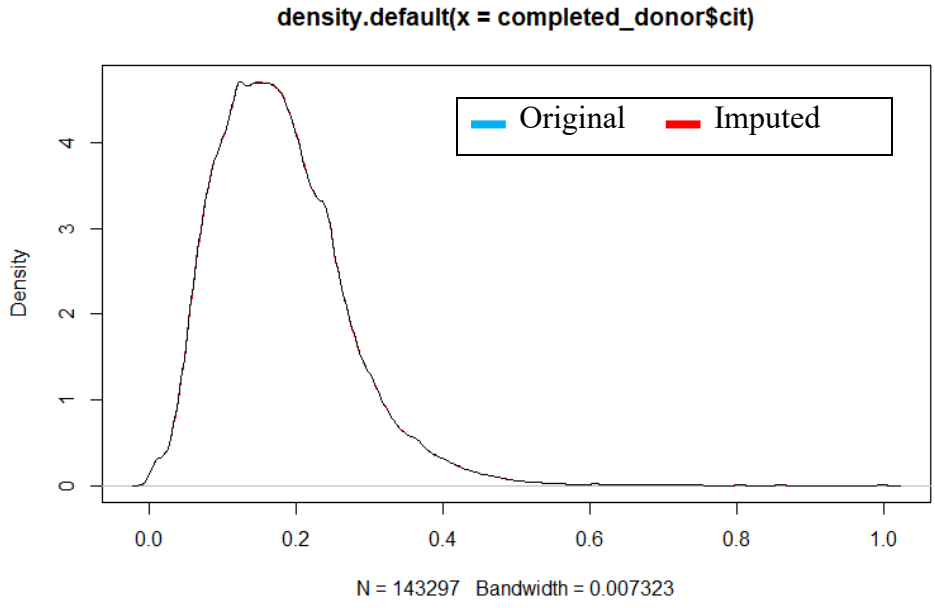

Figure 2: Density plot of the original dataset and dataset completed with imputation for cit variable with pmm

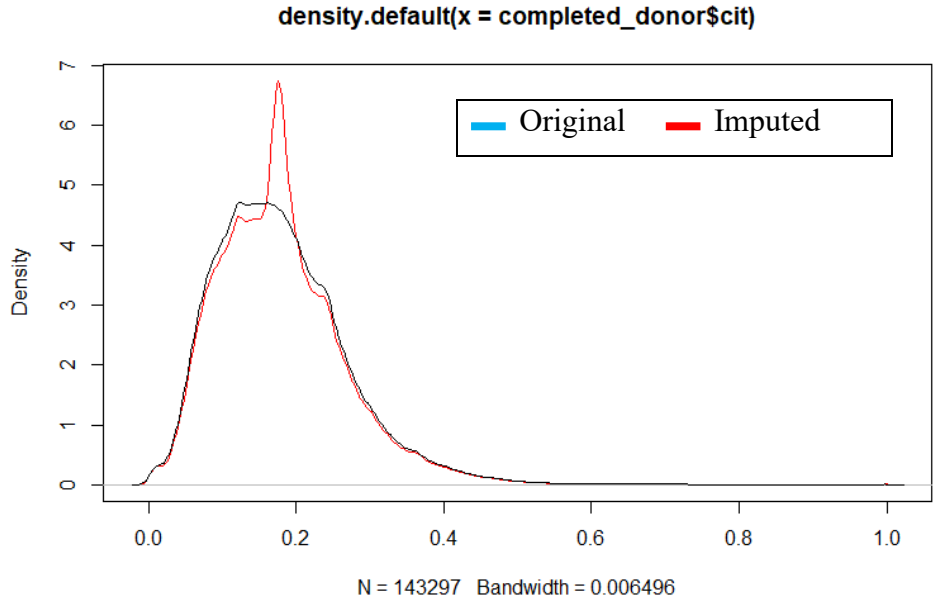

Figure 3: Density plot of the original dataset and dataset completed with imputation for cit variable with linear regression (with predicted values)

The recipient height variable (rht100) did not satisfy the KS test requirement when using the linear regression (with predicted values) imputation method. This variable did not

show as much distinction among density plots produced with the predictive mean matching (pmm) and linear regression (with predicted values) methods in comparison with the cit variable. Figure 4 shows the density plot for the rht100 variable using the pmm method while Figure 5 shows the density plot for that variable using linear regression (with predicted values).

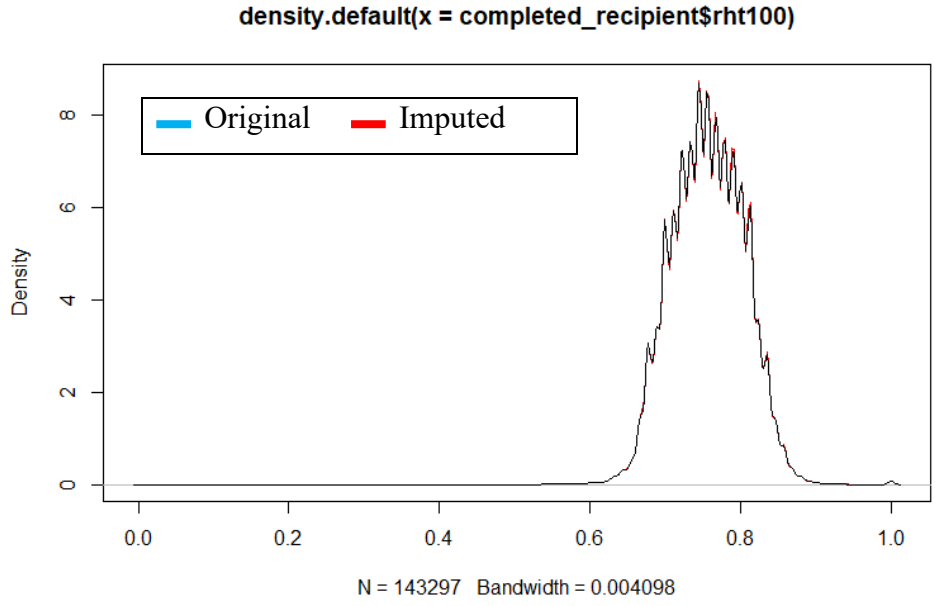

Figure 4: Density plot of the original dataset and dataset completed with imputation for rht100 variable with pmm

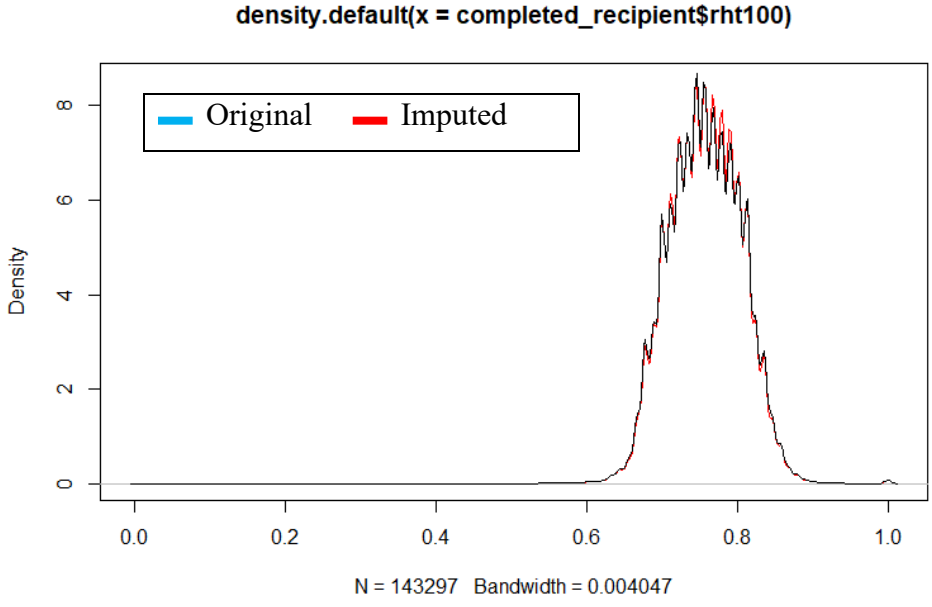

Figure 5: Density plot of the original dataset and dataset completed with imputation for rht100 variable with linear regression (with predicted values)

All of the alternate imputations generated match closely to the original dataset using the method we selected (pmm), as can be seen in Figure 6 which represents the density plot for the cold ischemia time (cit) variable consisting of the original data and all the possible alternate imputation values (5 different imputations). To be clear, this figure is showing distributions of the possible imputed values from the alternate imputations and not the dataset completed with imputed values as shown and compared in Figure 2.

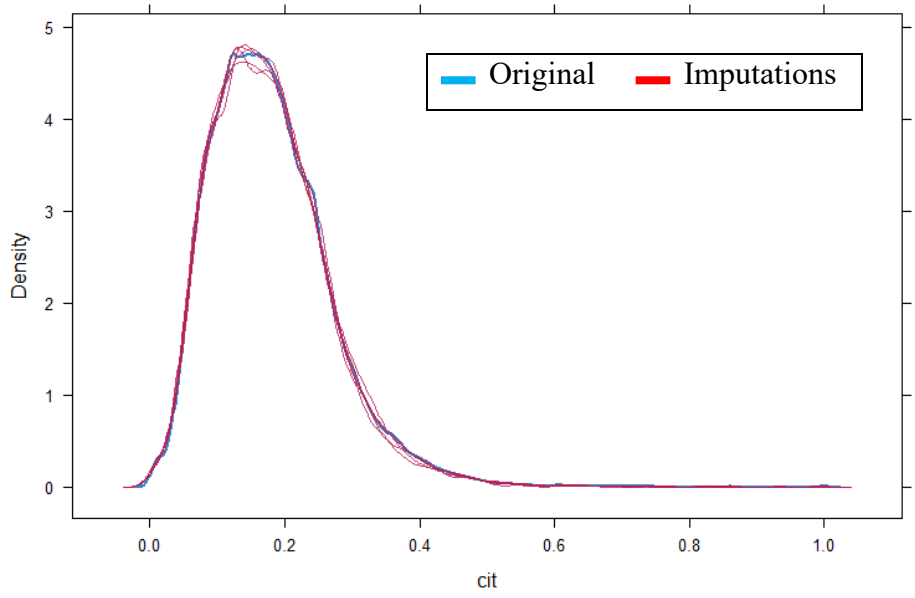

Figure 6: Density plot of the original dataset and multiple imputations generated for the cit variable with pmm

Additional tests involving other methods like linear regression ignoring model error and linear regression using bootstrap from the same package also satisfied the KS statistical test and provided similar density plots as the predictive mean matching methods we used [27]. However, in several of the cases involving these two methods, the imputations did not obey the bounds of the variable (by default) being imputed and even produced negative values which cannot be the case for those variables. Figure 7 shows the density plot of the cit variables' imputations produced with the linear regression using bootstrap method while Figure 8 shows the imputations with the linear regression ignoring model error method.

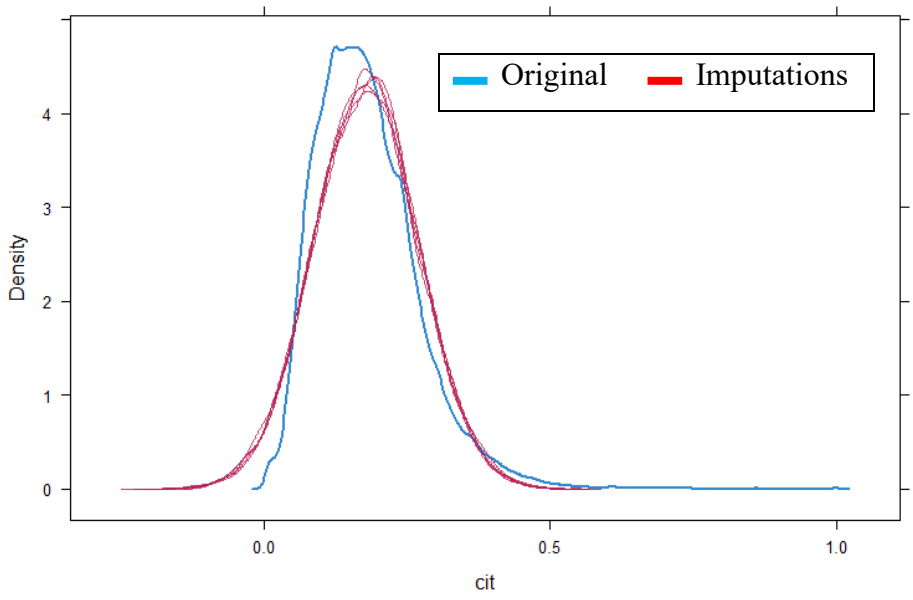

Figure 7: Density plot of the original dataset and multiple imputations generated for cit variable with linear regression using bootstrap

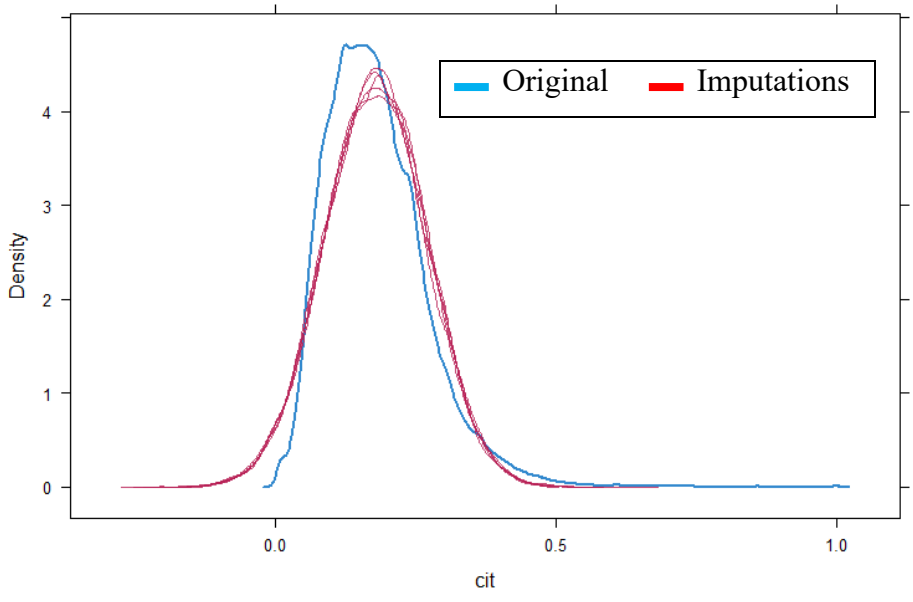

Figure 8: Density plot of the original dataset and multiple imputations generated for cit variable with linear regression ignoring model error

Results from the data imputation tasks are summarized below. Imputed numerical variable statistics (mean and s.d.) pre and post-imputation are mentioned in Table 13. The imputed

categorical variables are represented by their respective values' counts (and %) pre and post-imputation, in Table 14. In both tables, the frequency of missing values (and %) are mentioned with the feature names.

Table 13: Imputed numerical variable statistics (mean ± s.d) pre and post-imputation

	Donor height (cm) 5 (0.003)	**Cold Ischemia Time (hrs) 8059 (5.6)**	**Recipient weight (kg) 325 (0.2)**	**Recipient height (cm) 7484 (5.2)**
Mean ± s.d (original)	170.15 ± 14.53	17.77 ± 9.00	81.72 ± 19.21	170.31 ± 11.01
Mean ± s.d (completed)	170.15 ± 14.53	17.77 ± 8.99	81.72 ± 19.21	170.32 ± 11.00

Table 14: Imputed categorical variable distribution (frequency and %) pre and post-imputation

Feature	**Original (n = 143,297)**	**Imputed (n = 143,297)**
Functional Status - 7822 (5.4)		
10% - moribund	367 (0.2)	406 (0.2)
20% - very sick	1230 (0.8)	1333 (0.9)
30% - severely disabled	580 (0.4)	633 (0.4)
40% - disabled	2253 (1.5)	2396 (1.6)
50% - req consid assist	3356 (2.3)	3564 (2.4)
60% - req assist	11082 (7.7)	11665 (8.1)
70% - unable to do normal activity	21696 (15.1)	22644 (15.8)
80% - some sx	30243 (21.1)	31738 (22.1)
90% - minor sx	22916 (15.9)	24160 (16.8)
100% - no complaints	41752 (29.1)	44758 (31.2)
Peak PRA group - 23178 (16.1)		
0	89738 (62.6)	106040 (74.0)

Feature	Original (n = 143,297)	Imputed (n = 143,297)
1	20669 (14.4)	25409 (17.7)
2	9712 (6.7)	11848 (8.2)
ESRD Diagnosis - 4328 (3.0)		
Diabetes	40499 (28.2)	40842 (28.5)
GN	28331 (19.7)	29655 (20.6)
HTN	39501 (27.5)	40669 (28.3)
Other	18276 (12.7)	19221 (13.4)
PCKD	12362 (8.6)	12910 (9.0)
Donor Race - 30 (0.02)		
Black	19358 (13.5)	19364 (13.5)
Other	4656 (3.2)	4659 (3.2)
White	119253 (83.2)	119274 (83.2)
Recipient Diabetes - 1279 (0.8)		
DM	51382 (35.8)	51540 (35.9)
No DM	90636 (63.2)	91757 (64.0)
Donor Diabetes - 667 (0.4)		
Negative	133181 (92.9)	133781 (93.3)
Positive	9449 (6.5)	9516 (6.6)
HLA mismatch - 1073 (0.7)		
0	11070 (7.7)	11122 (7.7)
1	3440 (2.4)	3455 (2.4)
2	6363 (4.4)	6408 (4.4)
3	18488 (12.9)	18625 (12.9)
4	36568 (25.5)	36847 (25.7)
5	43797 (30.5)	44169 (30.8)
6	22498 (15.7)	22671 (15.8)
Recipient CMV - 6263 (4.3)		
Negative	43112 (30.0)	45127 (31.4)

Feature	Original (n = 143,297)	Imputed (n = 143,297)
Positive	93922 (65.5)	98170 (68.5)
Donor CMV - 600 (0.4)		
Negative	53432 (37.2)	53650 (37.4)
Positive	89265 (62.2)	89647 (62.5)
Donor Hepatitis C Virus - 240 (0.1)		
Negative	138962 (96.9)	139195 (97.1)
Positive	4095 (2.8)	4102 (2.8)
Donation after Cardiac death - 14 (0.009)		
No	125595 (87.6)	125607 (87.6)
Yes	17688 (12.3)	17690 (12.3)
Recipient Hypertension - 16543 (11.5)		
No	15287 (10.6)	18287 (12.7)
Yes	111467 (77.7)	125010 (87.2)
Recipient Cardiovascular disease - 24121 (16.8)		
No	115474 (80.5)	138790 (96.8)
Yes	3702 (2.5)	4507 (3.1)
Recipient Peripheral Vascular disease - 5491 (3.8)		
No	129621 (90.4)	134828 (94.0)
Yes	8185 (5.7)	8469 (5.9)
Recipient malignancy - 4728 (3.2)		
No	130479 (91.0)	134975 (94.1)
Yes	8090 (5.6)	8322 (5.8)
Donor Hypertension - 973 (0.6)		
No	103157 (71.9)	103835 (72.4)
Yes	39167 (27.3)	39462 (27.5)
Pre-emptive transplant - 1033 (0.7)		
No	126549 (88.3)	127420 (88.9)
Yes	15715 (10.9)	15877 (11.0)

Feature	Original (n = 143,297)	Imputed (n = 143,297)
Recipient Coronary Artery disease - 26591 (18.5)		
No CAD	105520 (73.6)	129807 (90.5)
CAD	11186 (7.8)	13490 (9.4)

4.2 PHASE 2A and PHASE 3 Results: Clustering

In this section, the results from individual clustering methods are presented, followed by consensus clustering results and their comparison with the individual clustering approach.

4.2.1 Base Clustering Results

This work focuses on the use of clustering algorithms that are capable of handling data consisting of qualitative and quantitative variables together. Two model-based clustering algorithms - i.e. Mixture model and KAMILA are employed to generate base clustering solutions. Other traditional distance-based clustering algorithms like k-prototypes, k-medoids and hierarchical clustering were briefly experimented with, to compare their performance with the algorithms we selected for our work. The distance-based methods produced worse silhouette scores for our selected number of clusters. Our selected methods are also less computationally expensive than some of the distance-based methods. As mentioned in the methodology section, the number of clusters (k) used in our work is 3. This was because if $k = 2$, it would result in a binary classification problem which would have had little clinical utility in the face of a single variable completely dichotomizing the dataset. Furthermore, any value > 3 affected clustering performances (visually and metric-wise). For the mixture model algorithm, $k > 3$ resulted in model errors possibly due to the data being unfit for more than 3 clusters. However, we were able to perform experiments with a different number of clusters with the KAMILA method. The change in silhouette scores with $k = 2\ to\ 8$ for this method is shown in Figure 9.

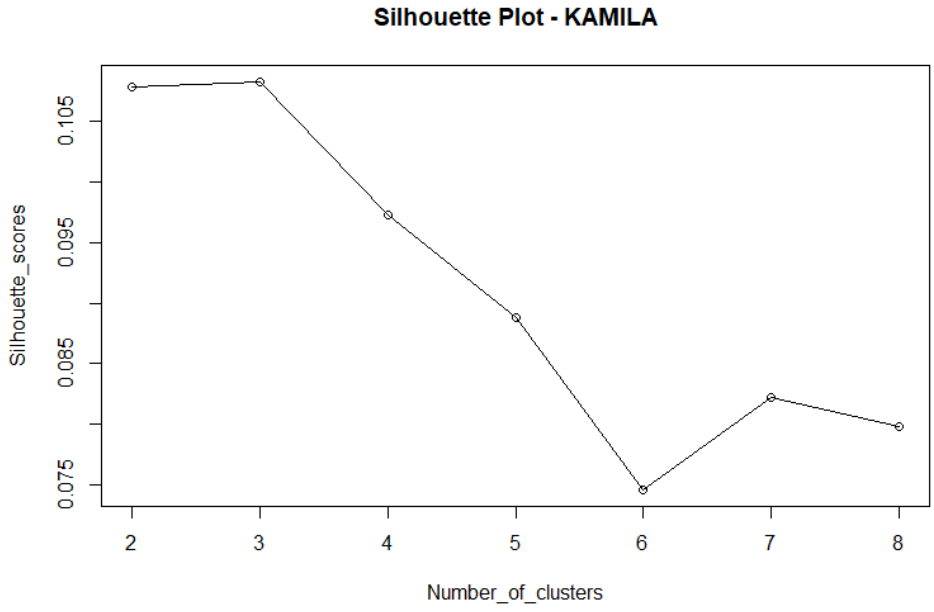

Figure 9: Silhouette score plot for k = 2 to 8 with KAMILA

A similar trend can be seen in the silhouette score plot using the SOM method as shown later in section 4.3.

We identified three internal evaluation indices that could work with our data. Based on the internal evaluation indices, KAMILA performed better among the two base clustering algorithms. Table 15 shows the scores of the various evaluation indices using the two methods when $k = 3$.

Table 15: Internal evaluation indices scores for base clustering algorithms with k = 3

Method	**Silhouette Scores (↑)**	**Dunn Index (↑)**	**CH Index (↑)**
KAMILA	0.108	0.0536	2760
Mixture model	0.091	0.0099	2514

These indices reflect the cohesion, separateness and compactness in the clusters produced by the clustering algorithm. A higher score is preferred for all three indices. The KAMILA

method produces better scores in all the metrics. These metrics have been calculated using the Gower distance matrix computed for mixed-type data.

The numerical variable means and standard deviations among clusters generated with the KAMILA method are shown in Table 16.

Table 16: Numerical variable statistics (mean ± s.d) among clusters with the KAMILA method

Cluster #	**cit (hrs)**	**dage (yrs)**	**ragetx (yrs)**	**survtime3 (yrs)**	**txtoevent (days)**	**vintage (yrs)**
1	16.96 ± 9.88	34.12 ± 15.17	49.93 ± 13.61	5.29 ± 2.13	1932.3 ± 781.5	3.54 ± 3.52
2	17.10 ± 9.31	36.98 ± 15.75	57.56 ± 9.71	4.99 ± 2.23	1824.7 ± 815.7	3.33 ± 2.77
3	17.80 ± 9.66	50.39 ± 10.32	57.09 ± 11.79	4.76 ± 2.35	1740.5 ± 859.4	3.57 ± 3.30

From the above table. a clear distinction can be observed in some of the numerical variables between clusters, particularly donor age (dage), recipient age (ragetx) and survival time (survtime3).

Heatmaps using the KAMILA method for some categorical variable distributions among clusters where there is a visible distinction are presented below.

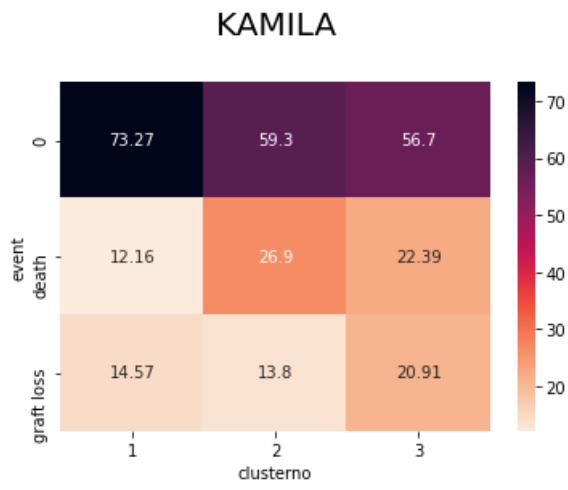

Figure 10: event variable distribution among clusters using the KAMILA method

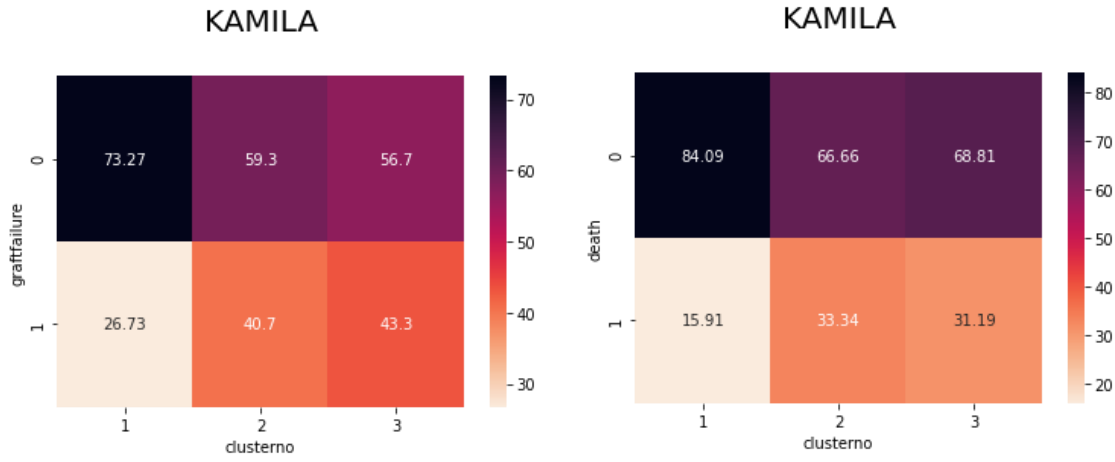

Figure 11: graftfailure (left) and death (right) variable distribution among clusters using the KAMILA method

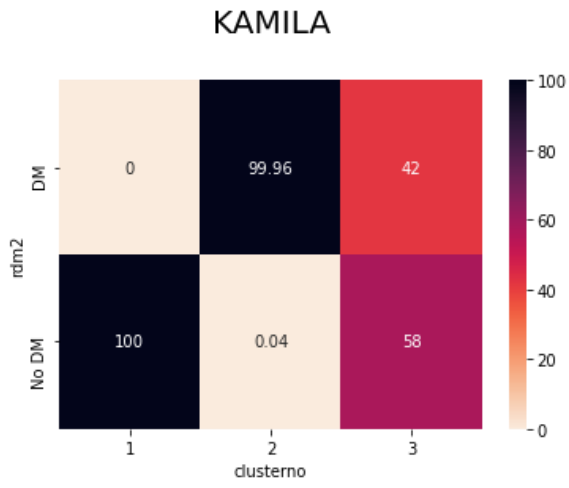

Figure 12: rdm2 variable distribution among clusters using the KAMILA method

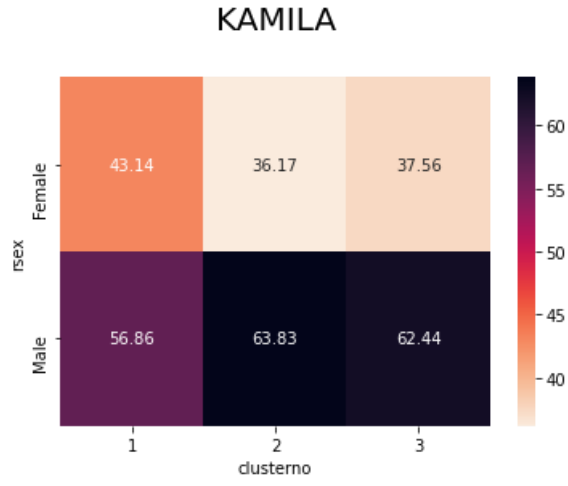

Figure 13: rsex variable distribution using the KAMILA method

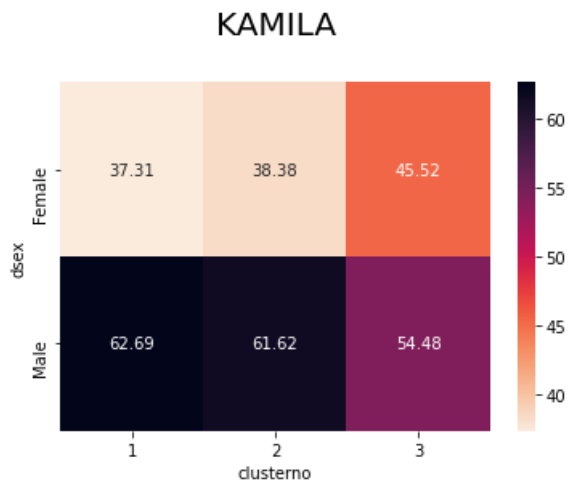

Figure 14: dsex variable distribution using the KAMILA method

4.2.2 Cluster Ensemble Results

To obtain a consensus among the two base clustering results, three consensus methods are used - i.e. k-modes, Majority voting and Latent Class Analysis (LCA). Traditionally used to cluster categorical data, k-modes and Latent Class Analysis (LCA) are able to work as consensus functions by treating the cluster label assignments from the base clustering solutions as categorical variables and obtaining a consensus. Table 17 shows the performances of the three consensus methods using the internal evaluation metrics.

Table 17: Internal evaluation indice scores for consensus clustering algorithms with k = 3

Method	**Silhouette Scores (↑)**	**Dunn Index (↑)**	**CH Index (↑)**
k-modes	0.108	0.0200	2704
Majority Voting	0.076	0.0282	2065
Latent Class Analysis (LCA)	0.113	0.0538	2843

LCA produced the best scores among the three consensus methods while majority voting performed the worst in terms of silhouette and Calinski-Harabasz scores. Additionally, LCA also attained the best evaluation scores and visualizations in comparison with the

base clustering algorithms. Moving forward, this is the method we will use to primarily describe our consensus clustering results.

The numerical variable means and standard deviations among clusters generated with the LCA method are shown in Table 18.

Table 18: Numerical variable statistics (mean ± s.d) among clusters with the LCA method

Cluster #	**cit (hrs)**	**dage (yrs)**	**ragetx (yrs)**	**survtime3 (yrs)**	**txtoevent (days)**	**vintage (yrs)**
1	17.13 ± 8.96	41.33 ± 15.83	58.28 ± 9.50	4.86 ± 2.28	1777.7 ± 833.4	3.34 ± 2.69
2	16.96 ± 9.88	34.12 ± 15.17	49.93 ± 13.61	5.29 ± 2.13	1932.3 ± 781.5	3.54 ± 3.52
3	18.16 ± 10.49	49.69 ± 10.23	55.35 ± 12.98	4.88 ± 2.33	1784.6 ± 853.8	3.71 ± 3.69

In the above table, we can observe a clear distinction among some of the variable means and standard deviations, particularly in the cases of donor age (dage), recipient age (ragetx) and survival time (survtime3).

Some of the categorical variable distributions with visible distinctions among clusters are shown below.

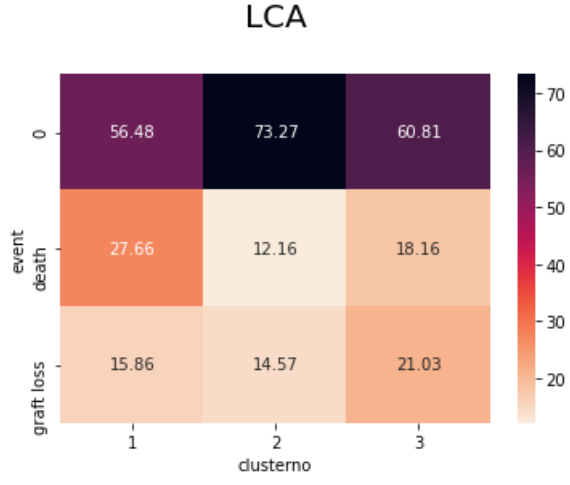

Figure 15: event variable distribution among clusters using the LCA method

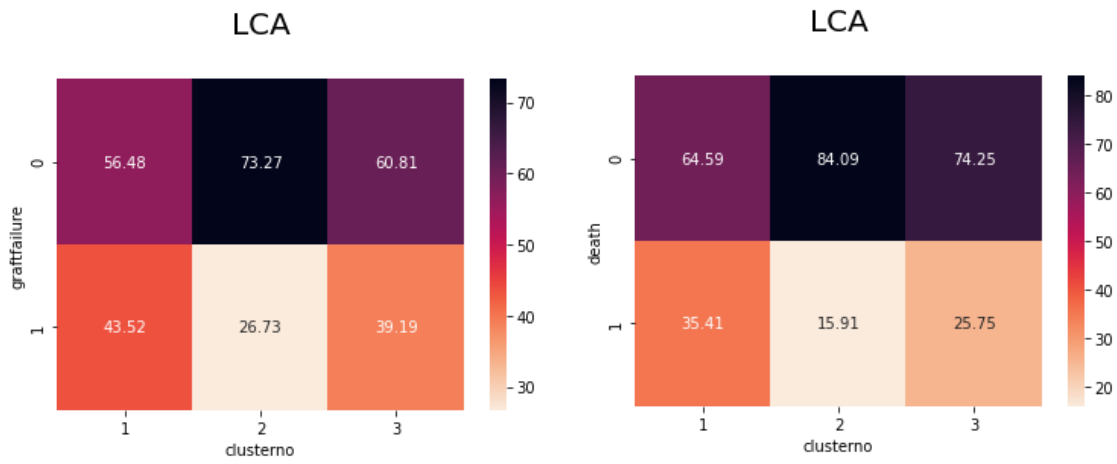

Figure 16: graftfailure (left) and death (right) variable distribution among clusters using the LCA method

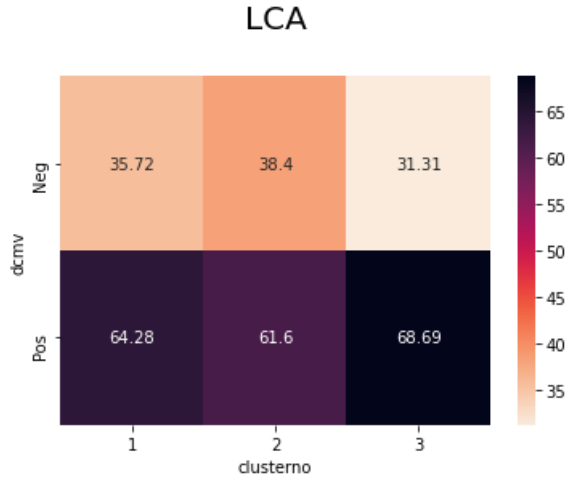

Figure 17: dcmv variable distribution among clusters using the LCA method

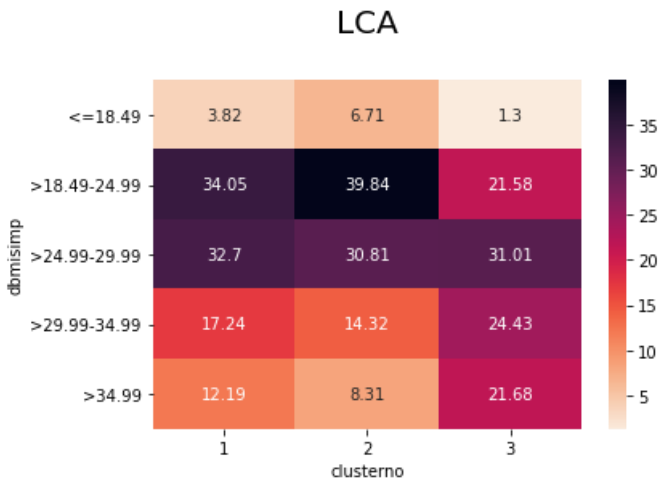

Figure 18: dbmisimp variable distribution among clusters using the LCA method

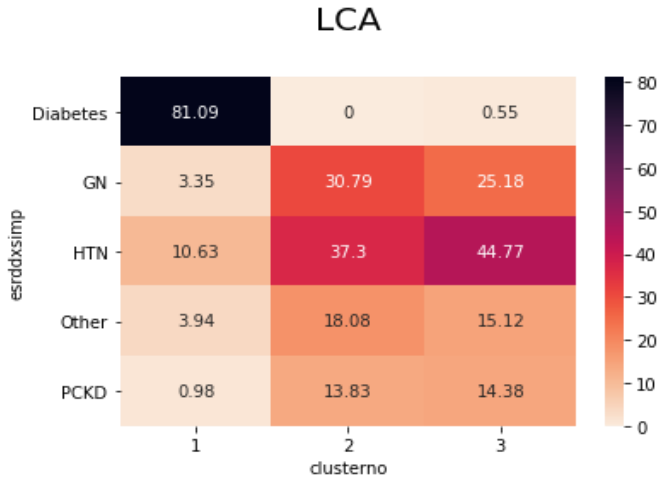

Figure 19: esrddxsimp variable distribution among clusters using the LCA method

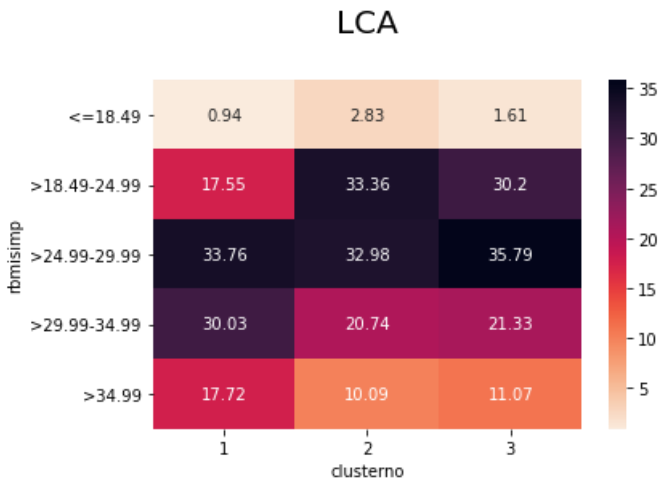

Figure 20: rbmisimp variable distribution among clusters using the LCA method

4.2.3 Comparison between Individual and Ensemble Clustering Results

Consensus clustering is a method of obtaining a more robust and higher-quality solution than individual clustering algorithms. In our work, Latent Class Analysis performed the best among all the methods involved (individual and consensus clustering). This can be seen using the internal evaluation indices' scores shown in Table 19.

Table 19: Evaluation indices' scores among various methods for 3 clusters

Method	**Silhouette Scores (↑)**	**Dunn Index (↑)**	**CH Index (↑)**
Individual Clustering			
KAMILA	0.108	0.0536	2760
Mixture model	0.091	0.0099	2514
Consensus Clustering			
k-modes	0.108	0.0200	2704
Majority Voting	0.076	0.0282	2065
Latent Class Analysis (LCA)	0.113	0.0538	2843
Alternative Solution			

Method	Silhouette Scores (↑)	Dunn Index (↑)	CH Index (↑)
Self-Organizing Map (SOM)	0.100	0.0512	2595

Although the internal evaluation scores of the LCA and SOM are slightly different, which could be due to the distance measure used, the two methods produce very identical results in variable distributions among clusters and t-SNE visualizations. This is discussed in Section 4.3.

In terms of cluster sizes, there is a commonality shared by the various methods. There is usually a cluster that is evidently the biggest, followed by two smaller clusters. The magnitude of variation in size between the two smaller clusters depends on the algorithm used. Table 20 describes the cluster sizes produced by the various methods.

Table 20: Cluster sizes generated by the various methods for k = 3

Method	Cluster Sizes (in decreasing order)
Individual Clustering	
KAMILA	11472, 7628, 6724
Mixture model	11061, 9098, 5665
Consensus Clustering	
k-modes	12499, 9184, 4141
Majority Voting	15699, 7266, 2859
Latent Class Analysis (LCA)	11472, 9583, 4769
Alternative Solution	
Self - Organizing Maps (SOMs)	12472, 9443, 3909

A comparison between LCA and KAMILA for some of the categorical variable distributions is presented below. An important point to be noted when comparing methods is that the cluster numbers are arbitrary and do not have any meaning by themselves. We

can observe how the variable distributions among clusters differ between methods in these heatmaps.

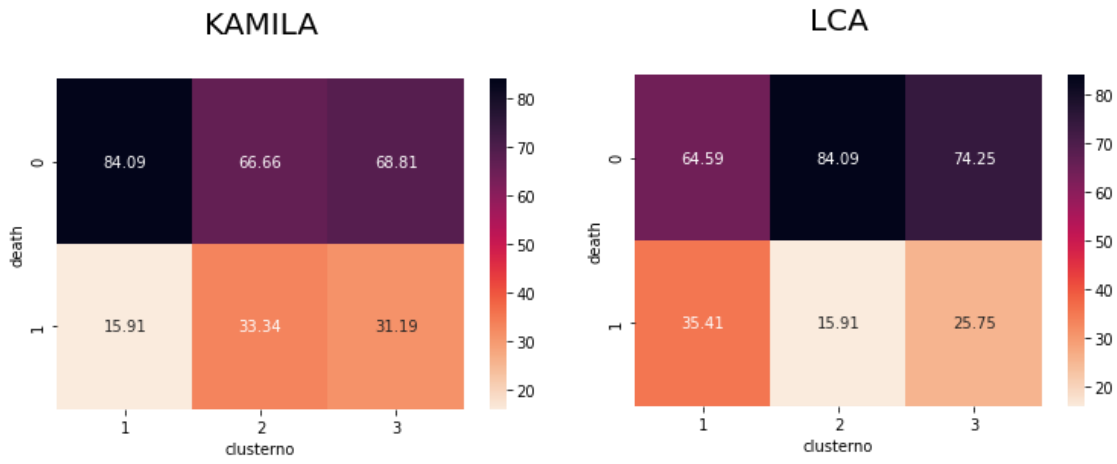

Figure 21: death variable cluster distribution with the KAMILA (left) and LCA (right) methods

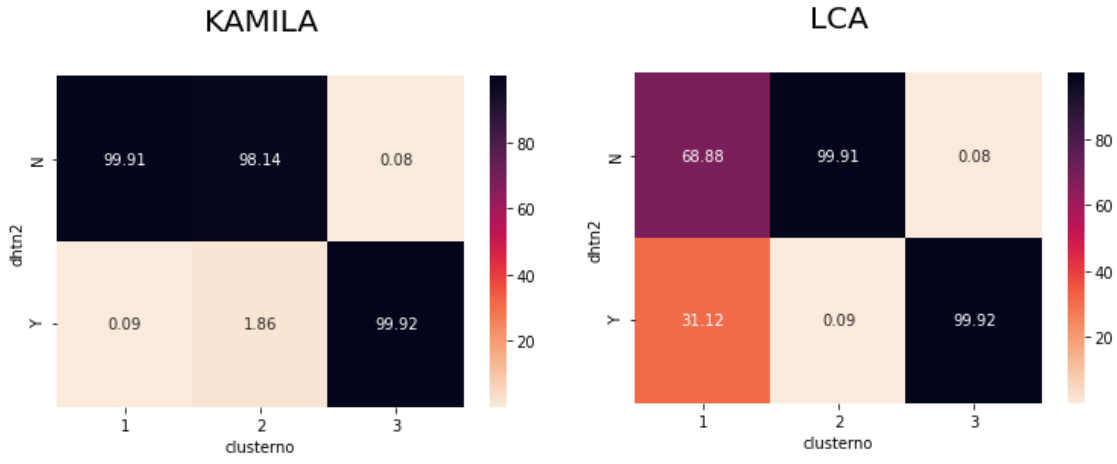

Figure 22: dhtn2 variable cluster distribution with the KAMILA (left) and LCA (right) methods

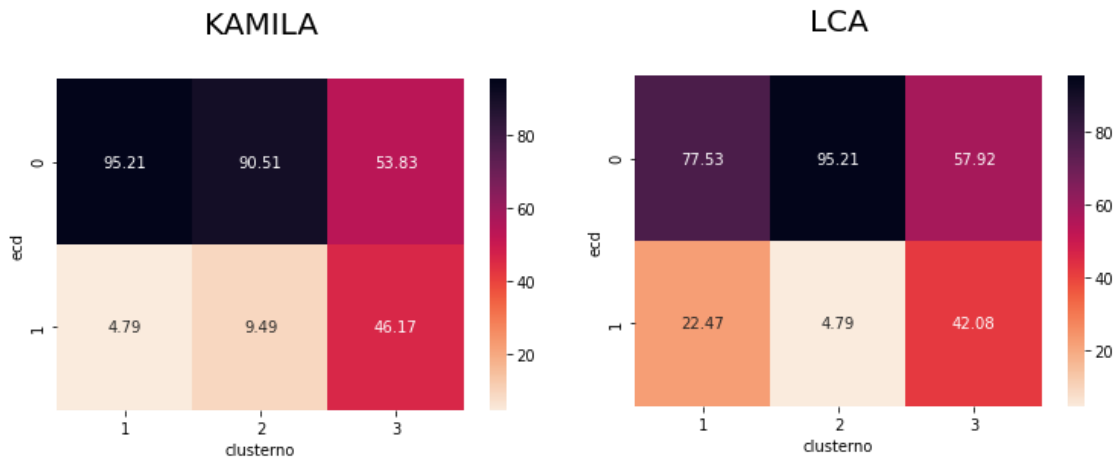

Figure 23: ecd variable cluster distribution with the KAMILA (left) and LCA (right) methods

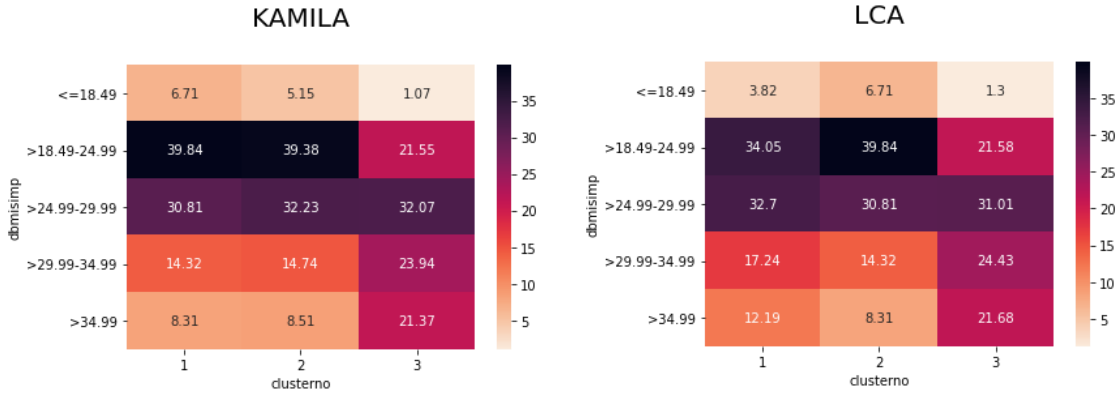

Figure 24: dbmisimp variable cluster distribution with the KAMILA (left) and LCA (right) methods

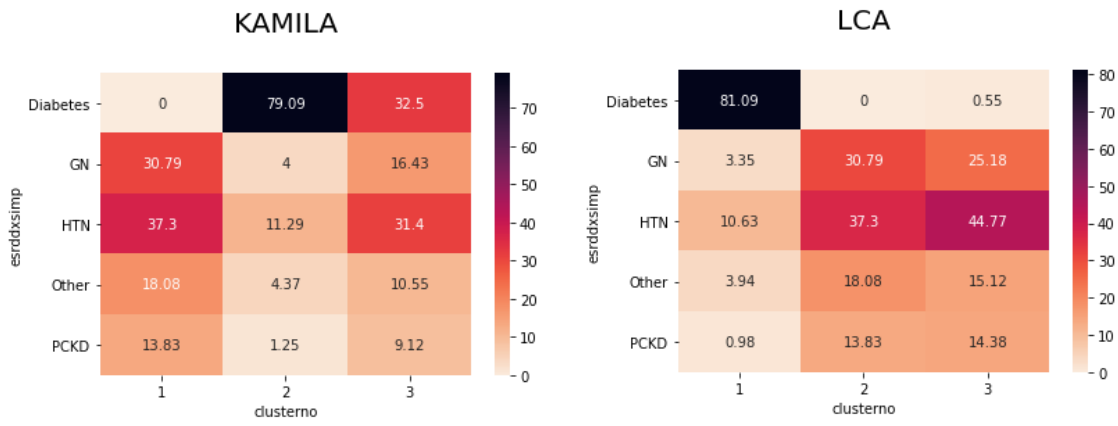

Figure 25: esrddxsimp variable cluster distribution with the KAMILA (left) and LCA (right) methods

In the majority of heatmaps shown above, we can observe the LCA consensus method producing a better distinction among clusters than the individual clustering method of KAMILA.

As we will see with respect to t-SNE visualizations as well, the cluster ensemble of LCA produces better clusters than any of the individual clustering algorithms. Cluster interpretation and descriptions from the LCA method are provided in Section 4.5.

4.3 PHASE 2B and PHASE 3 Results: Self-Organizing Maps

In our work, SOMs are used to provide an alternate solution that supports our consensus clustering results. This provides an additional method of validation for our approach in the absence of labels or ground truth for mixed type data. Similar to the silhouette plot in section 4.2.1 with the KAMILA algorithm, Figure 26 shows an identical trend in silhouette scores using the SOM algorithm where $k > 3$ clusters results in lower scores.

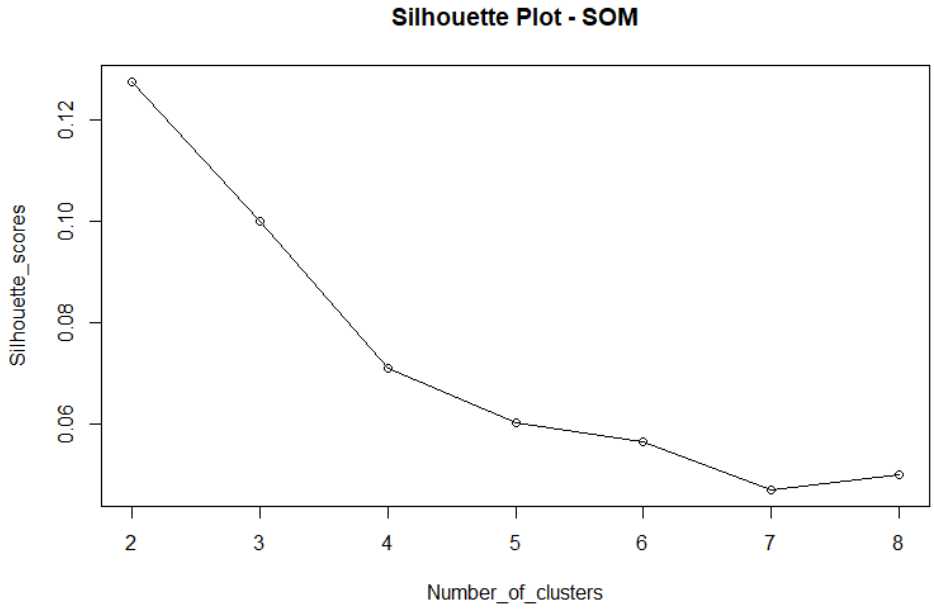

Figure 26: Silhouette score plot for k = 2 to 8 with SOM

The results obtained from SOMs match very closely with that produced by the ensemble clustering methods, particularly LCA. This is elaborated further in section 4.3.1. Figure 27 shows the average distances between units (or cells) of the Self-Organizing Maps representation generated from our data. This is based on a 20 x 20 rectangular topography SOM representation.

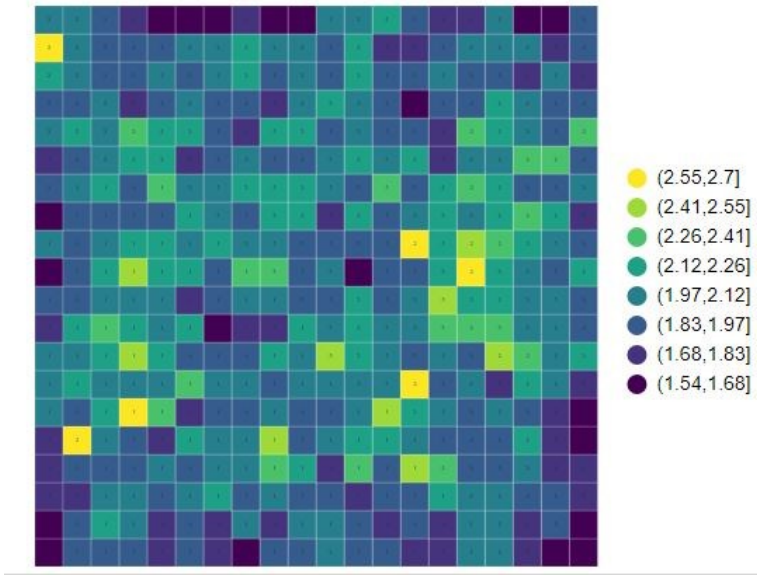

Figure 27: Average neighbor distance SOM representation

Each of the cells in this representation is a collection of points. SOMs are traditionally a visualization technique algorithm that produces a 2-d representation of the data. These SOMs inherently group similar items closer together. However, to obtain clearly marked boundaries and clusters, a clustering algorithm is applied to the SOM representation. In our work, we used hierarchical clustering with the ward D linkage measure on the SOM representation to obtain the distinct clusters shown in Figure 28. ward D produced a better SOM visualization and silhouette scores (clearer and more tightly grouped clusters) compared to the other linkage measures like average, single, mcquitty and complete linkage. It also produced a better result in comparison to the partitioning around medoids clustering algorithm applied to the SOM.

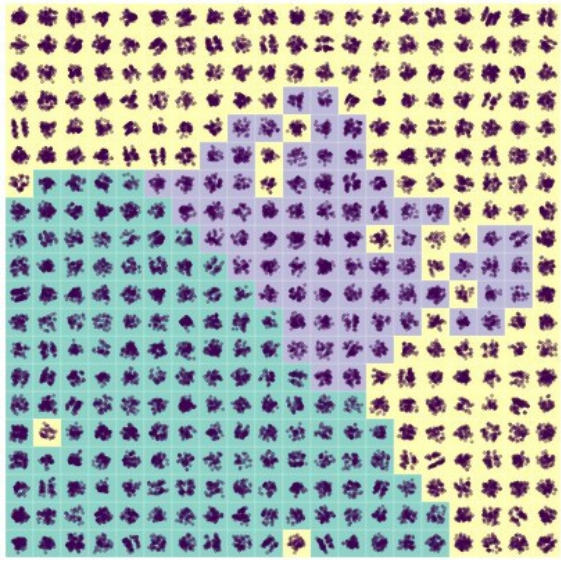

Figure 28: Rectangular SOM representation with 3 clusters using ward D linkage

We can also visualize how some features are distributed among clusters in the SOM representation. This closely matches the corresponding heatmap of that categorical variable. Figure 29 shows the donor hypertension (dhtn2) variable distributed among clusters in the SOM representation while Figure 30 shows the same variable's distribution as a heatmap.

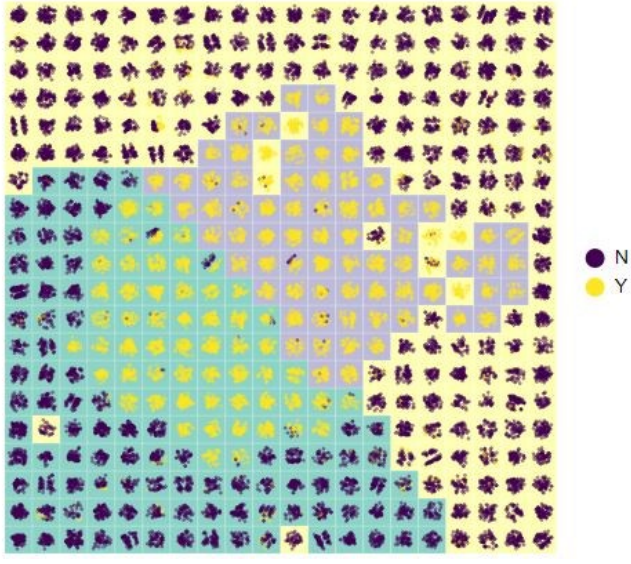

Figure 29: dhtn2 variable composition among clusters in SOM representation

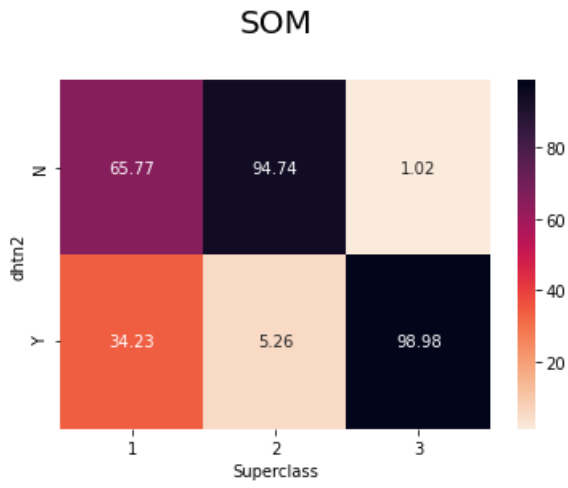

Figure 30: dhtn2 variable distribution among clusters using the SOM method

In Figure 29, we can observe how there is one cluster majorly composed of donors with no hypertension, one cluster almost entirely with donors having hypertension and a cluster that has a mixture of donors with and without hypertension. This is also reflected in the heatmap of that variable shown in Figure 30 above.

Numerical variable statistics (mean ± s.d) among clusters obtained with SOM are presented in Table 21.

Table 21: Numerical variable statistics (mean ± s.d) among clusters with the SOM method

Cluster #	**cit (hrs)**	**dage (yrs)**	**rage (yrs)**	**survtime3 (yrs)**	**txtoevent (days)**	**vintage (yrs)**
1	17.38 ± 9.49	41.90 ± 15.76	58.36 ± 9.55	4.84 ± 2.28	1770.3 ± 835.5	3.41 ± 2.84
2	16.93 ± 9.78	34.92 ± 15.35	50.30 ± 13.57	5.25 ± 2.16	1920.6 ± 790.5	3.50 ± 3.51
3	17.93 ± 9.72	49.43 ± 10.32	55.47 ± 12.93	4.93 ± 2.30	1801.6 ± 841.9	3.68 ± 3.50

We can observe clearly distinguishable differences among numerical variables, especially for donor age (dage), recipient age (ragetx) and survival time (survtime3) from the above table.

Some of the categorical variables where a clear difference among clusters can be observed are presented below.

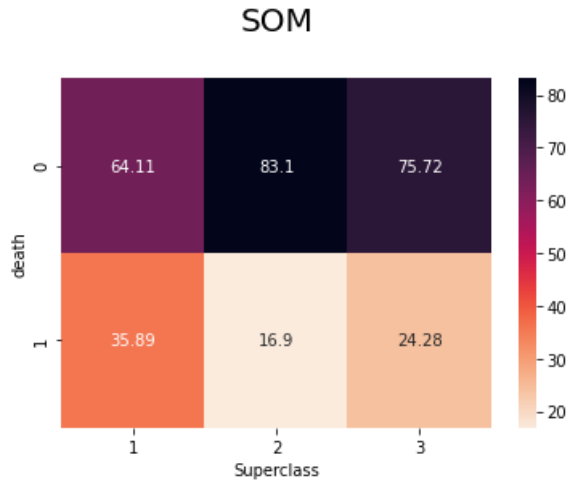

Figure 31: death variable distribution among clusters using the SOM method

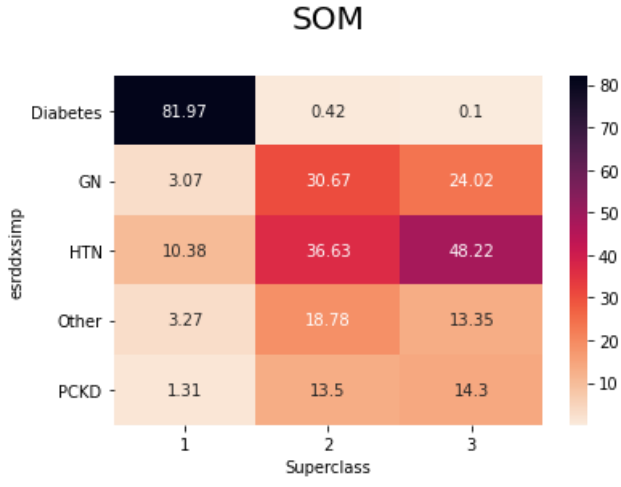

Figure 32: esrddxsimp variable distribution among clusters using the SOM method

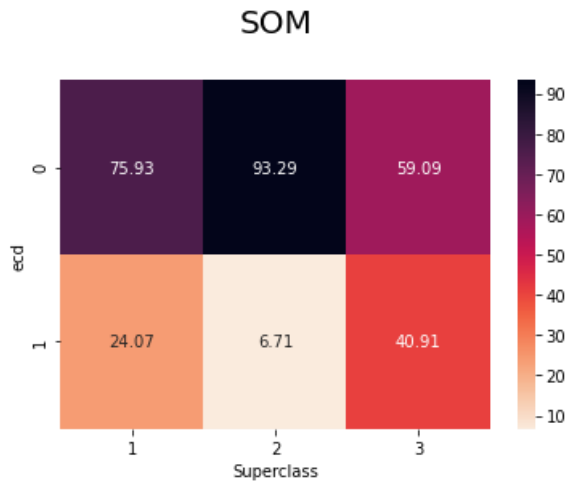

Figure 33: ecd variable distribution among clusters using the SOM method

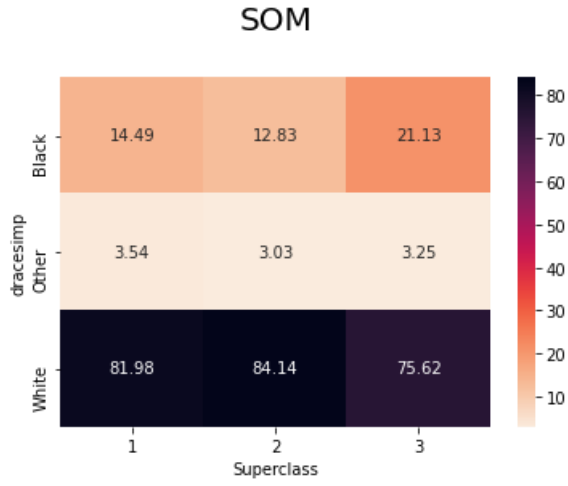

Figure 34: dracesimp variable distribution among clusters using the SOM method

4.3.1 Comparison between Self-Organizing Maps and Consensus Clustering

Self-Organizing Maps produce a very similar result to that of our consensus clustering methods, particularly Latent Class Analysis. Cluster sizes between SOMs and LCA are relatively similar as shown in Table 22.

Table 22: Cluster Sizes generated by LCA and SOM

Method	Cluster Sizes (in decreasing order)
Latent Class Analysis (LCA)	11472, 9583, 4769
Self - Organizing Maps (SOMs)	12472, 9443, 3909

Table 23 below shows the comparison between numerical variable means and standard deviations produced by the two methods. The numerical variable statistics among clusters generated by the two methods are identical for all the variables involved.

Table 23: Numerical variable statistics (mean ± s.d) among clusters with the LCA and SOM method

Method	Cluster #1	Cluster #2	Cluster #3
Cold Ischemia Time (cit) - hrs			
LCA	17.13 ± 8.96	16.96 ± 9.88	18.16 ± 10.49
SOM	17.38 ± 9.49	16.93 ± 9.78	17.93 ± 9.72
Donor Age (dage) - yrs			
LCA	41.33 ± 15.83	34.12 ± 15.17	49.69 ± 10.23
SOM	41.90 ± 15.76	34.92 ± 15.35	49.43 ± 10.32
Recipient Age (ragetx) - yrs			
LCA	58.28 ± 9.50	49.93 ± 13.61	55.35 ± 12.98
SOM	58.36 ± 9.55	50.30 ± 13.57	55.47 ± 12.93
Survival Time (survtime3) - yrs			
LCA	4.86 ± 2.28	5.29 ± 2.13	4.88 ± 2.33
SOM	4.84 ± 2.28	5.25 ± 2.16	4.93 ± 2.30
Days between transplant and event (txtoevent) - days			
LCA	1777.7 ± 833.4	1932.3 ± 781.5	1784.6 ± 853.8
SOM	1770.3 ± 835.5	1920.6 ± 790.5	1801.6 ± 841.9
No. of years on dialysis pre-transplant (vintage) - yrs			
LCA	3.34 ± 2.69	3.54 ± 3.52	3.71 ± 3.69
SOM	3.41 ± 2.84	3.50 ± 3.51	3.68 ± 3.50

The similarity between results produced by the two methods is also reflected in categorical variables as shown in the heatmaps below.

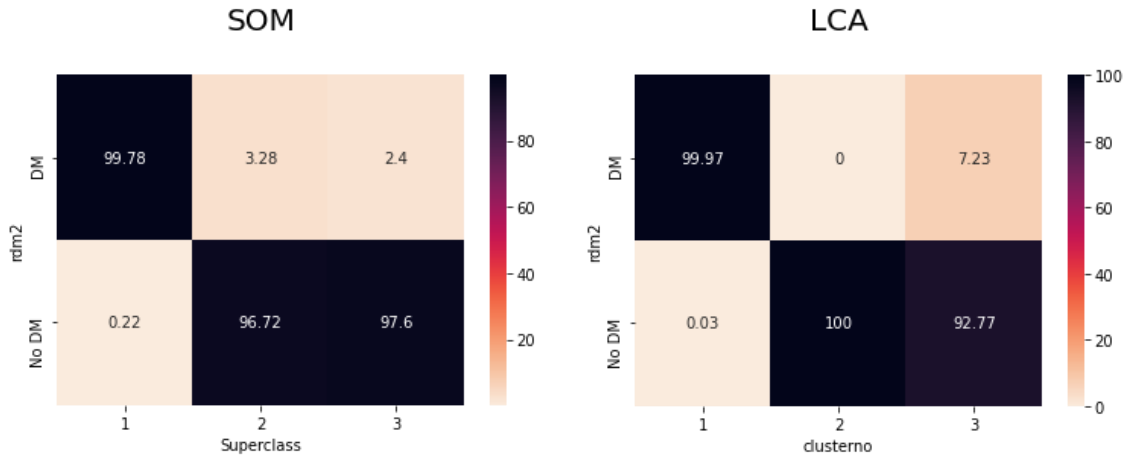

Figure 35: rdm2 variable cluster distribution between the SOM (left) and LCA (right) methods

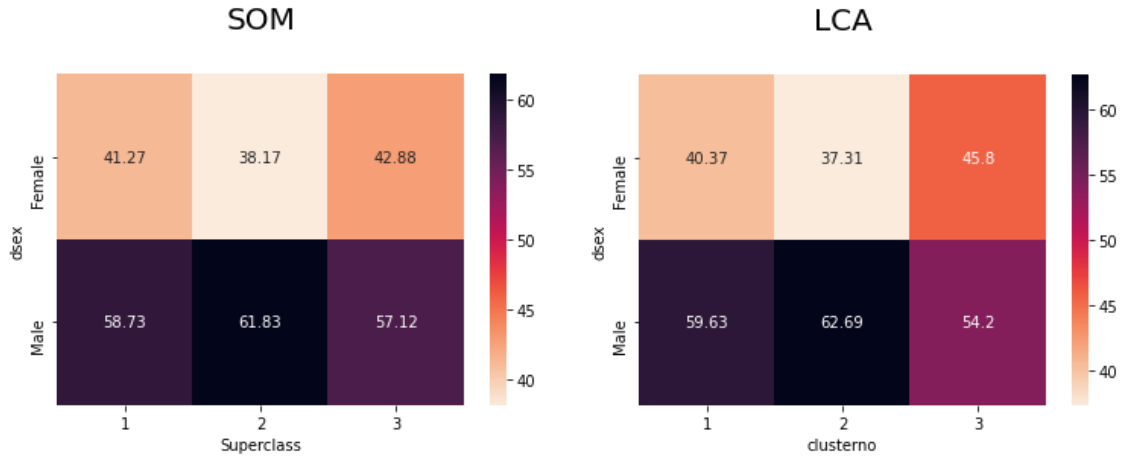

Figure 36: dsex variable cluster distribution between the SOM (left) and LCA (right) methods

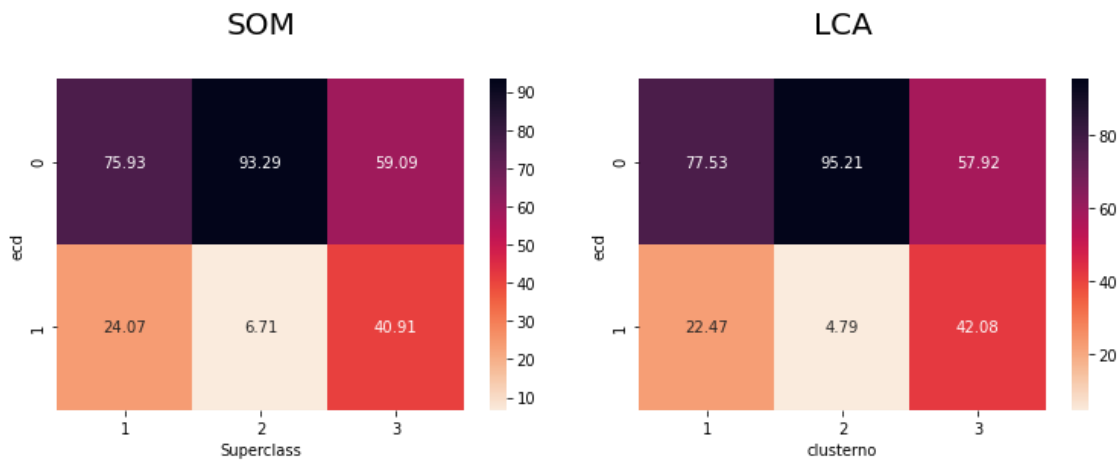

Figure 37: ecd variable cluster distribution between the SOM (left) and LCA (right) methods

We can represent the agreement between clusters obtained using the SOM and consensus clustering methods in the form of heatmaps. Figure 38 shows this agreement between the LCA and SOM methods.

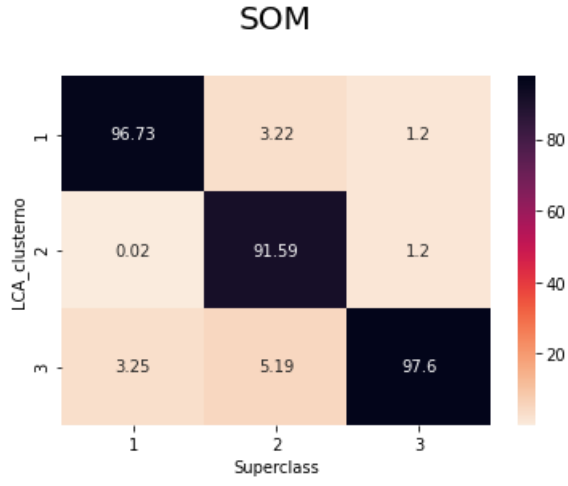

Figure 38: Agreement between clusters generated by the LCA and SOM methods

In the above figure, each cell represents how well clusters match or more specifically, how much of each cluster produced by SOM consists of samples from a cluster from the LCA method. For example, in Cluster 1 of SOM, about 96% of the data belongs to Cluster 1 of

the LCA consensus method, while only about 3% of the points belong to Cluster 3 of the LCA consensus method. Likewise, in Cluster 3 of SOM, about 97% of the data belongs to Cluster 3 of LCA while only about 1% of points belong to Cluster 1 or 2 of LCA. This tells us that the ensemble clustering methods produce results that are very similar to SOM's results. In comparison, Figure 39 shows the agreement between the KAMILA and SOM clusters with a weaker result or agreement.

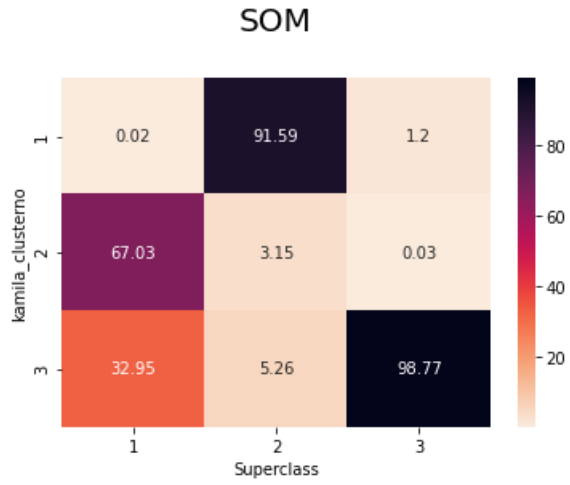

Figure 39: Agreement between clusters generated by the KAMILA and SOM methods

All the results mentioned in this section represent the close relationship between results produced by Self-Organizing Maps and the cluster ensemble of Latent Class Analysis (LCA). In that regard, SOMs are able to substantiate our consensus clustering results. Additionally, these results also represent how SOM could be used as an independent clustering approach in mixed-type data clustering scenarios and provide valuable results.

4.4 PHASE 4 Results: Visualization

Heatmaps and SOM visualizations have been shown previously. In addition to them, t-SNE is used to provide an overall view of the clusters obtained. Figure 40 shows the t-SNE visualization of the clusters produced by the Latent Class Analysis (LCA) consensus

clustering algorithm. This method produced the best clusters in terms of the t-SNE visualization, evaluation indices and agreement with SOM clusters.

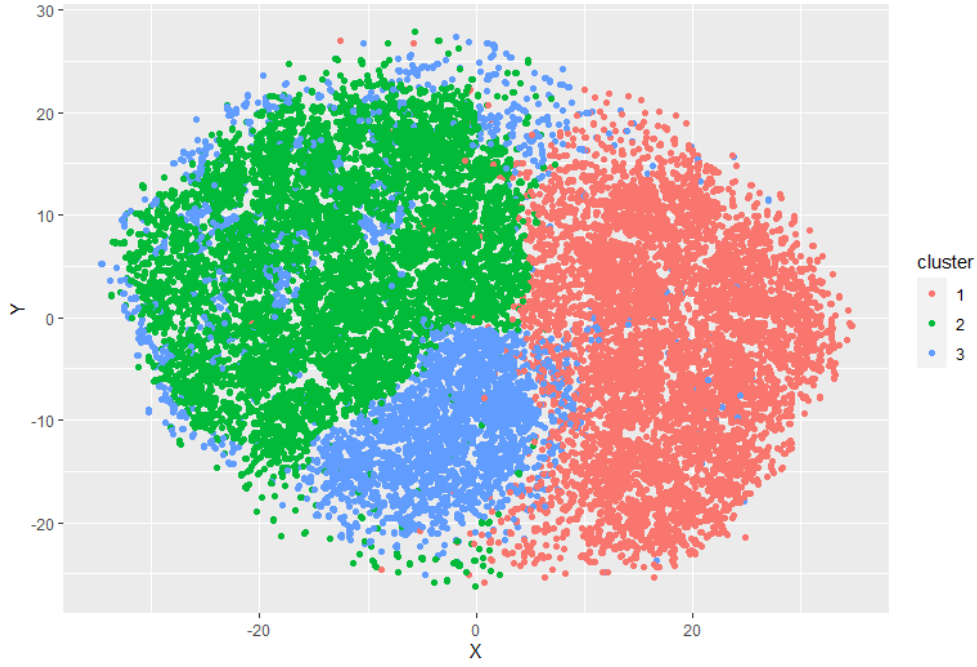

Figure 40: t-SNE representation of LCA clusters

The visualization for the SOM method is very similar to Latent Class Analysis (LCA) as shown in *Figure 41*.

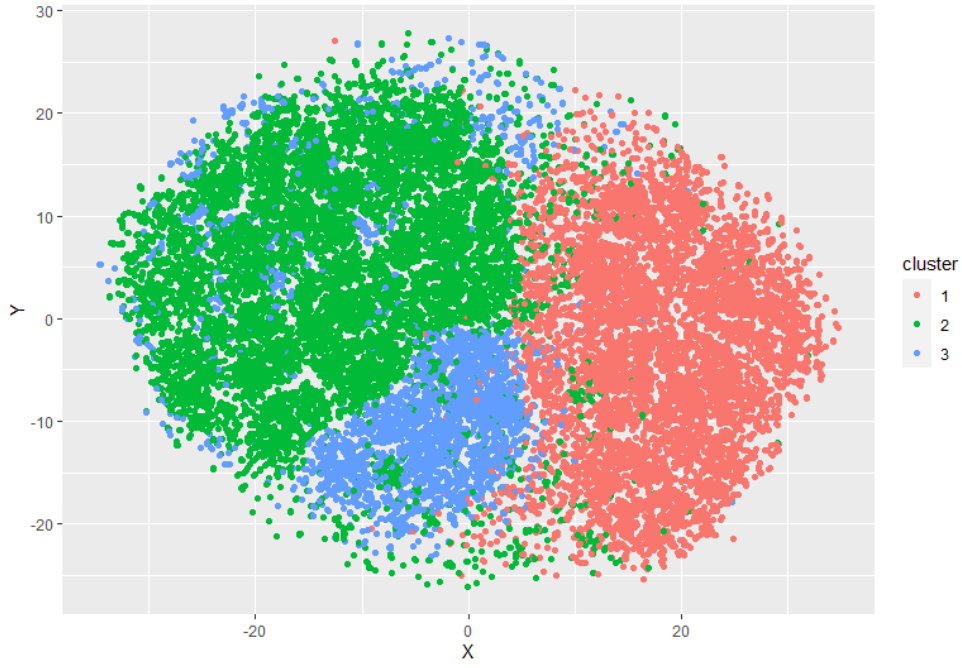

Figure 41: t-SNE representation of SOM clusters

Figure 42 and Figure 43 represent the t-SNE visualizations of the k-modes and majority voting consensus methods respectively. For k-modes, there is a more visible cluster overlap and randomness as opposed to LCA. In the majority voting visualization, we can observe a weaker overall cluster structure with significant dispersion and cluster overlap.

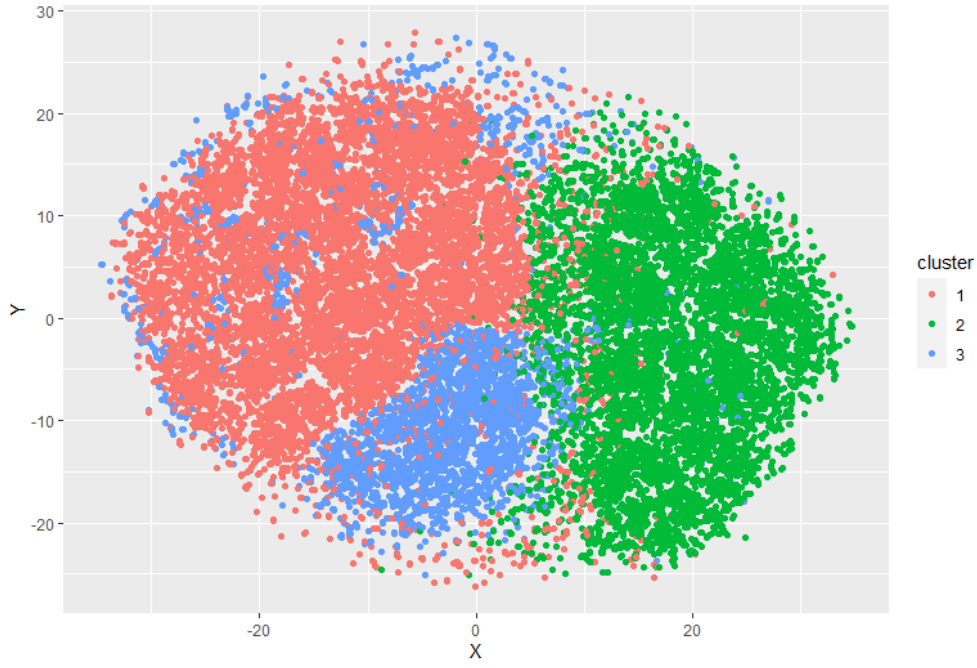

Figure 42: t-SNE representation of k-modes clusters

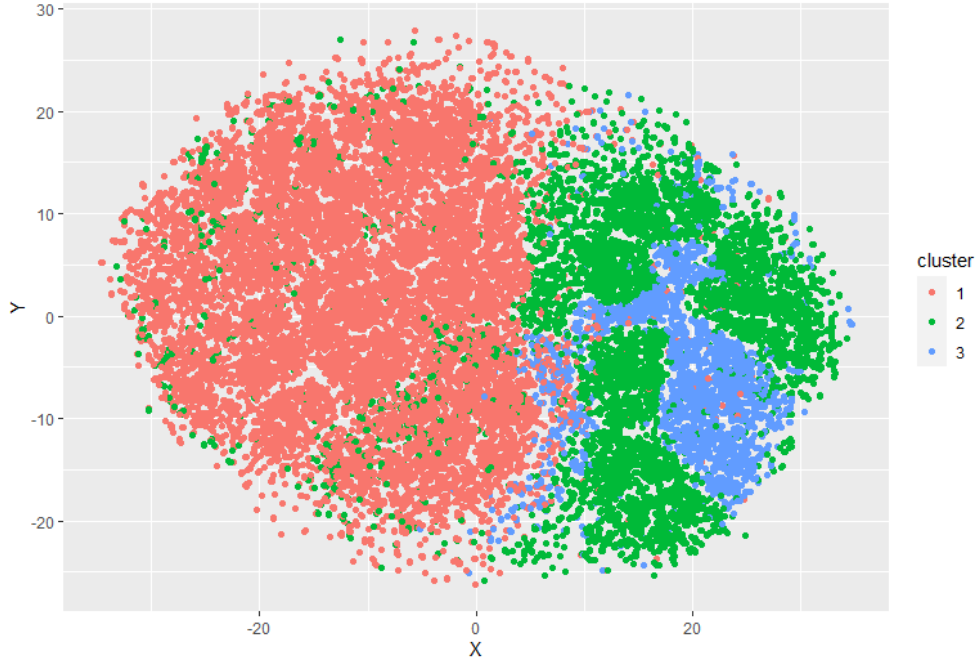

Figure 43: t-SNE representation of Majority Voting clusters

Figure 44 and Figure 45 show the t-SNE visualizations for the base clustering algorithms of KAMILA and mixture model respectively. There is significantly more visible cluster overlap for mixture model than any of the other methods.

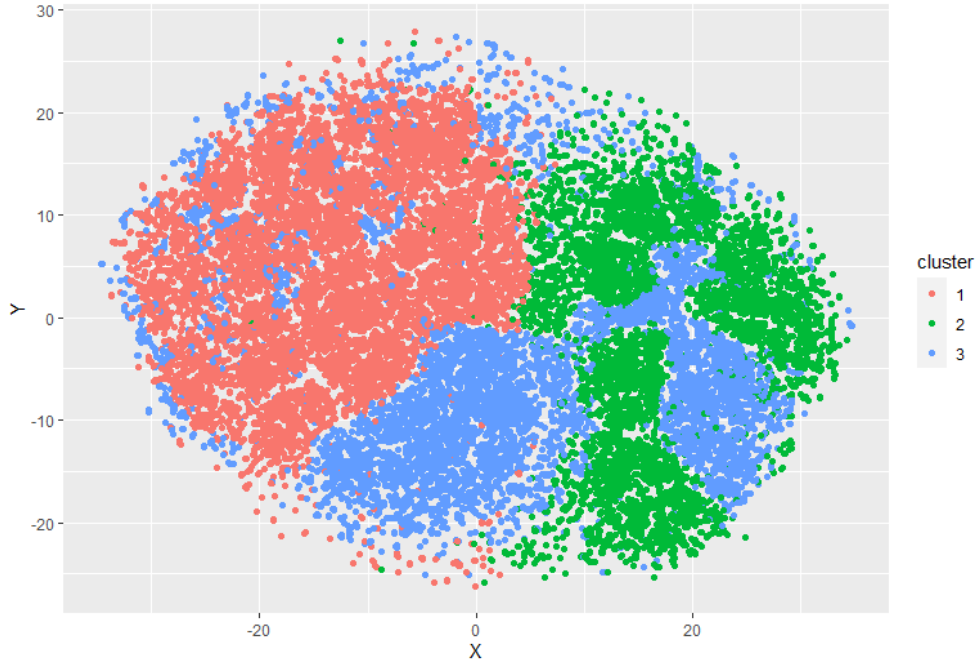

Figure 44: t-SNE representation of KAMILA clusters

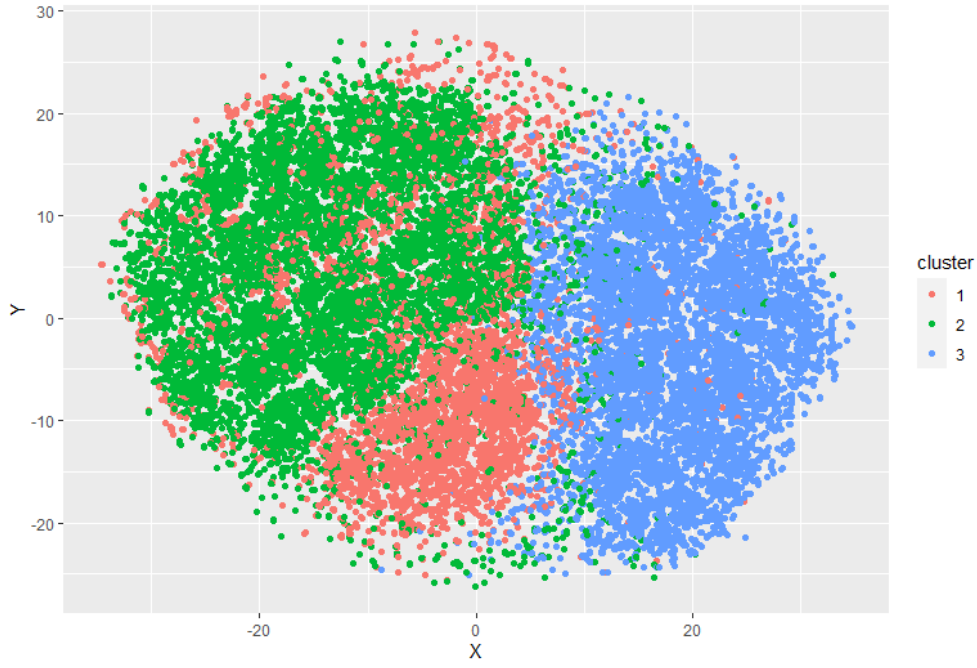

Figure 45: t-SNE representation of Mixture model clusters

We can visualize the cluster structures when $k > 3$ to observe how they shift with higher clusters. This is shown with the KAMILA algorithm in Figure 46 and Figure 47.

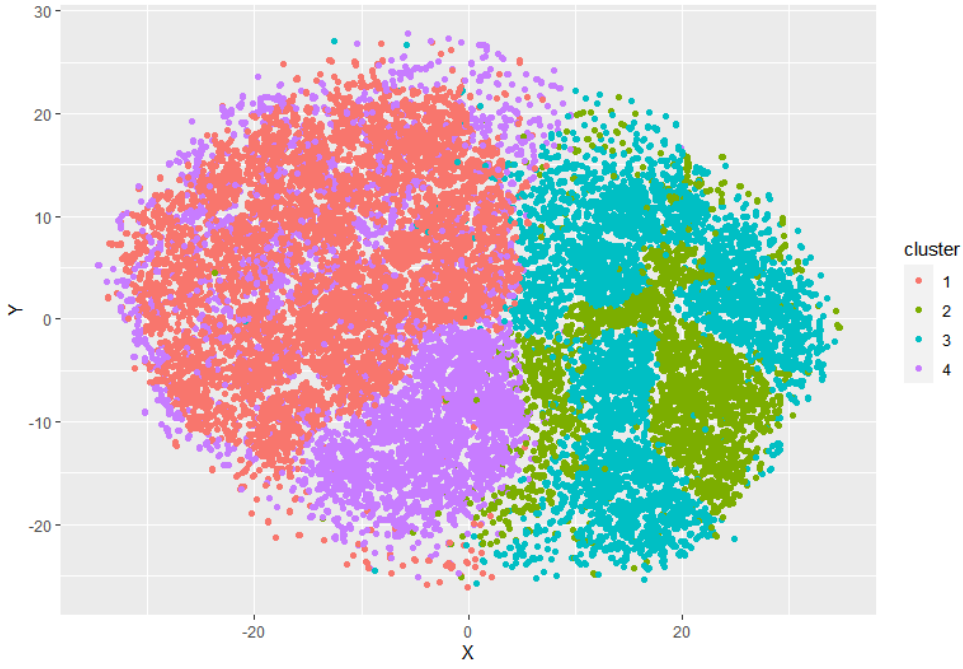

Figure 46: t-SNE visualization for KAMILA with 4 clusters

Figure 47: t-SNE visualization with KAMILA for 5 clusters

4.5 Cluster Interpretations to Derive Phenotypes

One of the primary goals of this work is to identify phenotypes among kidney transplant donors and recipients based on their clinical characteristics. With the use of heatmaps for categorical variables and statistical measures for numerical variables, we can make sense of the clusters being generated by the various clustering algorithms to identify phenotypes among our data. Some of the heatmaps and statistical measures have been shown previously throughout this section. Cluster descriptions based on all of the variable distributions are provided below to summarize the consensus clustering result findings. The cluster descriptions mentioned are based on the best-performing algorithm of Latent Class Analysis (LCA). These descriptions primarily highlight differentiating or unique features among clusters. Following that, we provide a detailed summary of numerical and categorical variable distributions in the clusters.

4.5.1 Cluster Descriptions

Cluster A

This is the biggest cluster. On average, this cluster has the youngest donors and recipients and the longest survival time for recipients post-transplant. There is also a slightly higher presence of female recipients contained within this cluster. The primary End-stage renal disease diagnosis is Glomerulonephritis (GN) and Hypertension. Most of the recipients here do not have diabetes, while most of the donors do not have hypertension. Donors mostly do not belong to the expanded criteria in this cluster. With regards to BMI, this cluster has the highest occurrence of recipients in the >18.49 - 24.99 category while having the lowest occurrence in the >29.99 - 34.99 and >34.99 categories. An identical trend is seen for donor BMIs as well.

Cluster B

This is the second-largest cluster. This group consists of the oldest recipients on average. It also has the least average number of years that the recipient spends on dialysis pre-transplant. The functional statuses between 40% and 70% seem to be most prevalent in this cluster compared to others. The majority of ESRD diagnoses here are Diabetes. Almost the entire population of recipients in this group have Diabetes. There is significantly more death occurring among the recipients. A substantial portion of the donors is from the expanded criteria. In comparison with other clusters, there is some presence of Peripheral Vascular Disease and also a higher presence of Coronary Artery disease. Unlike cluster A, there is a considerable portion of donors that have Hypertension. With respect to BMIs, recipients in the ranges of >29.99 - 34.99 and >34.99 occur more frequently in the population.

Cluster C

This is the smallest cluster among the three. The oldest donor population on average exists in this cluster while the average recipient age here also happens to be significantly higher than that of Cluster A. The recipients also happened to spend a greater number of years (on average) on dialysis pre-transplant. There is a slightly higher occurrence of female recipients here than in Cluster B and a higher occurrence of female donors than in other

clusters. The majority of the ESRD diagnoses here are Glomerulonephritis (GN) and Hypertension. This group has the highest presence of donors who identify from the black race. Most of the recipients here do not have diabetes. Donor CMV and donor Diabetes have a greater prevalence in the population here. This is also the cluster that has the greatest proportion of recipients that have suffered a graft loss. A substantial part of the donor population in this group belongs to the expanded criteria. Almost all the donors had hypertension. With regards to BMIs, donors in the ranges of >29.99 - 34.99 occur more frequently in this cluster.

Table 24 gives a detailed overview of the variable distributions among the different clusters produced from the Latent Class Analysis (LCA) consensus method. Categorical variables are represented by their frequencies (and %) in that cluster. Numerical variables are represented by their means and standard deviations among clusters (means ± s.d). The $p - values$ from the Kruskal-Wallis test for numerical variables and the chi-square test for categorical variables are also presented.

In all the statistical tests, the p-value was lesser than our chosen significance level of 0.05 for all the variables as shown in the table below. This indicates the strong association between the dependent variable (clusters) and all the independent variables (for categorical variables) as well as the significant difference among clusters for the numerical variables.

Table 24: Variable distributions among clusters. Categorical variables shown by count (% of cluster) and numerical variables shown by mean ± s.d. p-values of the two statistical tests are provided.

Features	**Clusters**			**p-value**
	A (n = 11,472)	**B (n = 9,583)**	**C (n = 4,769)**	
Functional status				
10% - moribund	25 (0.2)	23 (0.2)	7 (0.1)	<0.001
20% - very sick	105 (0.9)	92 (1)	39 (0.8)	
30% - severely disabled	57 (0.5)	46 (0.5)	16 (0.3)	
40% - disabled	217 (1.9)	223 (2.3)	79 (1.7)	

Features	Clusters			p-value
	A (n = 11,472)	B (n = 9,583)	C (n = 4,769)	
50% - req consid assist	186 (1.6)	347 (3.6)	86 (1.8)	
60% - req assist	688 (6.0)	855 (8.9)	310 (6.5)	
70% - unable to do normal activity	1930 (16.8)	1919 (20.0)	829 (17.4)	
80% - some sx	3720 (32.4)	2974 (31.0)	1488 (31.2)	
90% - minor sx	2849 (24.8)	1974 (20.6)	1220 (25.6)	
100% - no complaints	1695 (14.8)	1130 (11.8)	695 (14.6)	
Peak PRA group				
0	7747 (67.5)	6825 (71.2)	3382 (70.9)	<0.001
1	2533 (22.1)	1945 (20.3)	1029 (21.6)	
2	1192 (10.4)	813 (8.5)	358 (7.5)	
Recipient sex				
Female	4949 (43.1)	3357 (35.0)	1940 (40.7)	<0.001
Male	6523 (56.9)	6226 (65.0)	2829 (59.3)	
Donor sex				
Female	4280 (37.3)	3869 (40.4)	2184 (45.8)	<0.001
Male	7192 (62.7)	5714 (59.6)	2585 (54.2)	
Recipient race				
Black	3790 (33.0)	3283 (34.3)	1682 (35.3)	<0.001
Other	886 (7.7)	863 (9.0)	333 (7.0)	
White	6796 (59.2)	5437 (56.7)	2754 (57.7)	
Donor race				
Black	1469 (12.8)	1344 (14.0)	981 (20.6)	<0.001
Other	322 (2.8)	341 (3.6)	176 (3.7)	
White	9681 (84.4)	7898 (82.4)	3612 (75.7)	
ESRD Diagnosis				

Features	Clusters			p-value
	A (n = 11,472)	B (n = 9,583)	C (n = 4,769)	
Diabetes	0 (0.0)	7771 (81.1)	26 (0.5)	<0.001
GN	3532 (30.8)	321 (3.3)	1201 (25.2)	
HTN	4279 (37.3)	1019 (10.6)	2135 (44.8)	
Other	2074 (18.1)	378 (3.9)	721 (15.1)	
PCKD	1587 (13.8)	94 (1.0)	686 (14.4)	
Transplant type				
Left kidney	5333 (46.5)	4713 (49.2)	2364 (49.6)	<0.001
Right kidney	6139 (53.5)	4870 (50.8)	2405 (50.4)	
Recipient Diabetes				
DM	0 (0.0)	9580 (100.0)	345 (7.2)	<0.001
No DM	11472 (100.0)	3 (0.0)	4424 (92.8)	
Donor Diabetes				
Negative	11114 (96.9)	8862 (92.5)	3853 (80.8)	<0.001
Positive	358 (3.1)	721 (7.5)	916 (19.2)	
HLA mismatch				
0	680 (5.9)	575 (6.0)	181 (3.8)	<0.001
1	160 (1.4)	104 (1.1)	35 (0.7)	
2	464 (4.0)	359 (3.7)	152 (3.2)	
3	1543 (13.5)	1149 (12.0)	581 (12.2)	
4	3128 (27.3)	2528 (26.4)	1286 (27.0)	
5	3676 (32.0)	3235 (33.8)	1632 (34.2)	
6	1821 (15.9)	1633 (17.0)	902 (18.9)	
Recipient CMV				
Negative	3782 (33.0)	2698 (28.2)	1493 (31.3)	<0.001
Positive	7690 (67.0)	6885 (71.8)	3276 (68.7)	
Donor CMV				
Negative	4405 (38.4)	3423 (35.7)	1493 (31.3)	<0.001
Positive	7067 (61.6)	6160 (64.3)	3276 (68.7)	

Features	Clusters			p-value
	A (n = 11,472)	B (n = 9,583)	C (n = 4,769)	
Donor Hepatitis C Virus				
Negative	11194 (97.6)	9231 (96.3)	4686 (98.3)	<0.001
Positive	278 (2.4)	352 (3.7)	83 (1.7)	
Donation after Cardiac death				
No	9700 (84.6)	8176 (85.3)	4234 (88.8)	<0.001
Yes	1772 (15.4)	1407 (14.7)	535 (11.2)	
Event				
Censored	8406 (73.3)	5412 (56.5)	2900 (60.8)	<0.001
Death	1395 (12.2)	2651 (27.7)	866 (18.2)	
Graft loss	1671 (14.6)	1520 (15.9)	1003 (21.0)	
Graft failure				
Censored	8406 (73.3)	5412 (56.5)	2900 (60.8)	<0.001
Yes	3066 (26.7)	4171 (43.5)	1869 (39.2)	
Death				
Censored	9647 (84.1)	6190 (64.6)	3541 (74.3)	<0.001
Yes	1825 (15.9)	3393 (35.4)	1228 (25.7)	
Expanded criteria donor				
No	10923 (95.2)	7430 (77.5)	2762 (57.9)	<0.001
Yes	549 (4.8)	2153 (22.5)	2007 (42.1)	
Recipient hypertension				
No	1463 (12.8)	840 (8.8)	509 (10.7)	<0.001
Yes	10009 (87.2)	8743 (91.2)	4260 (89.3)	
Recipient Cardiovascular disease				
No	11210 (97.7)	9094 (94.9)	4643 (97.4)	<0.001
Yes	262 (2.3)	489 (5.1)	126 (2.6)	
Recipient Peripheral Vascular disease				
No	11288 (98.4)	8722 (91.0)	4661 (97.7)	<0.001
Yes	184 (1.6)	861 (9.0)	108 (2.3)	

Features	Clusters			p-value
	A (n = 11,472)	B (n = 9,583)	C (n = 4,769)	
Recipient malignancy				
No	10795 (94.1)	9064 (94.6)	4396 (92.2)	<0.001
Yes	677 (5.9)	519 (5.4)	373 (7.8)	
Donor Hypertension				
No	11462 (99.9)	6601 (68.9)	4 (0.1)	<0.001
Yes	10 (0.1)	2982 (31.1)	4765 (99.9)	
Pre-emptive transplant				
No	9952 (86.8)	8695 (90.7)	4211 (88.3)	<0.001
Yes	1520 (13.2)	888 (9.3)	558 (11.7)	
Recipient Coronary Artery disease				
CAD	659 (5.7)	1391 (14.5)	316 (6.6)	<0.001
No CAD	10813 (94.3)	8192 (85.5)	4453 (93.4)	
Recipient BMI				
<=18.49	325 (2.8)	90 (0.9)	77 (1.6)	<0.001
>18.49-24.99	3827 (33.4)	1682 (17.6)	1440 (30.2)	
>24.99-29.99	3784 (33.0)	3235 (33.8)	1707 (35.8)	
>29.99-34.99	2379 (20.7)	2878 (30.0)	1017 (21.3)	
>34.99	1157 (10.1)	1698 (17.7)	528 (11.1)	
Donor BMI				
<=18.49	770 (6.7)	366 (3.8)	62 (1.3)	<0.001
>18.49-24.99	4571 (39.8)	3263 (34.0)	1029 (21.6)	
>24.99-29.99	3535 (30.8)	3134 (32.7)	1479 (31.0)	
>29.99-34.99	1643 (14.3)	1652 (17.2)	1165 (24.4)	
>34.99	953 (8.3)	1168 (12.2)	1034 (21.7)	
Numerical Variables				
Cold ischemia time (hrs)	16.96 ± 9.88	17.13 ± 8.96	18.16 ± 10.49	<0.001
Donor age (yrs)	34.12 ± 15.17	41.33 ± 15.83	49.69 ± 10.23	<0.001

Features	Clusters			p-value
	A (n = 11,472)	B (n = 9,583)	C (n = 4,769)	
Recipient age at transplant (yrs)	49.93 ± 13.61	58.28 ± 9.50	55.35 ± 12.98	<0.001
Survival Time (yrs)	5.29 ± 2.13	4.86 ± 2.28	4.88 ± 2.33	<0.001
Survival Time (days)	1932.37 ± 781.54	1777.70 ± 833.44	1784.62 ± 853.88	<0.001
Time on dialysis pre-transplant (yrs)	3.54 ± 3.52	3.34 ± 2.69	3.71 ± 3.69	0.015

4.6 Discussion

The primary motivation of our work is to use unsupervised learning towards recognizing phenotypes that may exist among kidney transplant donors and recipients. To that goal, we employed individual clustering and consensus clustering algorithms. Additionally, Self-Organizing Maps was used to produce an alternate solution to support our results. Cluster ensembles can produce more robust clusters by combining the results from the individual clustering algorithms that operate in different ways.

In our results, we first discussed the outcome of imputation using MICE and the diagnostics involved. Kolmogorov-Smirnov (KS) statistical test was used in the imputation process to check the quality of the numerical variable imputations. We also briefly compared numerical variable imputation methods. Following that, the base clusterings and consensus clustering results are presented in addition to a comparison among them. Between the two base clustering algorithms of mixture model and KAMILA, KAMILA performed better with scores of 0.103, 0.0536 and 2760 for the Silhouette, Dunn index and Calinski-Harabasz metrics respectively. These model-based methods also produced better results than traditional partitional and hierarchical clustering methods like k-medoids, k-prototypes and hierarchical clustering (with different linkages) which can be limited by large datasets and parameter selection. There is the added issue of identifying an appropriate mixed-type distance measure for some of those algorithms [39]. As

mentioned by Hunt and Jorgensen in their work, some of these traditional clustering algorithms might not reflect randomness present in the data and also can lead to significantly varying results with small changes in the data [42]. Three different consensus methods of k-modes, Majority voting and Latent Class Analysis (LCA) were tested and evaluated. The Latent Class Analysis (LCA) consensus method produced the best results for the evaluation metrics, visualizations and distinction between cluster features among consensus and base clustering algorithms. The Silhouette, Dunn index and Calinski-Harabasz scores for the LCA method were 0.113, 0.0538 and 2843 respectively. Numerical variables represented by means and standard deviations and categorical variables represented by heatmaps are presented for the base clustering and consensus clustering results. They provide further insight into the clusters obtained. The alternate results produced by Self-Organizing Maps are discussed. Self-Organizing Maps also produces its own unique visualization for our data. This method generates clusters that are identical to those of LCA thus supporting our consensus clustering results. t-SNE visualizations of all the methods involved are presented and a close resemblance between SOM and LCA's clusters can be observed in them. They also are shown to have the best cluster structures among the methods.

Finally, the cluster interpretations from the LCA method in the form of cluster descriptions and a detailed summary of the variable distributions among clusters are discussed. Additionally, results from statistical tests to further validate our results have been presented. We were able to identify features, numerical and categorical that are clearly distinguishable among clusters. For numerical variables, donor age (dage), recipient age (ragetx) and survival time (survtime3) have particularly varying statistics among the clusters obtained. Several categorical variables have clear distinctions between clusters. This is true, especially for the outcome variables of event (event), graft failure (graftfailure), death (death) and expanded donor criteria (ecd). Among donor-related categorical variables, donor Hypertension (dhtn2), donor Diabetes (ddm), donor sex (dsex) and donor BMI (dbmisimp) are some variables that had a visible distinction among clusters. For recipient-related categorical variables, that was seen in ESRD Diagnosis (esrddxsimp), recipient Diabetes (rdm2), recipient sex (rsex), recipient BMI (rbmisimp), recipient Coronary Artery disease (rcad) variables.

To conclude the discussion above, the final cluster descriptions and summaries are based on the results of the Latent Class Analysis (LCA) consensus method. We were able to compare solutions from the individual and consensus clustering approaches using various measures. We used an additional solution from Self-Organizing Maps to support our results. This also strengthens the position of using SOMs as an independent clustering approach for mixed-type data and attain valuable results. Our solution shows clear distinctions for some of the donor and recipient characteristics among clusters. This is presented with numerical statistics, heatmaps and overall cluster distributions.

CHAPTER 5 CONCLUSION

5.1 Summary

Predicting kidney transplant survival and identifying phenotypes among donors and recipients is crucial for those involved in kidney transplantation. National kidney transplant data is an extremely rich source of information that can be leveraged to provide insights concerning the individuals involved. Being able to understand the patterns that may reside within this data has a massive potential to help experts in their decision-making. It could help them identify phenotypes and hence predict graft survival, design more specialized treatment plans, understand the possibilities of various outcomes taking place and recognize trends among these groups. In this work, we utilized a dataset from the Scientific Registry of Transplant Recipients (SRTR) that had 165,090 records spanning the years of 2000 - 2017. We performed significant data pre-processing followed by a comprehensive imputation process that consisted of two separate tasks. Multivariate Imputation by Chained Equations (MICE) is a powerful method of imputation that allowed us to simultaneously impute categorical and numerical variables using methods best suited for them. The cohort of individuals from the years of 2009 - 2011 (3 years) was prepared from the dataset for further analysis. Two different model-based clustering methods, KAMILA and Mixture model, capable of handling a mixed-type dataset, were applied to this cohort and gave us the base clustering solutions. Cluster ensemble methods that produced a consensus result from the base clusterings were employed in this work. The three consensus clustering methods were k-modes, Majority Voting and Latent Class Analysis (LCA). Simultaneously, as part of our methodology, Self-Organizing Maps with hierarchical clustering was used to support the results obtained from the cluster ensemble methods. Clusters identified through both approaches (consensus clustering and SOM) were very good and insightful. In addition to the evaluation indices of Silhouette, Dunn and Calinski-Harabasz, heatmaps and numerical statistics were generated to compare and evaluate the results obtained from the various methods. t-SNE visualizations were produced and compared among methods to give an idea of the overall cluster structures. Using t-SNE with Gower's distance allowed us to represent both categorical and numerical variables in the same space. The Latent Class Analysis (LCA) ensemble produced the best

results among all the methods (individual and consensus). This method also had the best agreement with the alternate solution produced by Self-Organizing Maps. By summarizing the information from the heatmaps and numerical statistics, three different potential groups were recognized, and their descriptions were generated. We were able to identify distinctions between clusters for the outcome variables of event, death, graft failure and expanded donor criteria in addition to several other non-outcome variables. These cluster interpretations can be used towards deriving phenotypes.

5.2 Limitations

One of the primary limitations in our work is caused by the data consisting of mixed-type attributes. This severely limited the number of clustering algorithms available at our disposal. The imputation process became much more extensive and complex. Current research in the area of mixed-type clustering is not as comprehensive as it is in the case of either numerical or categorical data clustering. Due to the nature of the data, it was challenging to identify evaluation techniques and indices that would otherwise be readily usable. Prior work involving mixed data clustering algorithms used popular datasets that had a ground truth or labels which allowed for some of the common evaluation metrics to be used. These previous works regularly focused on the same popular datasets and evaluated algorithms using them. The survey paper by Ahmad and Khan also identified this problem of the lack of use of performance metrics not based on labels or a ground truth [39]. Another major limitation is that we had to constantly move back and forth between using R and python due to the lack of packages available for mixed-data clustering and analysis. This increased the overall time required to perform experiments and made it a more complex and tedious process. With regard to the evaluation metrics used in this work, we initially tried to use another distance metric that seemed to be more insightful. However, due to the lack of available packages to compute this, Gower's distance was used. The alternate distance metric took in more information in its computation, rather than the simple 1's and 0's that Gower's distance uses when comparing categorical values. It considered the co-occurrence of variables when generating the distance and automatically

provided weights for the variables involved. This was proposed by Ahmad and Dey with an emphasis on the k-mean clustering algorithm in their work [111].

Due to the large size of the dataset and the type of data, parts of our methodology like imputation were computationally expensive. When experimenting with different methods and approaches, there were instances where an algorithm wouldn't compute or take too long. This was the case when we tried using the LCE consensus method in our work from the diceR package [94], [104]. We had found a package for the alternate distance metric that we tried to implement but failed to make it work for our data. This was the DisimForMixed package in R [112]. There was also the issue of massive storage space being consumed by some of the distance calculations due to the size of the dataset. Although we did try various parameters for some of the algorithms involved, it was a challenging problem due to the large number of methods and parameters. This can be explored further in future work.

With regard to visualization, it was difficult to identify methods that could represent both categorical and numerical variables in the same space without any discretization or encoding that may have resulted in a loss of information.

5.3 Future Work

Our research opens a lot of avenues for future work. We can try and evaluate other types of mixed-type data clustering algorithms. We can look into variable importance among clusters generated. Additional cluster ensemble generation techniques and methods can be explored and compared to include greater diversity and information. Visualization techniques capable of handling mixed-type data could be further researched or developed to provide alternative representations. Identifying diagnostic measures for categorical variable imputation could be a valuable addition to the imputation process. We can experiment with larger datasets or more variables to identify these phenotypes. Further hyperparameter tuning of the algorithms involved can be experimented with, to potentially improve clustering performances. SOMs which are shown to be an efficient and valuable

clustering approach, can be further explored independently and not just as a supporting method.

Unlike traditional clustering problems that focus on either numerical or categorical data, there is a lot of work that can be done in the area of mixed-type data clustering, especially without labels or a ground truth. Considering a lot of the real-world data, especially in medicine and healthcare, consists of mixed-type data, it is extremely important to work on this. Our research can be used to aid further work by researchers and nephrologists in the domain of kidney transplantation.

5.4 Disclaimer

The data reported here have been supplied by the Hennepin Healthcare Research Institute as the contractor for the SRTR. The interpretation and reporting of these data are the responsibility of the authors and in no way should be seen as an official policy of or interpretation by the SRTR or the US Government.

REFERENCES

[1] K. U. Eckardt, J. Coresh, O. Devuyst, R. J. Johnson, A. Köttgen, A. S. Levey, and A. Levin, "Evolving importance of kidney disease: from subspecialty to global health burden," *Lancet (London, England)*, vol. 382, no. 9887, pp. 158–169, 2013, doi: 10.1016/S0140-6736(13)60439-0.

[2] M. A. Abbasi, G. M. Chertow, and Y. N. Hall, "End-stage renal disease.," *BMJ Clin. Evid.*, vol. 2010, Jul. 2010.

[3] M. Abecassis, S. T. Bartlett, A. J. Collins, C. L. Davis, F. L. Delmonico, J. J. Friedewald, R. Hays, A. Howard, E. Jones, A. B. Leichtman, R. M. Merion, R. A. Metzger, F. Pradel, E. J. Schweitzer, R. L. Velez, and R. S. Gaston, "Kidney Transplantation as Primary Therapy for End-Stage Renal Disease: A National Kidney Foundation/Kidney Disease Outcomes Quality Initiative (NKF/KDOQI^{TM}) Conference," *Clin. J. Am. Soc. Nephrol.*, vol. 3, no. 2, p. 471, Mar. 2008, doi: 10.2215/CJN.05021107.

[4] G. G. Garcia, P. Harden, and J. Chapman, "The Global Role of Kidney Transplantation," *Kidney Blood Press. Res.*, vol. 35, no. 5, pp. 299–304, Jun. 2012, doi: 10.1159/000337044.

[5] M. Tonelli, N. Wiebe, G. Knoll, A. Bello, S. Browne, D. Jadhav, S. Klarenbach, and J. Gill, "Systematic Review: Kidney Transplantation Compared With Dialysis in Clinically Relevant Outcomes," *Am. J. Transplant.*, vol. 11, no. 10, pp. 2093–2109, Oct. 2011, doi: 10.1111/J.1600-6143.2011.03686.X.

[6] C. Legendre, G. Canaud, and F. Martinez, "Factors influencing long-term outcome after kidney transplantation," *Transpl. Int.*, vol. 27, no. 1, pp. 19–27, Jan. 2014, doi: 10.1111/TRI.12217.

[7] Z. M. Soler, J. M. Hyer, V. Ramakrishnan, T. L. Smith, J. Mace, L. Rudmik, and R. J. Schlosser, "Identification of chronic rhinosinusitis phenotypes using cluster analysis," *Int. Forum Allergy \& Rhinol.*, vol. 5, no. 5, pp. 399–407, 2015, doi: https://doi.org/10.1002/alr.21496.

[8] W. C. Moore, D. A. Meyers, S. E. Wenzel, W. G. Teague, H. Li, X. Li, R. D'Agostino, M. Castro, D. Curran-Everett, A. M. Fitzpatrick, B. Gaston, N. N. Jarjour, R. Sorkness, W. J. Calhoun, K. F. Chung, S. A. A. Comhair, R. A. Dweik, E. Israel, S. P. Peters, W. W. Busse, S. C. Erzurum, and E. R. Bleecker, "Identification of Asthma Phenotypes Using Cluster Analysis in the Severe Asthma Research Program," *Am. J. Respir. Crit. Care Med.*, vol. 181, no. 4, pp. 315–323, 2010, doi: 10.1164/rccm.200906-0896OC.

[9] A. Tariq, P. M. J., S. P. J., O. Emily, W. D. J., P. I. L., K. D. W., L. K. L., O. C. M., and F. G. Michael, "Clinical Implications of Chronic Heart Failure Phenotypes Defined by Cluster Analysis," *J. Am. Coll. Cardiol.*, vol. 64, no. 17, pp. 1765–1774, Oct. 2014, doi: 10.1016/j.jacc.2014.07.979.

[10] Y. Horiuchi, S. Tanimoto, A. H. M. M. Latif, K. Y. Urayama, J. Aoki, K. Yahagi, T. Okuno, Y. Sato, T. Tanaka, K. Koseki, K. Komiyama, H. Nakajima, K. Hara, and K. Tanabe, "Identifying novel phenotypes of acute heart failure using cluster analysis of clinical variables," *Int. J. Cardiol.*, vol. 262, pp. 57–63, 2018, doi: https://doi.org/10.1016/j.ijcard.2018.03.098.

[11] S. Bailly, L. Grote, J. Hedner, S. Schiza, W. T. McNicholas, O. K. Basoglu, C. Lombardi, Z. Dogas, G. Roisman, A. Pataka, M. R. Bonsignore, J.-L. Pepin, and E. S. Group, "Clusters of sleep apnoea phenotypes: A large pan-European study from the European Sleep Apnoea Database (ESADA)," *Respirology*, vol. 26, no. 4, pp. 378–387, Apr. 2021, doi: https://doi.org/10.1111/resp.13969.

[12] A. Joshi, A. Gangopadhyay, M. Banerjee, G. Baffoe-Bonnie, V. Mohanlal, and R. Wali, "A clustering method to study the loss of kidney function following kidney transplantation," *Int. J. Biomed. Eng. Technol.*, vol. 3, no. 1–2, pp. 64–82, 2010, doi: 10.1504/IJBET.2010.029652.

[13] T. Vaulet, G. Divard, O. Thaunat, E. Lerut, A. Senev, O. Aubert, E. Van Loon, J. Callemeyn, M.-P. Emonds, A. Van Craenenbroeck, K. De Vusser, B. Sprangers, M. Rabeyrin, V. Dubois, D. Kuypers, M. De Vos, A. Loupy, B. De Moor, and M. Naesens, "Data-driven Derivation and Validation of Novel Phenotypes for Acute Kidney Transplant Rejection using Semi-supervised Clustering," *J. Am. Soc. Nephrol.*, vol. 32, no. 5, pp. 1084–1096, 2021, doi: 10.1681/ASN.2020101418.

[14] C. Thongprayoon, P. Vaitla, C. C. Jadlowiec, N. Leeaphorn, S. A. Mao, M. A. Mao, P. Pattharanitima, J. Bruminhent, N. J. Khoury, V. D. Garovic, M. Cooper, and W. Cheungpasitporn, "Use of Machine Learning Consensus Clustering to Identify Distinct Subtypes of Black Kidney Transplant Recipients and Associated Outcomes," *JAMA Surg.*, vol. 157, no. 7, p. E221286, Jul. 2022, doi: 10.1001/JAMASURG.2022.1286.

[15] C. Thongprayoon, S. A. Mao, C. C. Jadlowiec, M. A. Mao, N. Leeaphorn, W. Kaewput, P. Vaitla, P. Pattharanitima, S. Tangpanithandee, P. Krisanapan, F. Qureshi, P. Nissaisorakarn, M. Cooper, and W. Cheungpasitporn, "Machine Learning Consensus Clustering of Morbidly Obese Kidney Transplant Recipients in the United States," *J. Clin. Med.*, vol. 11, no. 12, p. 3288, Jun. 2022, doi: 10.3390/JCM11123288.

[16] A. Gangopadhyay, A. Joshi, and R. Wali, "A spectral clustering technique for studying post-transplant kidney functions," *IHI'12 - Proc. 2nd ACM SIGHIT Int. Heal. Informatics Symp.*, pp. 201–208, 2012, doi: 10.1145/2110363.2110388.

[17] C. Villeneuve, M.-L. Laroche, M. Essig, P. Merville, N. Kamar, A. Coubret, I. Lacroix, S. Bouchet, D. Fruit, P. Marquet, A. Rousseau, and on behalf of the E. study group, "Evolution and Determinants of Health-Related Quality-of-Life in Kidney Transplant Patients Over the First 3 Years After Transplantation," *Transplantation*, vol. 100, no. 3, 2016, [Online]. Available: https://journals.lww.com/transplantjournal/Fulltext/2016/03000/Evolution_and_Determinants_of_Health_Related.31.aspx

[18] S. A. A. Naqvi, K. Tennankore, A. Vinson, P. C. Roy, and S. S. R. Abidi, "Predicting Kidney Graft Survival Using Machine Learning Methods: Prediction Model Development and Feature Significance Analysis Study," *J. Med. Internet Res.*, vol. 23, no. 8, p. e26843, Aug. 2021, doi: 10.2196/26843.

[19] F.-X. Paquette, A. Ghassemi, O. Bukhtiyarova, M. Cisse, N. Gagnon, A. Della Vecchia, H. A. Rabearivelo, and Y. Loudiyi, "Machine Learning Support for Decision-Making in Kidney Transplantation: Step-by-step Development of a Technological Solution," *JMIR Med. Informatics*, vol. 10, no. 6, p. e34554, Jun. 2022, doi: 10.2196/34554.

[20] K. D. Yoo, J. Noh, H. Lee, D. K. Kim, C. S. Lim, Y. H. Kim, J. P. Lee, G. Kim, and Y. S. Kim, "A Machine Learning Approach Using Survival Statistics to Predict Graft Survival in Kidney Transplant Recipients: A Multicenter Cohort Study," *Sci. Reports 2017 71*, vol. 7, no. 1, pp. 1–12, Aug. 2017, doi: 10.1038/s41598-017-08008-8.

[21] A. Decruyenaere, P. Decruyenaere, P. Peeters, F. Vermassen, T. Dhaene, and I. Couckuyt, "Prediction of delayed graft function after kidney transplantation: comparison between logistic regression and machine learning methods," *BMC Med. Inform. Decis. Mak.*, vol. 15, no. 1, p. 83, 2015, doi: 10.1186/s12911-015-0206-y.

[22] E. Mark, D. Goldsman, B. Gurbaxani, P. Keskinocak, and J. Sokol, "Using machine learning and an ensemble of methods to predict kidney transplant survival," *PLoS One*, vol. 14, no. 1, pp. 1–13, 2019, doi: 10.1371/journal.pone.0209068.

[23] B. Peng, H. Gong, H. Tian, Q. Zhuang, J. Li, K. Cheng, and Y. Ming, "The study of the association between immune monitoring and pneumonia in kidney transplant recipients through machine learning models," *J. Transl. Med.*, vol. 18, no. 1, p. 370, 2020, doi: 10.1186/s12967-020-02542-2.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[25] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, Jun. 2001, doi: 10.1093/BIOINFORMATICS/17.6.520.

[26] S. F. Buck, "A Method of Estimation of Missing Values in Multivariate Data Suitable for Use with an Electronic Computer," *J. R. Stat. Soc. Ser. B*, vol. 22, no. 2, pp. 302–306, Jul. 1960, doi: 10.1111/J.2517-6161.1960.TB00375.X.

[27] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate Imputation by Chained Equations in R," *J. Stat. Softw.*, vol. 45, no. 3, pp. 1–67, Dec. 2011, doi: 10.18637/JSS.V045.I03.

[28] C. A. Bernaards, T. R. Belin, and J. L. Schafer, "Robustness of a multivariate normal approximation for imputation of incomplete binary data," *Stat. Med.*, vol. 26, no. 6, pp. 1368–1382, Mar. 2007, doi: 10.1002/SIM.2619.

[29] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?," *Int. J. Methods Psychiatr. Res.*, vol. 20, no. 1, p. 40, Mar. 2011, doi: 10.1002/MPR.329.

[30] Y. Liu and A. De, "Multiple Imputation by Fully Conditional Specification for Dealing with Missing Data in a Large Epidemiologic Study," *Int. J. Stat. Med. Res.*, vol. 4, no. 3, pp. 287–295, Aug. 2015, doi: 10.6000/1929-6029.2015.04.03.7.

[31] J. W. Graham, "Missing Data Analysis: Making It Work in the Real World," *http://dx.doi.org.ezproxy.library.dal.ca/10.1146/annurev.psych.58.110405.085530*, vol. 60, pp. 549–576, Nov. 2008, doi: 10.1146/ANNUREV.PSYCH.58.110405.085530.

[32] R. A. Schnoll, M. Rukstalis, E. P. Wileyto, and A. E. Shields, “Smoking Cessation Treatment by Primary Care Physicians: An Update and Call for Training,” *Am. J. Prev. Med.*, vol. 31, no. 3, pp. 233–239, Sep. 2006, doi: 10.1016/J.AMEPRE.2006.05.001.

[33] A. Kekitiinwa, K. J. Lee, A. S. Walker, A. Maganda, K. Doerholt, S. B. Kitaka, A. Asiimwe, A. Judd, P. Musoke, and D. M. Gibb, “Differences in factors associated with initial growth, CD4, and viral load responses to ART in HIV-infected children in Kampala, Uganda, and the United Kingdom/Ireland,” *J. Acquir. Immune Defic. Syndr.*, vol. 49, no. 4, pp. 384–392, Dec. 2008, doi: 10.1097/QAI.0B013E31818CDEF5.

[34] X. Basagaña, J. Barrera-Gómez, M. Benet, J. M. Antó, and J. Garcia-Aymerich, “A Framework for Multiple Imputation in Cluster Analysis,” *Am. J. Epidemiol.*, vol. 177, no. 7, pp. 718–725, Apr. 2013, doi: 10.1093/AJE/KWS289.

[35] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm,” *J. R. Stat. Soc. Ser. C (Applied Stat.*, vol. 28, no. 1, pp. 100–108, 1979, Accessed: Oct. 19, 2022. [Online]. Available: http://www.jstor.org/stable/2346830

[36] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” *5-TH BERKELEY Symp. Math. Stat. Probab.*, pp. 281--297, 1967, Accessed: May 23, 2022. [Online]. Available: https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Some-methods-for-classification-and-analysis-of-multivariate-observations/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsms

[37] Z. Huang, “Clustering large data sets with mixed numeric and categorical values,” *FIRST PACIFIC-ASIA Conf. Knowl. Discov. DATA Min.*, pp. 21--34, 1997, Accessed: May 23, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.9984

[38] Z. Huang, “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values,” *Data Min. Knowl. Discov. 1998 23*, vol. 2, no. 3, pp. 283–304, 1998, doi: 10.1023/A:1009769707641.

[39] A. Ahmad and S. S. Khan, “Survey of State-of-the-Art Mixed Data Clustering Algorithms,” *IEEE Access*, vol. 7, pp. 31883–31902, 2019, doi: 10.1109/ACCESS.2019.2903568.

[40] V. Melnykov and R. Maitra, “Finite mixture models and model-based clustering,” *Stat. Surv.*, vol. 4, pp. 80–116, 2010, doi: 10.1214/09-SS053.

[41] G. Preud’homme, K. Duarte, K. Dalleau, C. Lacomblez, E. Bresso, M. Smaïl-Tabbone, M. Couceiro, M. D. Devignes, M. Kobayashi, O. Huttin, J. P. Ferreira, F. Zannad, P. Rossignol, and N. Girerd, “Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark,” *Sci. Reports 2021 111*, vol. 11, no. 1, pp. 1–14, Feb. 2021, doi: 10.1038/s41598-021-83340-8.

[42] L. Hunt and M. Jorgensen, “Theory & Methods: Mixture model clustering using the MULTIMIX program,” *Aust. N. Z. J. Stat.*, vol. 41, no. 2, pp. 154–171, Jun. 1999, doi: 10.1111/1467-842X.00071.

[43] C. B. Storlie, S. M. Myers, S. K. Katusic, A. L. Weaver, R. G. Voigt, P. E. Croarkin, R. E. Stoeckel, and J. D. Port, “Clustering and variable selection in the presence of mixed variable types and missing data,” *Stat. Med.*, vol. 37, no. 19, pp. 2884–2899, Aug. 2018, doi: 10.1002/SIM.7697.

[44] C. Biernacki, G. Celeux, C. Biernacki, and G. Celeux, “MIXMOD : a software for model-based classification with continuous and categorical data To cite this version : MIXMOD :,” 2010, Accessed: May 23, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00469522

[45] R. Lebret, S. Iovleff, F. Langrognet, C. Biernacki, G. Celeux, and G. Govaert, “Rmixmod: The R Package of the Model-Based Unsupervised, Supervised, and Semi-Supervised Classification Mixmod Library,” *J. Stat. Softw.*, vol. 67, no. 6, pp. 1–29, Oct. 2015, doi: 10.18637/JSS.V067.I06.

[46] L. Hunt and M. Jorgensen, “Clustering mixed data,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 4, pp. 352–361, Jul. 2011, doi: 10.1002/WIDM.33.

[47] D. McParland and I. C. Gormley, “Model based clustering for mixed data: clustMD,” *Adv. Data Anal. Classif.*, vol. 10, no. 2, pp. 155–169, Jun. 2016, doi: 10.1007/S11634-016-0238-X/TABLES/3.

[48] D. P. Byar and S. B. Green, “The choice of treatment for cancer patients based on covariate information.,” *Bull. Cancer*, vol. 67, no. 4, pp. 477–490, 1980.

[49] D. F. Andrews and A. M. Herzberg, “Prognostic Variables for Survival in a Randomized Comparison of Treatments for Prostatic Cancer,” in *Data: A Collection of Problems from Many Fields for the Student and Research Worker*, New York, NY: Springer New York, 1985, pp. 261–274. doi: 10.1007/978-1-4612-5098-2_47.

[50] L. Hunt and M. Jorgensen, “Mixture model clustering for mixed data with missing information,” *Comput. Stat. Data Anal.*, vol. 41, no. 3–4, pp. 429–440, Jan. 2003, doi: 10.1016/S0167-9473(02)00190-1.

[51] M. Marbac, C. Biernacki, and V. Vandewalle, “Model-based clustering of Gaussian copulas for mixed data,” *Commun. Stat. - Theory Methods*, vol. 46, no. 23, pp. 11635–11656, 2017, doi: 10.1080/03610926.2016.1277753.

[52] I. Morlini, “A latent variables approach for clustering mixed binary and continuous variables within a Gaussian mixture model,” *Adv. Data Anal. Classif.*, vol. 6, no. 1, pp. 5–28, 2012, doi: 10.1007/s11634-011-0101-z.

[53] A. Foss, M. Markatou, B. Ray, and A. Heching, “A semiparametric method for clustering mixed data,” *Mach. Learn.*, vol. 105, no. 3, pp. 419–458, Dec. 2016, doi: 10.1007/S10994-016-5575-7/TABLES/16.

[54] A. H. Foss and M. Markatou, “kamila: Clustering Mixed-Type Data in R and Hadoop,” *J. Stat. Softw.*, vol. 83, pp. 1–44, Feb. 2018, doi: 10.18637/JSS.V083.I13.

[55] J. Jimeno, M. Roy, and C. Tortora, “Clustering Mixed-Type Data: A Benchmark Study on KAMILA and K-Prototypes,” in *Data Analysis and Rationality in a Complex World*, 2021, pp. 83–91.

[56] L. Alhusain and A. M. Hafez, “Cluster ensemble based on Random Forests for genetic data,” *BioData Min.*, vol. 10, no. 1, pp. 1–25, Dec. 2017, doi: 10.1186/S13040-017-0156-2/TABLES/4.

[57] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, “Ensemble clustering in medical diagnostics,” *Proc. IEEE Symp. Comput. Med. Syst.*, vol. 17, pp. 576–581, 2004, doi: 10.1109/CBMS.2004.1311777.

[58] A. Strehl, A. Strehl, J. Ghosh, and C. Cardie, “Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions,” *J. Mach. Learn. Res.*, vol. 3, pp. 583--617, 2002, Accessed: May 25, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.309

[59] J. J. Shen, P. H. Lee, J. J. A. Holden, and H. Shatkay, “Using Cluster Ensemble and Validation to Identify Subtypes of Pervasive Developmental Disorders,” *AMIA Annu. Symp. Proc.*, vol. 2007, p. 666, 2007, Accessed: May 25, 2022. [Online]. Available: /pmc/articles/PMC2655836/

[60] N. Iam-On, S. Garrett, C. Price, and T. Boongoen, “Link-based cluster ensembles for heterogeneous biological data analysis,” *Proc. - 2010 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2010*, pp. 573–578, 2010, doi: 10.1109/BIBM.2010.5706631.

[61] D. Dua and C. Graff, “UCI Machine Learning Repository.” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[62] J. Z. Huang, “A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining,” 1997.

[63] H. Luo, F. Kong, and Y. Li, “Combining Multiple Clusterings Via k-Modes Algorithm,” in *Advanced Data Mining and Applications*, 2006, pp. 308–315.

[64] H. G. Ayad and M. S. Kamel, “On voting-based consensus of cluster ensembles,” *Pattern Recognit.*, vol. 43, no. 5, pp. 1943–1953, May 2010, doi: 10.1016/J.PATCOG.2009.11.012.

[65] D. A. Linzer and J. B. Lewis, “poLCA: An R Package for Polytomous Variable Latent Class Analysis,” *J. Stat. Softw.*, vol. 42, no. 10, pp. 1–29, Jun. 2011, doi: 10.18637/JSS.V042.I10.

[66] J. P. Ferreira, K. Duarte, J. J. V. McMurray, B. Pitt, D. J. Van Veldhuisen, J. Vincent, T. Ahmad, J. Tromp, P. Rossignol, and F. Zannad, “Data-Driven Approach to Identify Subgroups of Heart Failure With Reduced Ejection Fraction Patients With Different Prognoses and Aldosterone Antagonist Response Patterns,” *Circ. Heart Fail.*, vol. 11, no. 7, Jul. 2018, doi: 10.1161/CIRCHEARTFAILURE.118.004926.

[67] J. K. Vermunt and J. Magidson, “Latent Class Cluster Analysis,” *Appl. Latent Cl. Anal.*, pp. 89–106, Dec. 2002, doi: 10.1017/CBO9780511499531.004.

[68] T. Kohonen, “The self-organizing map,” *Neurocomputing*, vol. 21, no. 1–3, pp. 1–6, Nov. 1998, doi: 10.1016/S0925-2312(98)00030-7.

[69] J. Vesanto and E. Alhoniemi, “Clustering of the self-organizing map,” *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 586–600, May 2000, doi: 10.1109/72.846731.

[70] J. Ong and S. Abidi, “Data Mining Using Self-Organizing Kohonen Maps: A Technique for Effective Data Clustering & Visualization.,” 1999, pp. 261–264.

[71] M. Y. Kiang, “Extending the Kohonen self-organizing map networks for clustering analysis,” *Comput. Stat. Data Anal.*, vol. 38, no. 2, pp. 161–180, Dec. 2001, doi: 10.1016/S0167-9473(01)00040-8.

[72] H. López García and I. Machón González, “Self-organizing map and clustering for wastewater treatment monitoring,” *Eng. Appl. Artif. Intell.*, vol. 17, no. 3, pp. 215–225, 2004, doi: https://doi.org/10.1016/j.engappai.2004.03.004.

[73] J. C. Gower, “A General Coefficient of Similarity and Some of Its Properties,” *Biometrics*, vol. 27, no. 4, p. 857, Dec. 1971, doi: 10.2307/2528823.

[74] M. Weatherall, J. Travers, P. M. Shirtcliffe, S. E. Marsh, M. V. Williams, M. R. Nowitz, S. Aldington, and R. Beasley, “Distinct clinical phenotypes of airways disease defined by cluster analysis,” *Eur. Respir. J.*, vol. 34, no. 4, pp. 812–818, Oct. 2009, doi: 10.1183/09031936.00174408.

[75] Ö. Akay and G. Yüksel, “Clustering the mixed panel dataset using Gower’s distance and k-prototypes algorithms,” *https://doi.org/10.1080/03610918.2017.1367806*, vol. 47, no. 10, pp. 3031–3041, Dec. 2017, doi: 10.1080/03610918.2017.1367806.

[76] D. Ebbert and S. Dutke, “Patterns in students’ usage of lecture recordings: A cluster analysis of self-report data,” *Res. Learn. Technol.*, vol. 28, 2020, doi: 10.25304/RLT.V28.2258.

[77] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, no. C, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.

[78] N. Bolshakova and F. Azuaje, “Cluster validation techniques for genome expression data,” *Signal Processing*, vol. 83, no. 4, pp. 825–833, Apr. 2003, doi: 10.1016/S0165-1684(02)00475-9.

[79] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” *Commun. Stat. Theory methods*, vol. 3, no. 1, pp. 1–27, 1974, doi: 10.1080/03610927408827101.

[80] G. W. Milligan and M. C. Cooper, “An examination of procedures for determining the number of clusters in a data set,” *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985, doi: 10.1007/BF02294245.

[81] C. Hennig and T. F. Liao, “How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification,” *J. R. Stat. Soc. Ser. C (Applied Stat.*, vol. 62, no. 3, pp. 309–369, 2013, doi: https://doi.org/10.1111/j.1467-9876.2012.01066.x.

[82] M. Hassani and T. Seidl, “Using internal evaluation measures to validate the quality of diverse stream clustering algorithms,” *Vietnam J. Comput. Sci.*, vol. 4, no. 3, pp. 171–183, 2017, doi: 10.1007/s40595-016-0086-9.

[83] L. der Maaten and G. Hinton, “Visualizing data using t-SNE.,” *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[84] W. McKinney, “{D}ata {S}tructures for {S}tatistical {C}omputing in {P}ython,” in *{P}roceedings of the 9th {P}ython in {S}cience {C}onference*, 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.

[85] J. Reback, W. McKinney, Jbrockmendel, J. Van den Bossche, T. Augspurger, P. Cloud, S. Hawkins, Gfyoung, Sinhrks, M. Roeschke, A. Klein, T. Petersen, J. Tratner, C. She, W. Ayd, S. Naveh, Patrick, M. Garcia, J. Schendel, A. Hayden, D. Saxton, V. Jancauskas, M. Gorelli, R. Shadrach, A. McMaster, P. Battiston, S. Seabold, K. Dong, Chris-b1, and H-vetinari, “pandas-dev/pandas: Pandas 1.2.4.” Zenodo, Apr. 12, 2021. doi: 10.5281/ZENODO.4681666.

[86] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nat. 2020 5857825*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[87] “datetime — Basic date and time types — Python 3.10.5 documentation.” https://docs.python.org/3/library/datetime.html#module-datetime (accessed Jun. 14, 2022).

[88] R. J. A. Little, “Missing-Data Adjustments in Large Surveys,” *J. Bus. Econ. Stat.*, vol. 6, no. 3, p. 287, Jul. 1988, doi: 10.2307/1391878.

[89] S. van Buuren, “Flexible Imputation of Missing Data, Second Edition,” *Flex. Imput. Missing Data, Second Ed.*, Jul. 2018, doi: 10.1201/9780429492259/FLEXIBLE-IMPUTATION-MISSING-DATA-STEF-VAN-BUUREN.

[90] L. Kaufman and P. J. Rousseeuw, “Partitioning Around Medoids (Program PAM),” in *Finding Groups in Data*, John Wiley & Sons, Ltd, 1990, pp. 68–125. doi: https://doi.org/10.1002/9780470316801.ch2.

[91] R Core Team, “R: A Language and Environment for Statistical Computing.” Vienna, Austria, 2021. [Online]. Available: https://www.r-project.org/

[92] G. Szepannek, “clustMixType: User-Friendly Clustering of Mixed-Type Data in R,” *R J.*, pp. 200–208, 2018, doi: 10.32614/RJ-2018-048.

[93] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, “cluster: Cluster Analysis Basics and Extensions.” 2022. [Online]. Available: https://cran.r-project.org/package=cluster

[94] D. Chiu and A. Talhouk, “diceR: Diverse Cluster Ensemble in R.” 2021. [Online]. Available: https://cran.r-project.org/package=diceR

[95] J. Boelaert, E. Ollion, and J. Sodoge, “aweSOM: Interactive Self-Organizing Maps.” 2021. [Online]. Available: https://cran.r-project.org/package=aweSOM

[96] G. Brock, V. Pihur, S. Datta, and S. Datta, “clValid: An R Package for Cluster Validation,” *J. Stat. Softw.*, vol. 25, no. 4, pp. 1–22, 2008, doi: 10.18637/jss.v025.i04.

[97] C. Hennig, “fpc: Flexible Procedures for Clustering.” 2020. [Online]. Available: https://cran.r-project.org/package=fpc

[98] J. H. Krijthe, "{Rtsne}: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation." 2015. [Online]. Available: https://github.com/jkrijthe/Rtsne

[99] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani, "Welcome to the {tidyverse}," *J. Open Source Softw.*, vol. 4, no. 43, p. 1686, 2019, doi: 10.21105/joss.01686.

[100] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. \& Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[101] M. L. Waskom, "seaborn: statistical data visualization," *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/joss.03021.

[102] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. development team, "Jupyter Notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016, pp. 87–90. [Online]. Available: https://eprints.soton.ac.uk/403913/

[103] RStudio Team, "RStudio: Integrated Development Environment for R." Boston, MA, 2020. [Online]. Available: http://www.rstudio.com/

[104] D. S. Chiu and A. Talhouk, "DiceR: An R package for class discovery using an ensemble driven approach," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–4, Jan. 2018, doi: 10.1186/S12859-017-1996-Y/FIGURES/2.

[105] H. Wickham, R. François, L. Henry, and K. Müller, "dplyr: A Grammar of Data Manipulation." 2022.

[106] M. Kuhn and H. Wickham, "Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles." 2020. [Online]. Available: https://www.tidymodels.org

[107] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. [Online]. Available: https://ggplot2.tidyverse.org

[108] D. Eddelbuettel and R. François, "{Rcpp}: Seamless {R} and {C++} Integration," *J. Stat. Softw.*, vol. 40, no. 8, pp. 1–18, 2011, doi: 10.18637/jss.v040.i08.

[109] D. Eddelbuettel, *Seamless {R} and {C++} Integration with {Rcpp}*. New York: Springer, 2013. doi: 10.1007/978-1-4614-6868-4.

[110] D. Eddelbuettel and J. J. Balamuta, "Extending extit{R} with extit{C++}: A Brief Introduction to extit{Rcpp}," *Am. Stat.*, vol. 72, no. 1, pp. 28–36, 2018, doi: 10.1080/00031305.2017.1375990.

[111] A. Ahmad and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data Knowl. Eng.*, vol. 63, no. 2, pp. 503–527, Nov. 2007, doi: 10.1016/J.DATAK.2007.03.016.

[112] H. A. Pathberiya, "DisimForMixed: Calculate Dissimilarity Matrix for Dataset with Mixed Attributes." 2016. [Online]. Available: https://cran.r-project.org/package=DisimForMixed