Qluster: An Easy-to-Implement Generic Workflow for Robust Clustering of Health Data

Cyril Esnault *, Melissa Rollot, Pauline Guilmin, Jean-Daniel Zucker

Posted Date: 6 January 2023

doi: 10.20944/preprints202205.0215.v3

Keywords: clustering; robustness; generic; workflow; algorithms

Article

Qluster: An Easy-to-Implement Generic Workflow for Robust Clustering of Health Data

Cyril Esnault [1,*], Melissa Rollot [1], Pauline Guilmin [1] and Jean-Daniel Zucker [2,3]

[1] Quinten France, 8 rue Vernier, 75017, Paris France
[2] Sorbonne University, IRD, UMMISCO, F-93143, Bondy, France
[3] Sorbonne University, INSERM, NUTRIOMICS, F-75013, Paris, France
* Correspondence: author: Cyril Esnault, cyrilesnault9@gmail.com

Abstract The exploration of heath data by clustering algorithms allows to better describe the populations of interest by seeking the sub-profiles that compose it. This therefore reinforces medical knowledge, whether it is about a disease or a targeted population in real life. Nevertheless, contrary to the so-called conventional biostatistical methods where numerous guidelines exist, the standardization of data science approaches in clinical research remains a little discussed subject. This results in a significant diversity in the execution of data science projects, whether in terms of algorithms used, reliability and credibility of the designed approach. Taking the path of parsimonious and judicious choice of both algorithms and implementations at each stage, this paper proposes Qluster, a practical workflow for performing clustering tasks. Indeed, this workflow makes a compromise between (1) genericity, as it is suitable regardless of the data volume (small/big) and regardless of the nature of the variables (continuous/qualitative/mixed), (2) ease of implementation, as it is based on few easy-to-use software packages, and (3) robustness, through the stability evaluation of the final clusters and through recognized algorithms and implementations. This workflow can be easily automated and/or routinely applied on a wide range of clustering projects. It can be useful both for data scientists with little experience in the field to make data clustering easier and more robust, and for more experienced data scientists who are looking for a straightforward and reliable solution to routinely perform preliminary data mining. A synthesis of the literature on data clustering as well as the scientific rationale supporting the proposed workflow is also provided. Finally, a detailed application of the workflow on a concrete use case is provided, along with a practical discussion for data scientists. An implementation on the Dataiku platform is available upon request to the authors.

Keywords: clustering; robustness; generic; workflow; algorithms

1. Introduction

Health data is of great importance to public health, research and medical development. It is any data related to the health conditions, outcomes and quality of life of an individual or population. Health data may be collected during the course of ongoing patient care (e.g., claims data, medical records, administrative data) or as part of a formal clinical trial program.

In health data analysis, clustering methods are a primary tool, by finding pockets of homogeneity within a heterogeneous population, to uncover different disease phenotype, stages of a disease, or variation in disease outcomes (*Franti¨* et al. 2022). A precise understanding of the clusters of patients suffering from a disease ultimately allows for the overall improvement of their care (*Windgassen* et al. 2018). In this respect, there is an extensive literature to discuss clustering tasks, be it for the choice of appropriate clustering methods (*Obembe* et al. 2019), for the clustering algorithms for large data (*Herawan* et al. 2014), for clustering methods for qualitative/mixed data (*Hennig* 2013), for methods to assess clustering quality, stability and number of clusters (*Lange* et al. 2004; *Nietto* et al. 2017), or for performing in-depth comparative statistical analysis of methods (*Jain* 2010; *Nagpal* et al. 2013).

The profusion of methods makes it difficult for most data scientists to choose and systematically apply a methodology that is complete, reasonably fast and satisfactory from a robustness point of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

view with respect to the clinical question they are trying to answer. The data scientist is indeed confronted with a very wide range of choices regarding both the algorithms and their implementations (including R, Python) in particular according to the nature of the data and their volume. Furthermore, on the contrary to the more "conventional" bio-statistical methods in clinical studies, the lack of clear guidelines on the data science approaches to be used leads to a greater subjectivity in the choice of approaches, and in particular, those of clustering in clinical data. This makes the statistical analysis plans for observational studies proposed by data scientists and the results obtained more variable.

The clustering process involves many decision steps, from the data preparation step to the evaluation of clusters' stability and clusters' description. To the authors' knowledge, there is not yet a single, simplified workflow in the literature that is easy to implement for both expert and non-expert health data scientists (with off-the-shelf tools in R or Python), well-supported by the literature, generic (e.g., regardless of the nature -continuous/binary/categorical/mixed- or volume of data -small/large), which facilitates its routine application. Most of the articles that come close to this goal focus on process automation (autoML, e.g., *Kamoshida* et al. 2020) or clustering methods comparison (*Wiwie* et al. 2015). This work (obviously) does not pretend to impose a single solution to a clustering problem as experience and literature have both shown that there is no single solution to a clustering task (see in particular *Kleinberg* (2002)). However, this work aims to give the data scientist guidance through an easy framework that can be used in routine practice in a wide variety of cases. This article is thus intended to:

i health data scientists, companies, or institutions that need a general workflow for routine -possibly automated - clustering projects on data of various types and volumes (e.g. for preliminary data mining), or

ii health data scientists with limited experience, who are looking for both an overview of the literature and an accessible and reusable workflow with concrete practical recommendations to quickly implement a complete unsupervised clustering approach adapted to various projects.

In this paper, we propose in section 2 a synthesis of the different methods related to unsupervised clustering in the literature and in relation to the implementations available in R or Python. In section 3, we propose *Qluster*, a generic workflow for clustering tabular data of any nature and size, while considering *(1)* the literature guidelines on how to perform robust clustering, and *(2)* the availability and ease of use of R or Python implementations for data scientist users. Then, in section 4, we detail this workflow through a step-by-step application on the open access Cardiovascular Disease[1]practice dataset to help data scientists to reapply this workflow on their own project with concrete recommendations. We then provide a practical discussion in section 5 and conclude in section 6. Data scientists will also find in both section 2 and section 4 all the necessary literature (rationale) to support the use of this workflow, which greatly facilitates the tedious work of writing the statistical analysis plan with innovative approaches (data science) in clinical research.

2. Statistical rationale and literature review on data clustering

This section discusses the state-of-the-art of clustering methods in the general clustering process (see in Figure 1 the steps 2 to 4. Step 2 will be discussed both in this section and in Section 4 through an illustrative example).

2.1. Overview of unsupervised clustering methods

As generally defined, clustering is "the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)" (*Altman* et al. 2017; "Cluster analysis" 2021). Such a task is used to group homogeneous subsets of observations to better understand their global heterogeneity. This is particularly true in clinical data analysis to describe disease heterogeneity, stratify patients and obtain

[1] https://www.kaggle.com/sulianova/cardiovascular-disease-dataset?select=cardio_train.csv

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

profiles of targeted populations. The result of a clustering task is in general an assignment of the input data into a fixed number of clusters. Two categories of clustering methods are usually distinguished according to the nature of such assignment: hard and soft clustering methods. Hard clustering provides a partition in which each object in the data set is assigned to one and only one cluster. Soft (or Fuzzy) clustering generates a fuzzy partition that provides a degree of membership of each object to a given cluster. This gives the possibility to express that objects belong to more than one cluster at the same time. It is of note that the definition of a cluster itself is not very precise which partly explains why there are so many clustering algorithms (*Estivill-Castro* 2002).

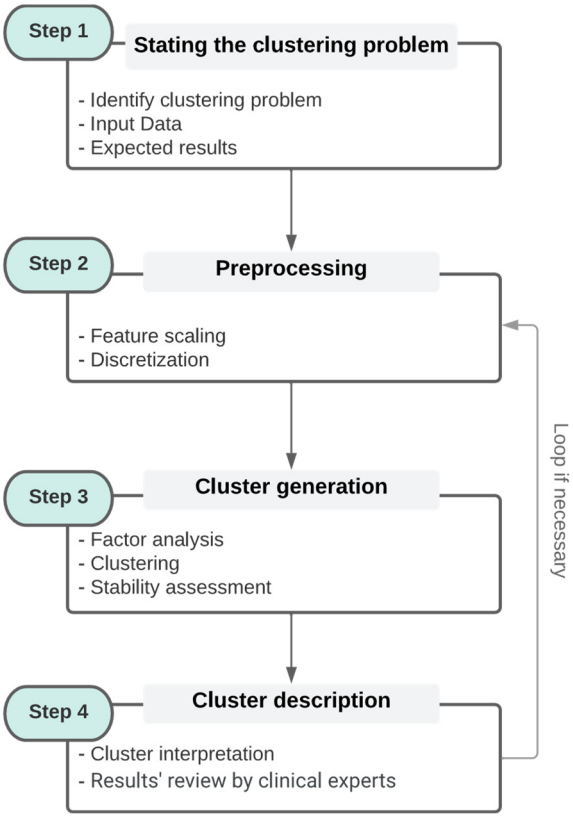


Figure 1. The general clustering process (It is broken down into four steps. Step 1 corresponds to the identification of the problem and the collection of data. Step 2 is a pre-processing step that includes the different transformations of the data. Step 3 is the clustering of the data itself while step 4 is the interpretation of the clusters with respect to the original data. The contents of each box (steps) are examples and not sub-steps that must be followed).

In the field of Machine Learning, clustering methods pertain to the so-called unsupervised learning methods. Clustering should not be confused with the field of Subgroup Discovery, which also aims at finding groups but in a supervised way, for example to identify prognostic factors of an outcome or predictive factors of the treatment effect on an outcome (*Esnault* et al. 2020; *Zhou* et al. 2019). The many clustering algorithms that exist in the literature (*Fahad* et al. 2014 ; *Ahmad* et al. 2019; *Xu* et al. 2010) can be classified according to the cluster models (centroid, connectivity, group, distribution, density, graph, etc.). Among the wide variety of methods, they are three main types, all producing a hard partition of the observations (Figure 2, which is adapted from Figure 1 in *Fahad* et al. (2014)):

- **Partitioning-based methods** (e.g. *K*-means (*MacQueen* 1967), *K*-medoid (*Jin* et al. 2010), PAM (*Ng* et al. 1994), *K*-modes (*Huang* 1997), *K*-prototype (*Huang* 1998), CLARA (*Kaufman* et al. 2009), FCM (*Bezdek, Ehrlich,* et al. 1984), etc[2].),

[2] For further details, see *Celebi* (2014).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

- **Hierarchical-based methods** (e.g. BIRCH (*Zhang* et al. 1996), CURE (*Guha* et al. 1998), ROCK (*Guha* et al. 2000), etc.),
- **Density-based methods** (e.g. DBSCAN (*Ester* et al. 1996), DENCLUDE (*Keim* et al. 1998), etc.).

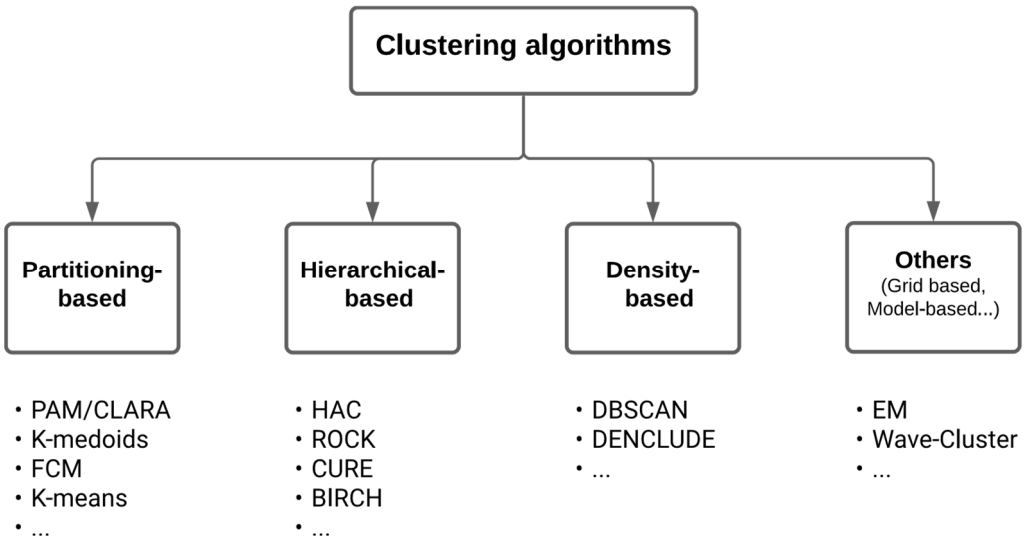


Figure 2. A taxonomy of clustering algorithms. Below the boxes are listed well-known algorithms of the corresponding types.

The **first** type of method is considered to be the most popular class of clustering algorithms for its ease-of-implementation, simplicity, efficiency, and empirical success (*Jain* 2010). It aims at directly obtaining a single data partition into *K* clusters. Partitioning-based methods require setting the number *K* of clusters which is rarely known a priori but can be estimated from the data using several known methods (*Calinski´* et al. 1974; *Gordon* 1999; *Halkidi* et al. 2001; *Hennig* 2013; *Hennig* 2014; *Meila˘* 2007; *Milligan* et al. 1985). These include the optimization of internal validity metrics that reflect the compactness and separation of the clusters (e.g. average Silhouette Width, Davies-Bouldin index, Calinski-Harabasz index, Dunn index, etc.). Equally, some of the partitioning-based methods rely on a random initialization of different *K*-centroids which can lead to different outputs (local optimum), non-reproducible clusters, or wrong or empty clustering. Some solutions exist, such as the *K*-means++ algorithm, which includes a smart centroid initialization method for the *K*-means algorithm (*Arthur* et al. 2007). The goal is to spread out the initial centroid by assigning the first centroid randomly then selecting the rest of the centroids based on the maximum squared distance. The idea is therefore to push the centroids as far as possible from one another. Similarly, the PAM algorithm is a deterministic *K*-medoid algorithm that directly integrates an initialization procedure called BUILD. During the BUILD phase the first medoid is selected to be the one that has the minimum cost, with cost being the sum over all distances to all other points.

The **second** type produces a hierarchy of clusters, called dendrogram, especially useful when one needs several hard partitions at different hierarchical levels (i.e., from a macro vision with a few groups, to a micro vision with many groups). These hierarchical methods have a major drawback though, that once a step (merge or split) is performed, it is not undone, potentially making erroneous decisions impossible to correct: they are often greedy algorithms that optimize a local criteria without backtracking while the clustering problem is by definition a global optimization problem. Moreover, hierarchical-based methods generally have higher time and space complexities than partitioning-based ones, and rely on more input parameters which leaves more room for subjectivity regarding the choice of settings, with a direct impact on the generated clusters (*Fahad* et al. 2014). Some studies have also shown that hierarchical-based algorithms lead to worse clustering results than partitioning algorithms, suggesting that the latter are well-suited for clustering large datasets due to not only their relatively low computational requirements, but also comparable or even better clustering performance (*Kaushik* et al. 2014; *Zhao* et al. 2002).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

The **third** type of method does not either explicitly require a number of clusters nor does it mainly rely on a distance threshold from a "center" (like partitioning-based methods do). On the contrary, density-based methods rely on the estimated density of observations to perform the partitioning. Such a method is in this sense more local and allows to represent clusters whose topology are less induced by the sole distance used (like hyper-spheres when using the Euclidean distance in partitioning-based methods). This strategy may however be associated with a greater propensity to overfit data and greater difficulties to set up the hyper-parameters.

Beyond these three main types of methods there are many other alternative clustering approaches including grid-based methods (e.g. Wave-Cluster (*Sheikholeslami* et al. 1998), STING (*Wang* et al. 1997), ...). They perform the clustering on grids rather than on the whole dataset. There are also model-based methods that optimize the fit between the data and predefined models, assuming that the data is generated by a mixture of underlying probability distribution (e.g. mixture density model (EM (*Do* et al. 2008)), conceptual clustering (COBWEB (*Fisher* 1987)), neural networks model (SOMs (*Ciampi* et al. 2000))). More recently, a wide range of new approaches to clustering based on deep learning have emerged (*Aljalbout* et al. 2018); they are mainly used to cluster unstructured data. Deep neural networks (DNN) can be effective ways to map a high-dimensional data space to a lower dimensional feature space, and thus improving clustering results. Nevertheless, DNN often requires large datasets and a procedure for post-hoc interpretability of the clusters (the representation learned by DNN architectures is not easily understandable). Finally, recent methods for graph clustering focus more specifically on finding sets of nodes in networks or graphs that have more connections within the set than outside the set (*Sieranoja* et al. 2022). For more on the types of clustering algorithms and their suitability to the data types (categorical, text, multimedia, stream, time series, etc.), see *Oyelade* et al. (2019).

2.2. Choosing an appropriate clustering approach

The choice of the appropriate approach to be used rely on many aspects (*Fahad* et al. 2014; *Ahmad* et al. 2019) such as:

i its ability to handle the desired type of data (binary / nominal / ordinal / numerical),
ii the dimensionality of the data (see e.g. *Mittal* et al. (2019)),
iii the size of the data (small to large data),
iv the availability of reliable implementations in software (e.g. R and Python - the 2 most used statistical software by data scientists).

Partitioning-based methods are known to be composed of many variants to directly handle continuous (*K*-means, PAM, CLARA, FCM, etc.), categorical (*K*-modes, *K*-medoid, etc.) and mixed variables (*K*-prototype, KAMILA (*Foss* et al. 2018), etc.). In addition, the ability of some of these algorithms to directly handle input dissimilarity matrices facilitates the pre-transformation of the original data into data of the desired type prior to clustering, using suited distance measure (e.g. *McCane* et al. 2008). This method is particularly used to convert categorical or mixed data into numerical data, as there is more literature and algorithms implemented in software for continuous data. (e.g. scikit-learn[3] in Python, or *cluster*[4] and *fpc*[5] R packages. The two latter packages both provide a large number of clustering and clusters stability assessments methods, as well as functions to compute dissimilarity matrices and describe the results). Another known alternative consists of one-hot-encoding categorical data into binary variables and treating the latter as continuous (e.g. in *Li* et al. 2017). It is however necessary to downweigh the obtained variables so that no more weight is given to the original variables with more modalities. Finally, dimensionality reduction methods, such as factor analysis (Principal Component Analysis (PCA) for continuous data, Multiple Correspondence Analysis (MCA) for qualitative data, Factor Analysis of Mixed Data (FAMD) for

[3] https://scikit-learn.org/stable/

[4] https://cran.r-project.org/web/packages/cluster/index.html

[5] https://cran.r-project.org/web/packages/fpc/index.html

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

mixed data (*Pages`* and *Husson* 2017)), can be used before the clustering as a first step to transform the data into numerical components (i.e., the coordinates of the observations on each dimension).

Factor analysis has many other advantages for clustering tasks, such as reducing the dimensionality (making easier the clustering task), reducing noises (by removing the last components that only bear random noise, leading to a more robust unsupervised learning) and dealing with variables that bear similar information and/or are highly correlated (*Pages`* and *Husson* 2017). In the case of qualitative data, a convenient practice for accommodating cluster-level observation heterogeneity in MCA is to adopt a two-step sequential, tandem approach (*Arabie* et al. 1994): in the first step a low-dimensional representation of the categorical variables is obtained via MCA; in the second step some variety of cluster analysis is used to identify a set of relatively homogeneous observations groups on the basis of the low-dimensional data. In addition to the ease in which the two-step sequential approach can be implemented, there can be substantive reasons for adopting this approach as well (*Green* et al. 1995). Alternative methods consist of using simultaneously both MCA and a clustering approach in a single framework, so that the low-dimensional data can be chosen to facilitate the identification of clusters (*Bock* 1987; *DeSarbo* et al. 1991; *De Soete* et al. 1994). However, these methods lack implementations (both in R and Python), hindering their use within a clustering workflow. Finally, the selection of the number of components to be kept is then the key step. This can be based on several methods including permutation tests (*Hwang* and *Takane* 2010), cross-validation-based methods (*Bro* et al. 2008, *Josse, Chavent,* et al. 2012), or methods based on the amount of information carried by each of the dimensions generated by the analysis, either compared to an average value equivalent to the Kaiser's rule in PCA (*Lorenzo-Seva* 2011) and/or represented by a scree plot (*Clausen* 1998; *Drennan* 2010). The latter method has been found to perform fairly well and is the most widely used to select the optimal number of dimensions (*Zwick* et al. 1986; *Bandalos* et al. 2009). It consists of looking at the bend in the falling curve (so called "elbow") indicating an optimal dimensionality (if there is no obvious elbow, one can choose the number of components just before a flat appears). It has been adapted from PCA (*Cattell* 1966) and used in the context of correspondence analysis (*Costa* et al. 2013). All factor analysis methods can notably be found in R in the well-known *FactoMineR*[6] package, and in Python in the *prince*[7] GitHub library (although issues are still open for the latter). Methods for estimating the number of dimensions to be kept can be found in R in many packages such as the *FactoMineR* and *missMDA*[8]packages, using cross-validation methods, or in the *factoextra*[9] package (e.g. scree plot). In Python and to the best of authors knowledge, one would need to code these methods to apply them as no specific functions were found.

Whether a factor analysis is performed as a first step or not, the need to choose a distance measure is critical, as some of them are only appropriate according to the type of data, or are preferred in some cases. Indeed, continuous data require appropriate distance measure to obtain the dissimilarity matrix (e.g. Euclidean and Manhattan distances), while categorical data are4 widely handled with simple matching methods (e.g. Hamming distance for symmetric measures, which is equivalent to Manhattan distance on binary variables, and Jaccard distance for asymmetric measures to favor positive co-occurrences over the negative ones). Methods to handle mixed data can consist of combining above-mentioned methods, such as the Gower distance (i.e. simple matching methods and Manhattan distance).

Finally, some aspects need careful attention when dealing with large data. This includes the size of the dataset, as the candidate algorithm must handle either or both high dimensionality and a massive number of observations (including outliers/noisy data), which makes difficult, and sometimes impossible, dissimilarity matrices to be computed. Equally, fast running time is essential with large data as the clustering needs to be performed several times, notably to assess cluster

[6] https://cran.r-project.org/web/packages/FactoMineR/index.html

[7] https://github.com/MaxHalford/prince

[8] https://cran.r-project.org/web/packages/missMDA/index.html

[9] https://cran.r-project.org/web/packages/factoextra/index.html

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

stability and optimize the clustering hyperparameters (e.g. in the use case in section 4, clustering was replicated 550 times). Few strategies exist to deal with massive data, such as relying on algorithms of lower complexity (e.g. *K*-modes and FCM that are O(n), (*Fahad* et al. 2014)). The latter however is quickly limited as computing time increases linearly with the size of data. Alternative methods consist of working on approximations (*Sieranoja* et al. 2019) or subsets of the whole dataset to cluster smaller datasets before generalizing them (e.g. Mini Batch *K*-means (*Sculley* 2010), CLARA (*Kaufman* et al. 2009), CLARANS (*Ng* et al. 2002)).

The CLARA algorithm is an extension to *K*-medoids methods (e.g. PAM), that is known to be more robust than *K*-means-based algorithms as they minimize a sum of dissimilarities instead of a sum of squared Euclidean distances (*Jin* et al. 2010). CLARA allows to deal with data containing a large number of observations (more than several thousand) using a sampling approach, in order to reduce computing time and RAM storage problems. Instead of finding medoids for the entire dataset, CLARA considers a small sample of the data and applies the PAM algorithm to generate an optimal set of medoids. CLARA repeats the sampling and clustering processes several times in order to minimize the sampling bias. In practice, its strength lies in the possibility to adjust the number of samples and the sample sizes, in order to both make calculation time acceptable and storage in RAM possible. This is indeed essential to enable the stability of clustering to be assessed and the best partitioning to be found by repeating the clustering process many times. Compared to CLARA, CLARANS presents a trade-off between the cost and the effectiveness of using samples to obtain clustering. Mini-batch *K*-means, CLARA and CLARANS can all be found both in R (e.g. respectively *cluster*, *FPC*, and *QTCAT*[10] packages) and in Python (e.g. respectively scikitlearn, *pycluster*[11] and *pyclustering*[12]). Please note though that both the quality and the maintenance of libraries on GitHub (*QTCAT*, *pycluster* and *pyclustering*) cannot be guaranteed by the present authors.

2.3. Methods for clusters description

The interpretability of clusters generated by clustering algorithms remains one of the most important challenges in clinical data analysis, as it is often the case with machine learning algorithms (*Vellido* 2020). Indeed, the best results will only make sense if they are interpretable by end users. Conventional methods do not provide consensus on how to characterize clusters and this is even more valid in the health sector, where the interpretation of clusters is a matter of medical knowledge of the data itself (*Kiselev* et al. 2019).

The simplest yet most efficient method remains to compute relevant intra- and inter-clusters descriptive statistics using the initial variables to identify a mapping of the generated clusters based on means or median values (resp. proportions) for continuous (resp. categorical) variables. This can be completed by performing clusterwise distributions comparison with overall distributions, as well as using hypothesis testing to identify input variables whose differences are statistically significant between clusters (*Bousquet* et al. 2015). Such implementations dedicated for clusters' description may be found in R (e.g. the *cluster.varstats()* function in the *FPC* package that also provides tables and plots) and to the best of our knowledge no such function was found in Python.

Alternative for making the clusters' description step easier may consist of learning an interpretable multiclass supervised classifier (e.g. decision tree) on clusters labels (outcome) to highlight characteristics and specificities associated with each group. Other methods propose to include the interpretability of clusters directly within the clustering algorithm, and not as a step done afterwards (*Bertsimas* et al. 2021), notably by adding tunable parameters related to interpretability (e.g. see *Saisubramanian* et al. (2020), with a Python implementation found on GitHub[13]).

---

[10] https://rdrr.io/github/QTCAT/qtcat/man/clarans.html

[11] https://github.com/daveti/pycluster

[12] https://pyclustering.github.io/

[13] https://github.com/sandysa/Interpretable Clustering

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

Finally, methods dedicated to the visualization of clusters make it easier their interpretation, such as PCA, multidimensional scaling (*Torgerson,* 1952), t-SNE (*Maaten and Hinton,* 2008) and uniform manifold approximation and projection (UMAP, *McInnes et al.,* 2018).

2.4. Methods for clustering validity and stability assessment

Clustering assessment step is an important phase to increase confidence in results and consists of evaluating both the clustering validity and stability.

Regarding the validity of clustering, one can first distinguish the external validity metrics (*Rezaei* et al. 2016) that can be used to compare the clusters obtained with the ground truth, which is rarely known. Then, the internal validity metrics that assess the goodness of a data partition using quantities inherited from the data, such as compactness (e.g. the maximum pairwise intra-cluster distances), connectedness (e.g. Connectivity metric) or separation (*Handl* et al. 2005; *Bezdek* and *Pal* 1998). The Dunn Validity Index and the Silhouette coefficient are both commonly used metrics, notably to define the optimal number of clusters, as they both assess the separation (i.e. the inter-cluster distances) over the compactness (intra-cluster distances). Although previous works have shown that there is no single internal cluster validation index that outperforms the other indices, *Arbelaitz* et al. (2013) compare a set of internal cluster validation indices in many distinct scenarios, indicating that the Silhouette coefficient yielded the best results in most cases. Alternatives exist for estimating the number of clusters in a dataset regardless of the clustering methods, such as the "gap statistic" that compares the change in within-cluster dispersion with that expected under an appropriate reference null distribution (R. *Tibshirani* et al. 2001).

Regarding the stability of clustering, several methods are proposed in the literature, by repeating the clustering process several times under conditions that are different from those of origin. These include procedures used in bioinformatics that remove one column at a time (*Datta* et al. 2006; *Handl* et al. 2005). Several metrics can then be computed between the set of clusters (*Brock* et al. 2008), such as the average proportion of non-overlap (APN), the average distance (AD) and the figure of merit (FOM). These methods are notably proposed in the *clValid*[14] R package and its main function *clValid()*. The latter includes many clustering algorithms (hierarchical, *K*-means, diana, fanny, som, model, sota, pam, clara, and agnes) and allows for a direct assessment of clusters' stability through the "validation" argument.

Other approaches consist in perturbing the original data, either using bootstrapping (*Efron* 1979; *Efron* and R. J. *Tibshirani* 1994), noising and/or sampling methods (*Hennig* 2008). The Jaccard similarity statistic is then often used as a metric for assessing stability, by computing the similarities of the original clusters to the most similar clusters in the resampled data. Such methods are implemented in the *FPC* R package, notably in the *clusterboot()* function. The latter is an all-inclusive package that also allows for clustering using a wide variety of algorithms. (e.g. *K*-means, hierarchical clustering, normal mixture models, PAM, CLARA, DBSCAN, spectral clustering, etc.), making it easy for a data scientist to generate, compare and assess stability of clusters.

As for Python, while there are packages to evaluate the internal validity of clusters (Silhouette coefficient, Rand Index, Calinski-Harabasz Index (*Calinski´* et al. 1974), see in particular the *sklearn.cluster* library), no Python library was found to evaluate the stability of generated clusters. This reinforces the fact that Python does not cover as easily as R the whole clustering process, as there is no Python package that includes all the steps of interest (clusters generation, internal validity evaluation, clustering optimization, clusters stability evaluation and clusters description), as in the *FPC* R package.

Finally, methods for testing cluster stability on a hold-out dataset are barely mentioned in the literature. This would consist of pre-allocating the observations in the test set into the clusters obtained from the learning set, and by clustering the test dataset to check for good allocation. Though no implementation has been found either in R or Python. We can see a simplified application of clustering of 2 independent datasets in *Saint Pierre* et al. (2020).

[14] https://cran.r-project.org/web/packages/clValid/index.html

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

3. The Qluster workflow

3.1. Research objective

Many statisticians/data scientists are confronted with the great number of algorithms and implementations for data clustering. This can make it difficult to manage clustering studies, and is likely to generate analytical strategies that are insufficiently rigorous, not consensual or not adapted to the problem. This is particularly the case for any statistician/data scientist in contract research organizations that provide support to healthcare industries, who has the responsibility to conduct clustering analyses but is still little experienced in using them. Our goal is to propose a practical workflow for data scientists because of its **genericity of application** (e.g. usable on small or big data, on continuous, categorical or mixed variables, on database of high-dimensionality or not, with multicollinearity or not, etc.) while preserving the **simplicity of implementation** and **use** (need for few packages, algorithms, parameters, …) and the **robustness and reliability** of the methodology (e.g. evaluation of the stability of clusters, use of proven algorithms and robust packages, etc.). The objective of this workflow is therefore not to be the solution to all situations, but to propose a simple and robust basis that is as generic as possible. In a way, a choice that aims to be "globally optimal" for practice, but not optimal in every case. This generic workflow can be useful both for data scientists with little experience in the field to make data clustering easier and more robust, and for more experienced data scientists who are looking for a straightforward and reliable solution to routinely perform preliminary data mining.

3.2. Method

The criteria[15] defining the properties of the desired workflow are the following:

- Criteria for achieving **genericity**: applicability to small and big data, applicability to continuous or categorical or mixt data, and management of high dimensionality.
- Criteria for achieving **ease of implementation and use**: number of packages used, of algorithms used, of parameters to tune, use of "all-inclusive" packages covering at best the general clustering process
- Criteria for achieving **robustness and reliability**: management of noise data, of multicollinearity, methods considered for clusters' stability assessment, reliability of packages used (e.g. hosting site, renown, …), reliability of algorithms used (e.g. renown, literature, …).

Facing the great diversity of packages[16] and algorithms, and considering our goal of preserving the simplicity of implementation and use of the desired generic workflow, we focused on handy[17] packages to cover main algorithms and steps in the general clustering process (see Figure 1). The latter can include functions e.g. for clustering, clusters optimization, clusters evaluation, clusters stability evaluation and clusters description (clustering algorithms suites). For Python, we considered the module *sklearn.cluster* from the scikit-learn library. For R, the following packages were selected: *FPC*, *cluster*, *clue* and *ClValid*. All these implementations, functions and algorithms that compose them are considered robust and therefore **meet part of the criteria** for achieving robustness and reliability.

3.3. Preliminary work

When relevant, we matched the selected implementations to the defined criteria (see Table 1 for R packages and skikit-learn library, and Appendix A for clustering algorithms that compose them).

---

[15] A discussion on other possible criteria and ways to integrate them into the workflow is presented in section 5.1 (e.g. management of missing data and outliers).

[16] e.g. CRAN Task View for cluster analysis: https://cran.r-project.org/web/views/Cluster.html

[17] e.g. of handy R packages: https://towardsdatascience.com/a-comprehensive-list-of-handy-r-packages-e85dad294b3d

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

Table 1 shows that neither the *cluster* R package nor the *sklearn-cluster* module in Python allow to evaluate the stability of clusters. As indicated in section 2, one should code this step oneself in python, or link (if possible) with other packages in R. Of the selected R packages, *FPC* was the most downloaded in 2021, and provides the most internal assessment metrics. *Clue* and *FPC* evaluate clusters stability by bootstrapping but only *FPC* include others methods such as noising, the complementarity of the two methods being recommended by *Hennig* (2008). *ClValid* on the other hand proposes simpler methods, mainly used in genomics, for evaluating the stability of clusters by removing one by one the variables.

Table 1. Description of selected software implementations (Number of downloads in 2021 is based on the cran_downloads() function in the cranlogs R library. NA stands for Not Applicable).

Libraries	Language	*Ease of implementation and use*				*Robustness*	Number of downloads in 2021
		Data processing	data clustering	Internal validation metrics	Clusters description	Cluster stability assessment	
FPC	R	Yes	Yes	Silhouette width, Calinski-Harabasz index, Hubert's gamma coefficient, Dunn index, Tibshirani and Walther's prediction strength, etc.	Yes	Bootstrap, noise, resampling, etc.	985853
cluster	R	Yes	Yes	Silhouette width, Gap statistic, etc.	Yes	/	891577
clue	R	No	Yes	Variance accounted for (VEF), Deviance accounted for (DEF), ...	Yes	Bootstrap	467260
ClValid	R	No	Yes	Connectivity, Silhouette width, Dunn Index, etc.	Yes	Removing each column, one at a time	96676
sklearn.cluster	Python	Yes	Yes	Rand index, Normalized Mutual Information (NMI), Adjusted Mutual Information (AMI), Silhouette width, Calinski-Harabasz Index, Davies-Bouldin Index, etc.	Yes	/	NA

The table in Appendix A was created based on Table 1 in *Fahad* et al. (2014), which we adapted for our purpose. Overall, Appendix A shows that none of the algorithms included in the selected packages satisfies all the properties sought in terms of genericity, simplicity of use and implementation, and robustness. For example, CLARA and Mini-batch *K*-means both allow very good handling of large data, are adapted to some extent to high dimensionality, and rely on few parameters to be optimized. However, they only apply to continuous data, and are not particularly suitable for noisy data. Also and unlike CLARA, Mini-batch *K*-means is only included in the scikit-learn module on Python.

This first synthesis work highlights the challenge to overcome.

3.4. Qluster

Based on the literature review (section 2) and the preliminary work (section 3.3), we propose the *Qluster* workflow (see Figure 3), a set of methods that together represent a good balance for data scientists to make clustering on health data in a practical, efficient, robust and simple way. It covers the cluster generation step (step 3) through: 1- factor analysis, 2- data clustering and 3- stability evaluation. The output of the factor analysis (PCA, MCA, or FAMD) is the matrix of the coordinates of the individuals on the factorial dimensions, a table of continuous variables, allowing then the clustering by a PAM algorithm. For an in-depth discussion regarding the *Qluster* workflow, see section 5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

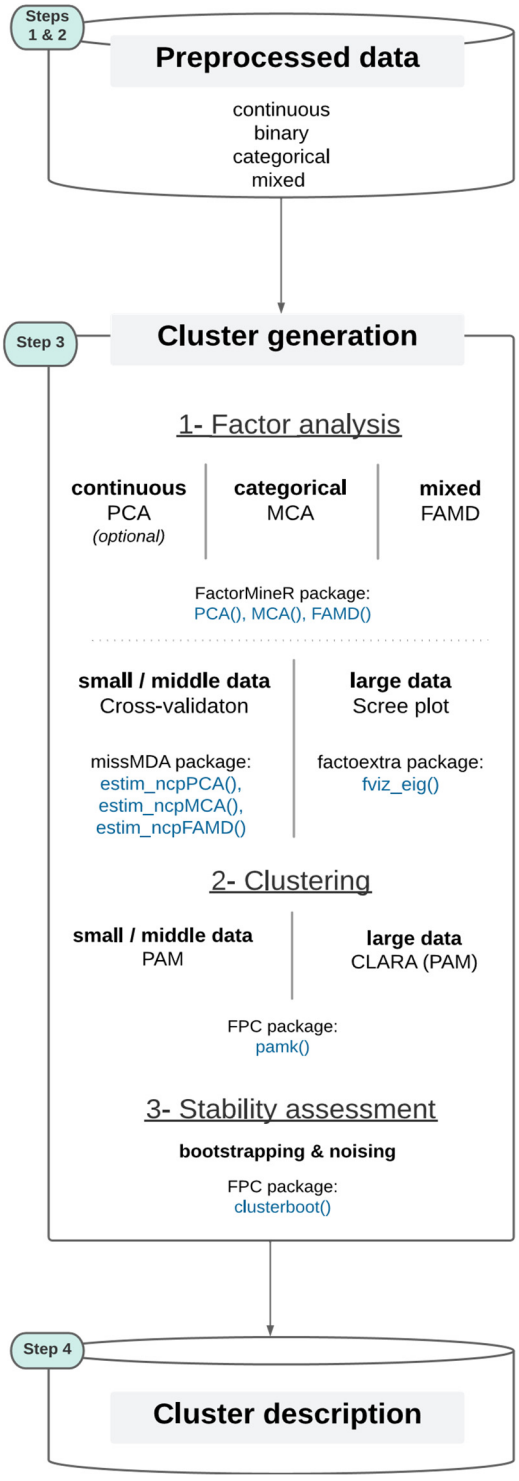


Figure 3. The Qluster workflow (The colored step pads correspond to those detailed in Figure 1.).

To summarize, *Qluster* tries to generalize clustering tasks through a **generic** framework that is:

- **Adapted to variables of any nature**, be it categorical only, continuous only, or a mix of both. This is made possible by transforming all data in the continuous setting (that is both more mature in the literature and simpler to process) using methods for factor analysis (MCA for categorical data only or FAMD for mixed data (*Pages`* 2004)). As mentioned in section 2.2, the latter also allows for dealing with **collinearity**, **high-dimensionality** and **noise**. It also makes the work for a clustering algorithm easier, as there are both fewer variables to deal with and greater clarity in information to cluster (factor analysis methods are themselves meant to uncover profiles into components of richer information).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

- **Adapted to datasets of any volume**, be it small or large data. Indeed, the same partitioning-based algorithm is used (PAM), either applied entirely on a dataset of reasonable size, or on samples of a large[18] dataset (CLARA algorithm), using the same *pamk()* function from the *FPC* R package. In addition to this **practical** aspect, PAM was chosen over the (widely used) *K*-means algorithm based on its ability to be deterministic and to deal with Manhattan distance which is **less sensitive to outliers** than the Euclidean distance (*Jin* et al. 2010). Moreover, PAM is known for its **simplicity of use** (fewer parameters e.g. than with DBSCAN or BIRCH (*Fahad* et al. 2014)) and is also implemented in an easy-to-use all-inclusive package (not the case for e.g. CLARANS, KAMILA, Mini batch *K*-means, DENCLUE and STING that are suited for large datasets but where assessing clusters stability would require extensive code development by data scientists). More details on the choice of the clustering algorithm can be found in section 5.

In addition, the *Qluster* workflow relies solely on four state-of-the-art R packages, allowing data scientists to quickly manage data of different nature and volume and perform robust clustering:

- Both the tasks of clustering and clusters stability assessment are handled using the *FPC* R package (functions *pamk()* and *clusterboot()* respectively). R has been chosen over Python because the former offers all the clustering methods desired, and no package including all the steps of interest to clustering was found for the latter (one would have to code some steps by him/herself, see section 2 for more details). The *clusterboot()* function offers many ways to assess clusters' stability, but one selects the two followings for routine practice and for their complementarity as mentioned in *Hennig* (2008): bootstrapping and noising.
- The factor analysis part is handled using the *FactoMineR* R package (function *PCA()*, *MCA()*, *FAMD()*, for continuous, categorical, and mixed data respectively - the latter function generalizing the others). This step is optional in the case where only continuous variables are in input[19]. To select the optimal number of components to keep, one recommends for small data to use the deterministic cross-validation technique implemented in the *missMDA* package (function *estim ncpPCA()*, *estim ncpMCA()*, *estim ncpFAMD()* (*Josse, Chavent,* et al. 2012)). As this method requires high computing time, the standard "elbow" method in a scree plot is recommended for large data, using the *factoextra* R package (function *fviz eig()*).

Finally, the *Qluster* workflow is operationalizable and implementable from end to end (see in Appendix in section B a picture of implementation in the Dataiku[20] platform. Available upon request: contact@quinten-france.com).

This generic workflow, usable in most situations, can be described through the following pseudocode (Algorithm 1):

[18] The notion of large data, as well as how to fix the hyperparameters *samples* and *sampsize* in CLARA algorithm, may vary according to the computing capabilities of the user's system. One recommends users to pre-tests different scenarios to adapt these thresholds to their own settings. For guidance, this workflow applied on the case study in section 4 (34,134 observations and 9 variables) took 5 hour and 30 minutes with 8 CPUs and 10 GB RAM

[19] It is worth noting that standardization of continuous data is recommended before using PCA, to not give excessive importance to the variables with the largest variances.

[20] https://www.dataiku.com/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

Algorithm 1: The *Qluster* pseudo-code

input : X: The input data
$packages$: FactoMineR, factoextra, FPC, missMDA
output : Q: A clustering of X and associated measures

if X *is continuous only* **then**
  F = *PCA(X)*, with F a FactoMineR object of class PCA
else if X *is categorical only* **then**
  F = *MCA(X)*, with F a FactoMineR object of class MCA
else
  `// mixed continuous and categorical`
  F = *FAMD(X)*, with F a FactoMineR object of class FAMD
end
Define from F the matrix M of coordinates of individuals on each dimension
if X *is "large"* **then**
  Apply *fviz_eig()* on F to select C_{opt}, a sufficient number of components
  Define the M_{opt} matrix as M restricted to C_{opt} components
  P = *pamk*(M_{opt}), with *usepam* = FALSE (CLARA), *criterion* = "asw", *scaling* = FALSE, and setting at convenience *samples*, *sampsize*, and *krange*
  Let K the optimal number of clusters in P
else
  `// X is not "large"`
  if X *is continuous only* **then**
    F_{ncp} = *estim_ncpPCA(X)*, with large [ncp.min, ncp.max] range
  else if X *is categorical only* **then**
    F_{ncp} = *estim_ncpMCA(X)*, with large [ncp.min, ncp.max] range
  else
    `// mixed continuous and categorical`
    F_{ncp} = *estim_ncpFAMD(X)*, with large [ncp.min, ncp.max] range
  end
  Retrieve the optimal number of dimensions C_{opt} from F_{ncp}
  Define the M_{opt} matrix as M restricted to C_{opt} components
  P = *pamk*(M_{opt}), with *usepam* = TRUE (PAM), *criterion* = "asw", *scaling* = FALSE, and setting at convenience *krange*
  Let K the optimal number of clusters in P
end
S = *clusterboot*(M_{opt}), with *bootmethod* = c("boot", "noise"), *krange* = K, *clustermethod* = "pamkCBI" and the same parameters as in the previous step. Loop on the *noise_level* parameter to test different noise levels
Return all useful results in Q

4. Detailed workflow methodology

4.1. The Cardiovascular Disease dataset and the objective

The Cardiovascular Disease[21] dataset includes 70,000 patients with or without a cardiovascular disease, and 12 variables (5 of them are continuous).

The following raw variables was used (raw variables name are in italic):

1. Age (days, converted into years) - *age*
2. Height (cm) - *height*
3. Weight (kg) - *weight*
4. Gender (M/F) - *gender*
5. Systolic blood pressure (SBP) (mmHg) - *ap hi*
6. Diastolic blood pressure (DBP) (mmHg) - *ap lo*
7. Cholesterol (categories 1: normal, 2: above normal, 3: well above normal) - *cholesterol*
8. Glucose (categories 1: normal, 2: above normal, 3: well above normal) - *gluc*
9. Smoking (Y/N) - *smoke*
10. Alcohol intake (Y/N) - *alco*
11. Physical activity (Y/N) - *active*
12. Presence or absence of cardiovascular disease (Y/N) - *cardio*

The objective of this section is to present in detail the application of the *Qluster* workflow proposed in section 3 on the following use case: to characterize the phenotypes of patients with a

[21] On Kaggle: https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

cardiovascular disease (subset of patients with *cardio* = Y). This represents 34,979 patients (about 50% of the whole population).

4.2. Step-by-step application of the Qluster workflow

The following section details the application of the *Qluster* workflow to the cardiovascular dataset to help scientists use it for their own project. Additional elements to the ones presented in section 2 supporting the present methodology are also provided when relevant. We first present the preprocessing of the dataset, in which notably the few continuous variables are converted into qualitative data, before applying a MCA, that is a data-reduction technique for exploring the associations among multiple categorical variables (*Greenacre* and *Blasius* 2006; *Greenacre* 1984; *Warwick* et al. 1989; *Murtagh* 2005; *Nishisato* 2019). Then, given the large size of the database, the CLARA algorithm is applied and optimized. Finally, clusters' stability is assessed and a brief interpretation of clusters is provided.

4.2.1. Data preparation

Features derivation and selection First, the Body Mass Index (BMI) variable was created from both the height and weight (*Ortega* et al. 2016). Then, outliers were detected by defining for each quantitative variable thresholds above or below which values are more likely to be inaccurate. Acceptable values should be in the following ranges: 18 ≤ Age < 120, 10 ≤ BMI < 100, SBP ≤ 400, DBP ≤ 200 (*Ortega* et al. 2016, Mayo Clinic[22], French HTA (HAS) reccomandations[23]). For simplicity, patients with at least one outlier were removed from the analysis (sensitivity analyses could be performed). Quantitative variables were then discretized in order to both create variables with clinical sense and enable the use of the MCA algorithm (see Table 2).

Nota bene: the database has no missing values.

Table 2. Description of quantitative feature engineering.

Name	Description	Modalities	Accepted value range
age	Age	*age* ≤55; *age* > 55	[18, 120[
BMI	BMI	Underweight: < 18.5 *kg/m*2; Normal: between 18.5 and 24.9 *kg/m*²; Overweight: between 25.0 and 29.9 *kg/m*²; Obese: ≥30.0 *kg/m*²	[10, 100[
high sbp	High systolic blood pressure	1: SBP > 130 *mmHg*; 0: SBP ≤130 *mmHg*	≤400
high dbp	High diastolic blood pressure	1: DBP > 80 *mmHg*; 0: DBP ≤80 *mmHg*	≤200

An additional binary *hypertension* variable were created based on both *high sbp* and *high dbp* variables that are used as a proxy for patients with hypertension (*hypertension* = 1 if *high sbp* = 1 and *high dbp* = 1 ; else *hypertension* = 0 (*Desormais* et al. 2021)).

Finally, the variables selected to discriminate the population must be chosen according to their medical relevance to the context of the study. To this end, the user must always consider the results he would obtain if a variable is included or not. In particular, the user has to ask himself whether

[22] https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/diagnosis-treatment/drc-20373417#:%CB%9C:text=Your%20blood%20pressure%20is%20normal,pressure%20below%2080%20mm%20Hg

[23] Prise en charge des patients adultes atteints d'hypertension arterielle essentielle - Actualisation 2005´ https://www.has-sante.fr/upload/docs/application/pdf/2011-09/hta 2005 - recommandations.pdf

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

active discrimination of the clusters by a variable is sought: considering the example of the two common variables age and race, if they are are actively included in the clustering step, it will tend to create groups of young versus old, Caucasian versus non-Caucasian patients. If not, such variables can be kept for a passive analysis of the generated clusters and assess a posteriori a possible heterogeneity on these variables. In this use case, we removed the height, weight, systolic and diastolic blood pressure features, as they are used to create the derived features listed above and are not useful alone for clustering.

At the end of these treatments, we obtained a database of 34,134 patients described with 11 variables.

Dealing with low prevalent features and modalities Clustering variables with low prevalence are known to be challenging in data analysis, especially for techniques that are very sensitive to data and/or anomalous cases (e.g. regression analysis and factor analysis (*Fahrmeir* et al. 2013)). Most common techniques consist of either gathering rare modalities in groups of higher frequency, or to discard the concerned modalities and/or variables. Additionally, binary clustering variables with low prevalence in the study population may be discarded from the analysis or grouped with other features when appropriate.

An arbitrary threshold of 10% was set to distinguish and eliminate features with rare modalities from the clustering features. This is consistent with recommendations before using Multiple Correspondence Analysis which over-weighs rare modalities and multi-modality variables (*Di Franco* 2016; *Le Roux* et al. 2010). When possible, modalities with less than 10% of prevalence were grouped with others based on medical relevance. As a result, both the Smoking (8.3% with *smoke* = Yes) and Alcohol intake (5.2% with *alco* = Yes) variables were ruled out to cluster data and were only used for a posteriori clusters description.

Moreover, some modalities were aggregated for the 2 following variables:

- Glucose (*gluc*): modalities 2 (above normal, 8.8%) and 3 (well above normal, 9.5%) were grouped into one modality 2 (above normal)
- BMI (*BMI*): modalities "underweight" (0.5%) and "normal" were grouped into one modality "underweight & normal".

Finally, the dataset used to perform MCA contains a total number of 9 categorical variables (age, BMI, high sbp, high dbp, hypertension, gluc, gender, cholesterol, physical activity).

4.2.2. Perform Multiple Correspondence Analysis

As with other methods for factor analysis (e.g. PCA, CA, etc.), MCA was combined with cluster analysis to capture data heterogeneity, through clusters of observations in the population that show distinctive patterns (*Testa* et al. 2021; *Mitsuhiro* et al. 2015; *Van de Velden* et al. 2014; *Hwang, Montreal´* , et al. 2006; *Buuren* et al. 1989).

The number of MCA components to be used was decided using the standard scree plot by identifying the "elbow" of the curve (method widely used with PCA (*Cattell* 1966)), while constraining eigenvalues to be strictly above a threshold of 0.11 equivalent of Kaiser's rule in PCA (i.e. 1 / *C* with *C* the number of categorical variables).

Based on the scree plot (see Figure 4), 3 dimensions were chosen, the 3rd marking a clear elbow in the curve (related eigenvalue: 0.12; related percentage of variance explained: 9.8%).

Moreover and for interpretation purposes, eigenvalues were corrected using the Benzecri correction[24] to consider that the binary coding scheme used in MCA creates artificial factors and

[24] One may prefer the Greenacre adjustment to Benzecri correction, which tends to be less optimistic than the Benzecri correction. Please note that both methods are not currently implemented in the proposed R packages and must therefore be implemented by data scientists if desired. R code for Benzecri correction is provided in Appendix in section C

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

therefore reduces the inertia explained (*Greenacre* 1984). The top 3 components gather 99.9% of inertia after correcting with Benzecri method (more details in Appendix in section D).

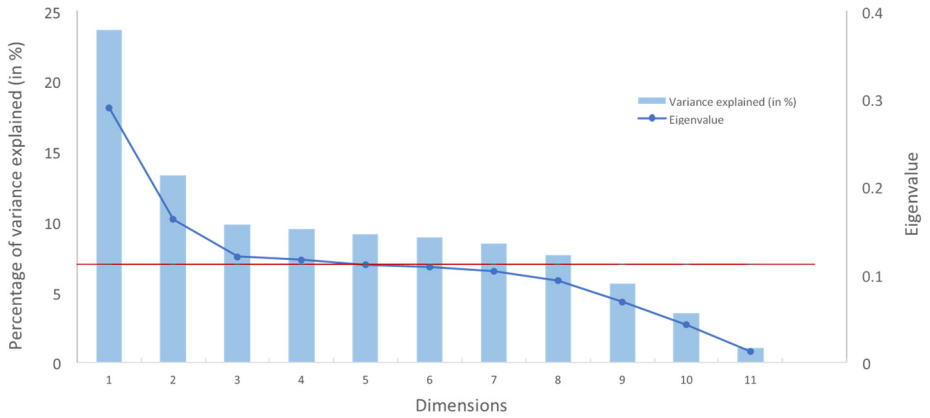


Figure 4. Scree plot of variance explained by dimension from a MCA *(In red, eigenvalue of 0.11).*

4.2.3. Clustering of the data

Parameters specification The CLARA algorithm was used through the *pamk()* function in the *FPC* R package (version 2.2-5), a reliable package for flexible procedures for clustering, and with the following main parameters:

- Distance measure: dissimilarity matrix was computed using the Manhattan distance. The latter is more robust and less sensitive to outliers than the standard Euclidean distance (Jin et al. 2010).
- Number K of clusters: From 3 to 11.
  - The number of clusters was optimized on the Average Silhouette Width (ASW) quality measure, that is an internal validity metric reflecting the compactness and separation of the clusters. The ASW is based on Silhouettes Width that were calculated for all patients in the best sample, i.e., the one used to obtain cluster medoids and generate clusters (*Rousseeuw* 1987).
  - The range of clusters to be tested was determined to enable the identification of phenotypically similar subgroups while not generating an excessive number of subgroups for interpretation.
- Number of samples and sample size: 100 samples of 5% study population size (1,706 patients).
  - Experiments have shown that 5 samples of size 40+2C (with C the number of variables in input) give satisfactory results (*Kaufman* et al. 1990). However, increasing these numbers is recommended (if feasible), to limit sampling biases and favor converging toward the best solution. Equally, the higher the sample size is, the higher it is representative of the entire dataset. We therefore recommends to pretest on his own material up to what parameters values the computation times are acceptable considering the size of the input dataset and the other steps of the workflow (including the clusters stability evaluation step, the most time consuming).

Other parameters include the non-scaling of the input data to not modify the observation space obtained from MCA.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

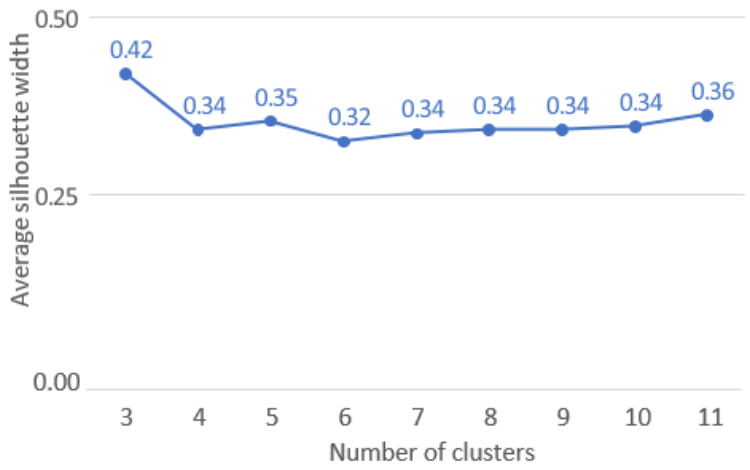


Figure 5. Average Silhouette Width for each set of clusters (K = 3 to 11).

Results: The optimal ASW was obtained for a pool of 3 clusters (ASW: 0.42, see Figure 5).

Homogeneity and separability of clusters were further studied by analyzing Silhouettes Width of patients in the best sample used to generate the clusters' medoids, using the *fviz silhouette()* function in the *factoextra* R package. As a reminder, the Silhouette Width characterizes both the cohesion of the cluster and its separation to the other clusters: a positive (respectively negative) Silhouette Width for a patient is in favor of a correct (respectively incorrect) affiliation to its own cluster.

The Figure 6 shows a high level of intra-cluster cohesion and inter-clusters separability as only few patients (in clusters 2 and 3) have negative Silhouettes. Clusterwise Silhouette Widths are also all positive (ASW of 0.42, 0.47, 0.30, for clusters 1, 2 and 3 respectively).

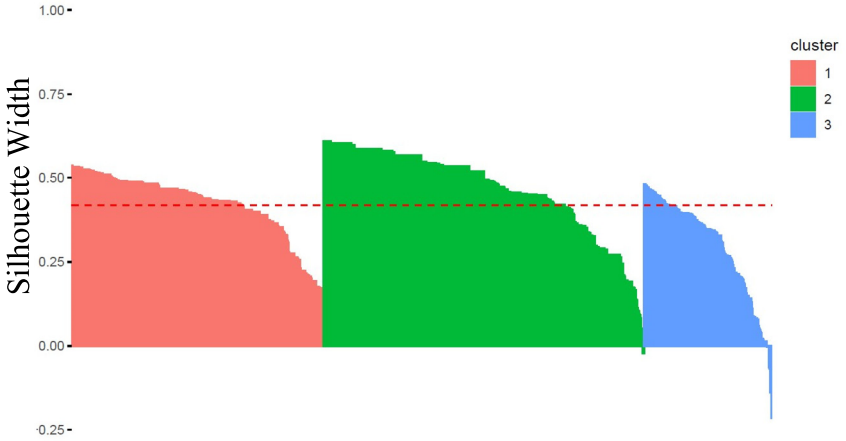


Figure 6. Silhouette Width for each patient within each of the 3 clusters *(the red line corresponds to the Average Silhouette Width).*

4.2.4. Clusters stability assessment

Parameters specification In order to evaluate clustering robustness, the clustering was performed several times on a cohort that was randomly modified. This allows to generate under perturbations new versions of the original clusters and thus to evaluate the stability of the clusters to them. The stability of the clusters is all the higher as the new versions of the clusters generated under perturbations are similar to the original clusters. The data perturbation step was performed using two approaches that may provide complementary information based on results in *Hennig* (2007): bootstrap and noise methods.

- Bootstrap approach:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

- This approach consists in performing the clustering as described in section 4.2.3 on $B = 50$ bootstrapped data (i.e. random sampling with replacement (Efron 1979; Efron and R. J. Tibshirani 1994)), using the clusterboot() function in the FPC R package (version 2.2-5).
- The Jaccard similarity metric is used to compute, for each cluster, the proximity between the clusters of patients obtained on the modified population and the original clusters. It is given by the number of patients in common between the new cluster (modified population) and the original cluster divided by the total number of distinct patients considered (i.e. present in either the new or the original cluster).
- For each cluster, the following results are provided:
  - the mean of the Jaccard similarity statistic
  - the number of times the cluster has been "dissolved", defined as a Jaccard similarity value ≤ 0.5. This value indicates instability.
  - the number of times the cluster has been "recovered", defined as a Jaccard similarity value ≥ 0.75. This value indicates stability.

There is some theoretical justification to consider a Jaccard similarity value smaller or equal to 0.5 as an indication of a "dissolved cluster", see *Hennig* (2008). Between 0.6 and 0.75, clusters may be considered as indicating patterns in the data, but which points exactly should belong to these clusters is highly doubtful.

- Noise approach:
  - This approach consists in performing the clustering as described in section 4.2.3 on $B = 50$ noisy data and for different values of noise, using the *clusterboot()* function in the *FPC* R package.
    - Level of noise values: from 1% to 10%
  - The number of times each cluster was "dissolved" and "recovered" is provided, as well as the mean of the Jaccard similarity statistic, according to the noise values.

Results: Clusters are all the more stable as the Jaccard similarity statistics and the number of recovered clusters are high, and as the number of dissolved clusters are low.

The results of the data perturbation step are:

- For the bootstrap approach, clusters 1, 2 and 3 have all 100% of Jaccard similarity statistics over 50 iterations. The 3 clusters were recovered for 100% of bootstrapped iterations, which characterizes a very high stability to resampling with replacement.
- For the noise approach, clusters 1, 2 and 3 have at worst (for 2% of noise) 100%, 98%, 96% of Jaccard similarity statistics respectively over 50 iterations. The 3 clusters were recovered for 100% of iterations and regardless of the level of noise (from 1% to 10%), which characterizes a very high stability to noise.

Regardless of the method used, the results seem therefore to be very robust and can certainly be explained by the large database's size and the small number of clusters retained in a context of synthetic data. Clusters' stability can be more variable on real cases.

It is worth noting that although the *clusterboot()* function can also provide useful results and plots of clusters' stability (histogram of Jaccard similarity statistics by cluster, summary information for each cluster, etc.) we did not provide them in this article since the obtained Jaccard similarity metrics were all around 100%.

4.2.5. Clusters interpretation

Descriptive statistics (proportions and lift values) were computed from variables included or not in the clustering step. Cluster 1 (n = 12,272 (36.0%)) groups patients who all have high values of diastolic and systolic blood pressure, and consequently hypertension. These patients are slightly more than the average with well above normal cholesterol values (18.5% versus 17.7%) and above normal glucose values (20.0% versus 18.3%). On the contrary, patients from clusters 2 and 3 (n = 15,477 (45.3%) and n = 6,385 (18.7%) respectively) are between 81% and 87% to have normal values of diastolic and systolic blood pressures, and none have hypertension. In contrast with cluster 2,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

patients from cluster 3 are many more with well above normal cholesterol values (26.6% versus 8.5%) and above normal glucose values (57.3% versus 0.8%).

Patients from clusters 1 and 2 are overall younger than cluster 3 (age ≤ 55: 54.7% and 51.8% versus 65.9%). Patients from clusters 1 and 3 are overall more obese than cluster 2 (41.4% and 44.5% versus 21.8%).

To summarize, among patients with a cardiovascular disease, cluster 1 gathers patients with hypertension, cluster 2 gathers patients healthier (although about the same age than cluster 1) and cluster 3 gathers slightly older patients with cholesterol and high levels of glucose (although no hypertension). Interestingly, description of cluster 1 is consistent with a poorer lifestyle (lift values of 1.21 and 1.28 for Smoke and Alcohol respectively) although this did not actively participate in partitioning. See Table 3 below for more details.

Table 3. Prevalence and lift values of each modality and by cluster (C1, C2 and C3 stand for Cluster 1, Cluster 2 and Cluster 3 respectively. Dark blue: ≤ 0.5; light blue: ≤ 1.0; yellow: ≤ 1.5; green: >1.5. Lift is defined as % in the cluster versus in the cohort.).

	Prevalence (% of patients)				Lift values		
Modality	**C1**	**C2**	**C3**	Cohort			
	n = 12272 (36.0%)	n = 15477 (45.3%)	n = 6385 (18.7%)	n = 34134 (100%)	C1	C2	C3
Female	62.1	64.2	71.4	64.8	0.96	0.99	1.10
Male	37.9	35.8	28.6	35.2	1.08	1.02	0.81
Cholesterol normal	60.6	90.7	16.5	66.0	0.92	1.37	0.25
Cholesterol above normal	20.8	8.5	26.6	16.3	1.28	0.52	1.63
Cholesterol well above normal	18.5	0.7	56.9	17.7	1.05	0.04	3.22
Glucose normal	80.0	99.2	42.7	81.7	0.98	1.21	0.52
Glucose above normal	20.0	0.8	57.3	18.3	1.10	0.04	3.13
Physical activity	81.0	76.8	79.6	78.8	1.03	0.97	1.01
Age ≤55	45.3	48.2	34.1	44.5	1.02	1.08	0.77
Age > 55	54.7	51.8	65.9	55.5	0.99	0.93	1.19
BMI obese	41.4	21.8	44.5	33.1	1.25	0.66	1.34
BMI overweight	36.5	37.3	36.1	36.8	0.99	1.01	0.98
BMI normal or underweight	22.1	40.9	19.5	30.1	0.73	1.36	0.65
High Systolic blood pressure	100.0	14.3	18.5	45.9	2.18	0.31	0.40
High Diastolic blood pressure	100.0	12.9	16.8	44.9	2.23	0.29	0.37
Hypertension	100.0	0	0	36.0	2.78	0	0
Smoke	10.1	7.2	7.6	8.3	1.21	0.87	0.91
Alcohol	6.6	3.8	5.7	5.2	1.28	0.74	1.09

5. Discussion

In this section we will first discuss some limitations of the *Qluster* workflow and possible enhancements, then discuss choices of parameters and the practical use of this workflow.

5.1. Limitations and proposition for enhancing this workflow

When large data is too large As often in data mining, one limit concerns the size of the data. It is clear that for massive data, where the number of rows is several million, specific algorithms such as grid-based methods or canopy pre-clustering algorithm (*McCallum* et al. 2000) are needed for the algorithms to scale up.

More specifically, in such case, factor analysis may be impossible to calculate, as it requires to make matrix calculations and to invert matrix of size n * p (n individuals, p binary variables). Please note though that in the case of categorical variables, one may prefer to use the anglo-saxon MCA method that applies the CA algorithm on a Burt table (p * p) instead of the complete disjunctive table (n * p), which is more efficient in computing time and thus more appropriate for large data (also implemented in the *MCA()* function in *FactoMineR* (*Greenacre* 2007)). Equally, in the case of very large

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

data, CLARA algorithm may be too time-consuming to be calculated as we still need to maintain enough samples and observations per sample for representativeness. For all these reasons, one suggests simply analyzing a random sample of the original dataset that is likely to be very representative of the latter while allowing the use of the *Qluster* workflow. Please note that *PCA()* and *FAMD()* are known to take more time for computation than *MCA()*. As a consequence, the notion of "large" data may depend on the nature of data (continuous, categorical, mixed). One also suggests (when possible) in a data preparation step to converge into one type of data (continuous only or categorical only). This is especially true for mixed data where the upstream scaling of the data can be challenging, and where the computation times by FAMD are more important. Alternatives may consist in not using the proposed workflow but algorithms which go fast on (very) large data such as Mini Batch *K*-means used on continuous variables or on one-hot-encoded categorical variables. However, in addition to the fact that it relies on the Euclidean distance only, these strategies may not allow the beforehand use of factor analysis because of data's size, as well as to easily and properly assess clusters' stability, which hinders confidence and support in results. One therefore recommends the first option.

Conversely, when the number of columns is greater than the number of rows ($p > n$), the dimension reduction step via factor analysis methods makes very sense to easily manage the high dimensionality of the data. However, in the most extreme cases where $p >> n$, standard factor methods may fail to yield consistent estimators of the loading vectors. Also, the results may be difficult to interpret. In such situations, standardized methods may be a solution to improve the robustness, consistency, and interpretability of results (e.g., penalized PCA, *Lee* et al. 2012). It is also recommended that, when possible, a subset of the variables be selected strictly prior to analysis.

Generalizability of this workflow to missing data Missing values management is not covered in this workflow, and it is therefore assumed that no missing values are present in the dataset. Indeed, both factor methods (PCA, MCA, FAMD) and proposed clustering methods (PAM, CLARA, ...) require data without missing values. However, this workflow can be easily generalized to missing data, using the same *missMDA* package as for performing the selection of the optimal number of dimensions in factor analysis, in order to impute in a first step missing values using factor methods. The latter are state-of-the-art methods for handling missing values (e.g. function *imputePCA()*, *imputeMCA()* and *imputeFAMD()* for simple imputations (*Audigier* et al. 2013)) and can thus easily be integrated and/or used in an automated workflow to handle missing data. In addition, this R package makes it possible to perform multiple imputations (*MIPCA()*, *MIMCA()* and *MIFAMD()*) for assessing incertitudes from imputed values and increasing confidence in results (*Josse, Pages` ,* et al. 2011). In this sense, the *Qluster* workflow is therefore adapted for handling missing data (see in Appendix in section E an example of the *Qluster* workflow adapted for handling missing values).

Discussion on using factor analysis as a first step As mentioned on several occasions, factor analysis allows the transformation of structured data of all kinds into continuous data, while dealing with large, collinear, noisy, high-dimensional data. It also facilitates clustering by aggregating groups of homogeneous information within dimensions. Nevertheless, it cannot always be guaranteed that the results will be "better" or "as good" with factor analysis in the clustering process. Similarly, the choice of factor analysis in this workflow comes with drawbacks that include the following items:

- The packages used cannot handle the ordinal nature of the variables. The latter must be treated as categorical or continuous.
- The observations x components matrix is continuous, although some raw variables could be categorical. This prevents the user from favoring (respectively, not favoring) positive cooccurrence over negative co-occurrence via the Jaccard (respectively, Hamming) distance

Alternatives can be to perform data dimension reduction using features selection methods, or manually, by grouping, transforming and/or deleting variables based on clinical expertise.

Discussion on using a single *K*-medoid algorithm In order to offer a simple, yet generic and robust workflow to make practical use of the same methodology in many applications, we performed an arbitrary (but well chosen) selection of both algorithms and software packages. In particular, the choice to use the PAM/CLARA algorithm is based on many aspects such as the fact that it is:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

- one of the best known, studied and used algorithm by the community, for general purpose, - adapted to the continuous setting (i.e., the most mature in the literature).
- meant for the most frequent use case of clustering (i.e., hard partitioning),
- suitable to the Manhattan distance, a less sensitive to outliers distance, unlike its counterpart on the euclidean distance (K-means),
- deterministic, thanks to its internal medoid initialization procedure, unlike the basic K-means algorithm that may lead to inconsistent or non-reproducible clusters
- requiring few parameters to set up (e.g. conversely to BIRCH and DBSCAN, see Fahad et al.(2014)),
- very well implemented within a recognized reference R package (the FPC package) facilitating its use within a complete and robust clustering approach,
- usable within the same R function (pamk()) regardless of the volume of data.

Yet, it is clear that other algorithms than the ones chosen could be routinely used instead, including those present in the *FPC* R package to facilitate its integration within the workflow (e.g. DBSCAN and HAC). In particular, it is well known that with non-flat geometry and/or uneven cluster size, DBSCAN is more appropriate than *K*-means and PAM. Equally, if the final goal is to obtain a hierarchy rather than a unique hard partition, the user will be encouraged to prefer an algorithm such as HAC, which can easily be used with the proposed packages in this workflow. However, the presence of additional parameters to tune or the lack of compatibility with massive data would require the workflow to become more complex. Also, this workflow is not meant to replace more in-depth work by the data scientist to find what is optimal on a specific case study. More experienced data scientists can use the generic *Qluster* workflow for a first look at the data, but are encouraged to adapt the general principles of this workflow to their own case study (e.g., finding the most suitable algorithm, etc.). Such adaptations would be out-of-scope of this workflow in the sense of the initial objectives: genericity of applications while preserving the simplicity of implementation and reliability/robustness of metholodogy.

Equally, the user may want to benchmark several clustering algorithms as suggested in *Hennig* (2020). The comparison of methods solutions can be based on information measures (e.g. entropy, mutual information, etc.), internal validity measures (e.g. silhouette, see section 2.4.), set-matching (i.e., mapping each cluster from the first clustering to the most similar cluster in the second clustering and computing recall, precision or some other measure) and pair counting (including dedicated visualization tools, see *Achtert* et al. (2012)). Some of these strategies are directly implemented in the *clusterbenchstats()* function from the *FPC* R package or in the *clValid()* function of the *clValid* R package. However, as our goal is to propose a simple-to-use workflow, this complexification - which would also greatly impact computing times and memory capacities - is left to the user's discretion. Moreover, multiplying the algorithms and combination of parameters forces one to rely more heavily on a purely statistical criterion (e.g. ASW) to select the "best" clustering of the data, although this may not reflect the best partitioning in the clinical sense. Indeed, ASW remains a criterion characterizing the average separability over all the clusters, and its optimum may miss (the set of) results that is clinically relevant and/or useful for the desired objective[25]. If the data scientist wishes to compare different algorithms, we would rather recommend to fully investigate the results from a well-chosen first algorithm (here PAM/Clara), before challenging it with others, in order to be less dependent on the sole selection criterion of the ASW. This paper thus takes the opposite view of the auto-ML literature by first advocating a full investigation of a parsimonious workflow made of well-chosen algorithms, rather than directly a broad coverage of algorithmic possibilities. Nevertheless, readers may be interested in recent areas of research around meta-clustering (*Caruana* et al. 2006) and ensemble clustering methods (*Greene* et al. 2004; *Alqurashi* et al. 2019). The first one aims to produce

[25] Similarly, the user may want to compare several methods for selecting the optimal number of clusters, including other direct methods (e.g. elbow method on the total within-cluster sum of square) or methods based on statistical testing (e.g. gap statistic)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

several partitioning results in order for the user to select those which are most useful. The second one is intended to combine the clustering of several methods with the aim of proposing a consensual result.

Discussion on the clusters' stability assessment step Bootstrapping and noising methods were chosen in the workflow for both their availability in the same function *clusterboot()* from the same package as for *pamk()*, and for their complementarity as recommended by *Hennig* (2007). Nevertheless, other methods may also be used as sensitivity analyses, including those proposed in the same *FPC* package. Furthermore, although this step allows for the clusters to be assessed, data scientists should keep in mind that stability is not the only important validity criterion - clusters obtained by very inflexible clustering methods may be stable but not valid, as discussed in *Hennig* (2008). Finally, although several choices were made to try to manage outliers as best as possible, such as using a *k*-medoid algorithm and the Manhattan distance, the *Qluster* workflow does not fully address the issues related to outliers and extreme values. One solution may be to define threshold values to manually detect extreme values as a pre-processing step (as in the case study in section 4), or to use more sophisticated statistical methods such as *Yang* et al. (2021).

Discussion on the clusters' interpretation Clusters' description is not covered in the *Qluster* workflow. However, many methods exist to interpret clusters (see section 2.3). Data scientists can easily generalize *Qluster* to the description of clusters by using the functions already present in the *FPC* package in order not to make the workflow too complex, such as *plotcluster()* and *cluster.varstats()* following methodologies recommended by *Hennig* (2004).

Discussion on the types of data that are supported by the Qluster workflow Although general, the *Qluster* workflow does not cover all types of data and it is clear that for medical imaging data, omics data or data in the form of signals, dedicated approaches must be considered. Nevertheless, most tabular data can be processed using the *Qluster* workflow. In this respect, and although the *Qluster* workflow was specifically designed in the context of healthcare data analysis, it can easily be applied in other fields.

5.2. Discussion and recommendation on the practical use of the workflow

Use of clusters stability as a criterion to be optimized Cluster stability assessment could be considered as a criterion to be optimized, by iterating on this step in order to make this property an integral part of the clustering process itself. For example, stability measures could be used to select the optimal number of clusters, assuming that the clustering results are more stable with the correct number of clusters (*Pasi Franti¨* 2020).

Attention should be paid, however, to the fact that the bootstrap and noise methods are more computationally and thus time-consuming than simple methods such as deleting variables one by one (methods used on biological measurements and proposed in the *ClValid* R package). Also, it may not be obvious to optimize the clustering on clusters' stability if the 2 proposed methods do not give similar results. For example, relatively to the noising method, the bootstrap method is more likely to give stable results with the increase in the volume of the dataset in the case of PAM, and the increase in the % of representativeness of the samples in the case of CLARA.

What if results are not satisfying The question of the ultimate relevance of clusters has not been covered in this workflow. It is worth pointing out that an absence of results may be a result in itself, as it can characterize a population that cannot be described in several homogeneous subgroups (either because such subgroups do not exist, or because the variables used do not allow to find them). Nevertheless, it is clear that, as in the Data Mining process, we can consider looping back on this workflow by changing certain parameters if the results are not satisfactory or if an important criterion of the clustering was not taken into account at the beginning (for example the maximum number of clusters, etc.). More generally, the data scientist is encouraged to keep in mind that the final objective of clustering is often the clinical relevance and usefulness of the results generated. In this sense and as mentioned in 5.1, it is not forbidden to relax a purely statistical criterion, such as the ASW (whose optimum may miss some relevant subgroups as it is an indicator of the overall separability) to better

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

represent the diversity of the population studied, or to favor the generation of hypotheses in the case where the statistical optimum only gives broad results not enough specific for the initial objective.

In the same spirit, negative Silhouette values are too pejoratively considered in the cluster's validity analysis (interpreted as clustering failures). Indeed, negative Silhouettes characterize patients who are, on average, closer to patients from another cluster than to the patients of their own cluster. Therefore, patients with a negative Silhouette may be informative of potential relationships between clusters, and should therefore be considered as potential additional information about disease history and phenotypic complexity, such as one cluster that is the natural evolution of another. Hence, it is recommended that an analysis of patients with negative Silhouettes be included in the workflow to better assess whether they are a reflection of "bad" clustering or the key to better understanding the disease.

What if the optimum number of clusters is the minimum of range *K*? In the case where the optimal number of clusters is the minimum of the range of *K* (as in our example in section 4), we recommend (if appropriate) that data scientists be testing for lower values of *K* to challenge the obtained optimum. Equally, if the optimum is obtained for $K = 2$, data scientists should test whether the dataset should be split into two clusters, using the Duda-Hart test that tests the null hypothesis of homogeneity in the whole dataset. This can be done using the same *pamk()* function by setting up the minimum of *K* to 1, or directly using the *dudahart2()* function (also in *FPC* R package). In any case, if the primary objective is to provide fine-grained knowledge of the study population, it will still be possible to provide results with the optimal *K* that was initially obtained, keeping in mind that the levels of inter-cluster separability and intra-cluster homogeneity are not really higher than those that would be obtained with a smaller number of clusters.

Using this workflow routinely The *Qluster* workflow is highly subject to automation for data scientists and companies that need a routine way to cluster clinical data. Indeed, data scientists may create a main function for applying this workflow, including by setting the nature of data (categorical/continuous/mixed), the volume (normal/large), and parameters related to each called function. It is worth mentioning, however, that the quality of the input data or the structure of the groups to be found are factors that may not allow the present workflow to identify relevant results every time. In this case, the data scientist can refer to the indications given above or, if necessary, consider an approach more adapted to his data.

6. Conclusion

In this article, we propose *Qluster*, a practical workflow for data scientists because of its **genericity of application** (e.g. usable on small or big data, on continuous, categorical or mixed variables, on database of high-dimensionality or not, etc.) while preserving the **simplicity** of implementation and use (e.g. need for few packages and algorithms, few parameters to tune, …), and the **robustness and reliability** of the methodology (e.g. evaluation of the stability of clusters, use of proven algorithms and robust packages, management of noisy or multicollinear data, etc.). It therefore does not rely on any innovative approach per se, but rather on a careful selection and combination of state-of-the-art clustering methods for practical purposes and robustness.

One of the underlying motivations is the observation of research teams, including statisticians/data scientists in contract research organizations that provide support to healthcare industries, who have the responsibility to conduct clustering analyses but are still little experienced in using them. They are also confronted with a very large literature, and a very large number of algorithms and implementations, which makes the exercise difficult. We believe that *Qluster* can improve (1) the quality of analyses carried out as part of such studies (thanks to *Qluster*'s criteria for **robustness** and **reliability**), promote and ease clustering studies (thanks to *Qluster*'s **genericity** and **simplicity** of use), and increase the skills of some of the statisticians/data scientists involved (thanks to the literature review provided and the general principles of *Qluster*). This workflow can also be used by more experienced data scientists for initial explorations on the data before designing more in-depth analyses.

 doi:10.20944/preprints202205.0215.v3

Finally, this workflow can be fully operationalized, using either scripted tools or a Data Science platform supporting the use of R packages. As an illustrative example, we made an implementation on the Dataiku platform of the *Qluster* workflow to process a kaggle dataset (see in Appendix in section B). This implementation is usable on the free edition and is made available on request (email: contact@quinten-france.com).

Supplementary Material: The Supplementary Material for this article can be found in Appendix.

Author contributions: All authors participated in writing the manuscript. All authors contributed to the revision of the manuscript, read and approved the submitted version. MR and PG contributed very significantly throughout this work. CE and JDZ was the main contributors to both the writing and the methodology.

Acknowledgment: The team would like to thank Dr. Martin Montmerle for initiating the scientific study that led to this research work. The team would also like to thank Dr. Martin Montmerle, Vincent Martenot, Valentin Masdeu, Pierre Tang and Sam Ekhtiari for their careful review of the article. Finally, the team would like to thank Quinten for providing the opportunity to conduct this work.

Conflict of Interest: The authors declare the following competing interests. Employment: CE, MR and PG are employed by Quinten.

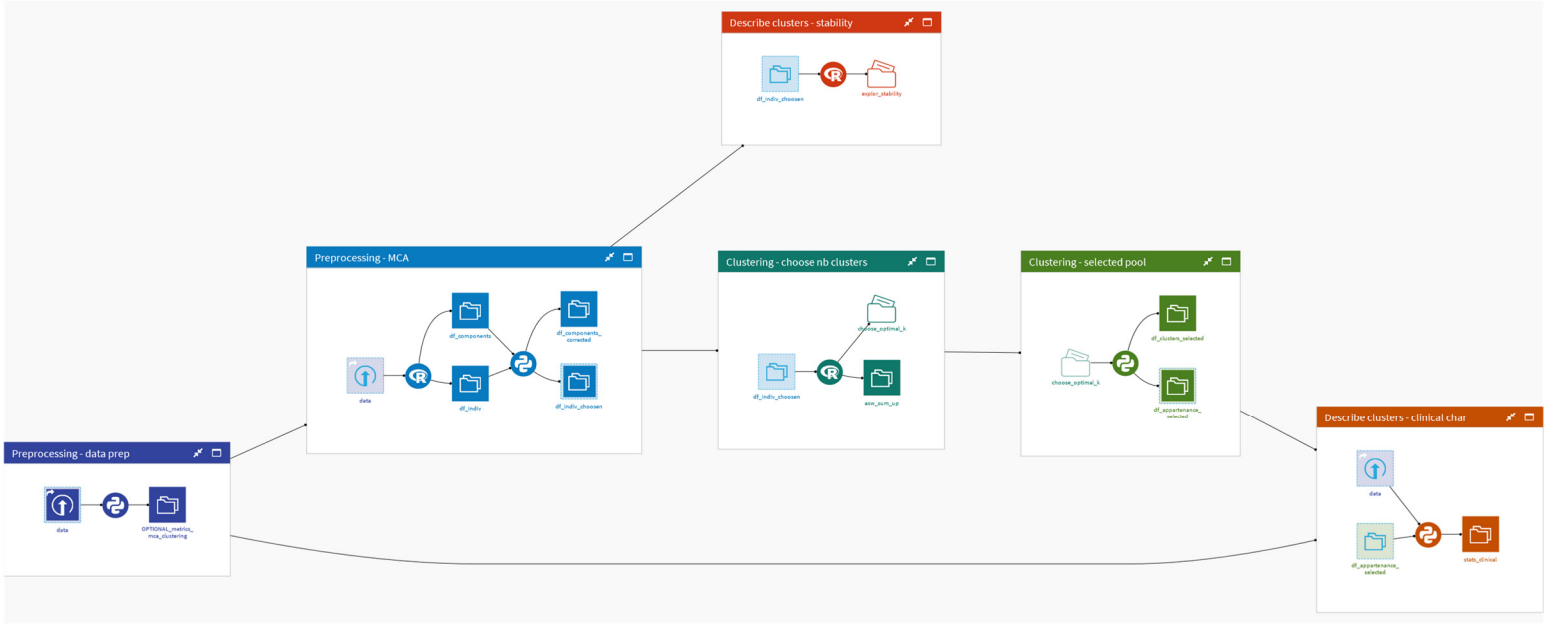
Appendix

doi:10.20944/preprints202205.0215.v3

A. *Description of algorithms in selected libraries on some of the criteria defining genericity, ease of implementation and use, and robustness*

Algorithms	Categories	*Genericity*			*Ease of implementation*						*Robustness*	
		handling large data / velocity	data type	handling high dimensionnality	Number of required parameters	Handy packages					handling noisy data	handling multicollinearity
						cluster	clue	ClValid	FPC	Sklearn		
Affinity propagation	Partitionning-based	-	continuous	No	2					x	No	No
CLARA	Partitionning-based	++	continuous	Partially	1	x		x	x		No	No
KMEANS	Partitionning-based	+	continuous	No	1			x	x	x	No	No
KMEDOIDS	Partitionning-based	-	continuous	Partially	1		x				Yes	No
KPROTOTYPES	Partitionning-based	-	mixed	No	2		x				No	No
Mean Shift	Partitionning-based	-	continuous	No	1					x	No	No
Mini Batch KMEANS	Partitionning-based	++	continuous	Yes	1					x	No	No
PAM	Partitionning-based	-	continuous	No	1	x	x	x	x		No	No
SOTA	Model-based	-	continuous	No	1			x			No	No
AGNES	Hierarchical	-	continuous	No	1	x		x		x	No	No
BIRCH	Hierarchical	++	continuous	Partially	2					x	Yes	No
DIANA	Hierarchical	-	continuous	No	1	x		x			No	No
MONA	Hierarchical	-	binary	No	0	x					No	No
DBSCAN	Density-based	+	continuous	No	2				x	x	No	No
OPTICS	Density-based	+	continuous	No	2						Yes	No

B. Example of an implementation of the Qluster workflow on the Dataiku platform



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

C. R Code to implement Benzecri correction from MCA eigenvalues

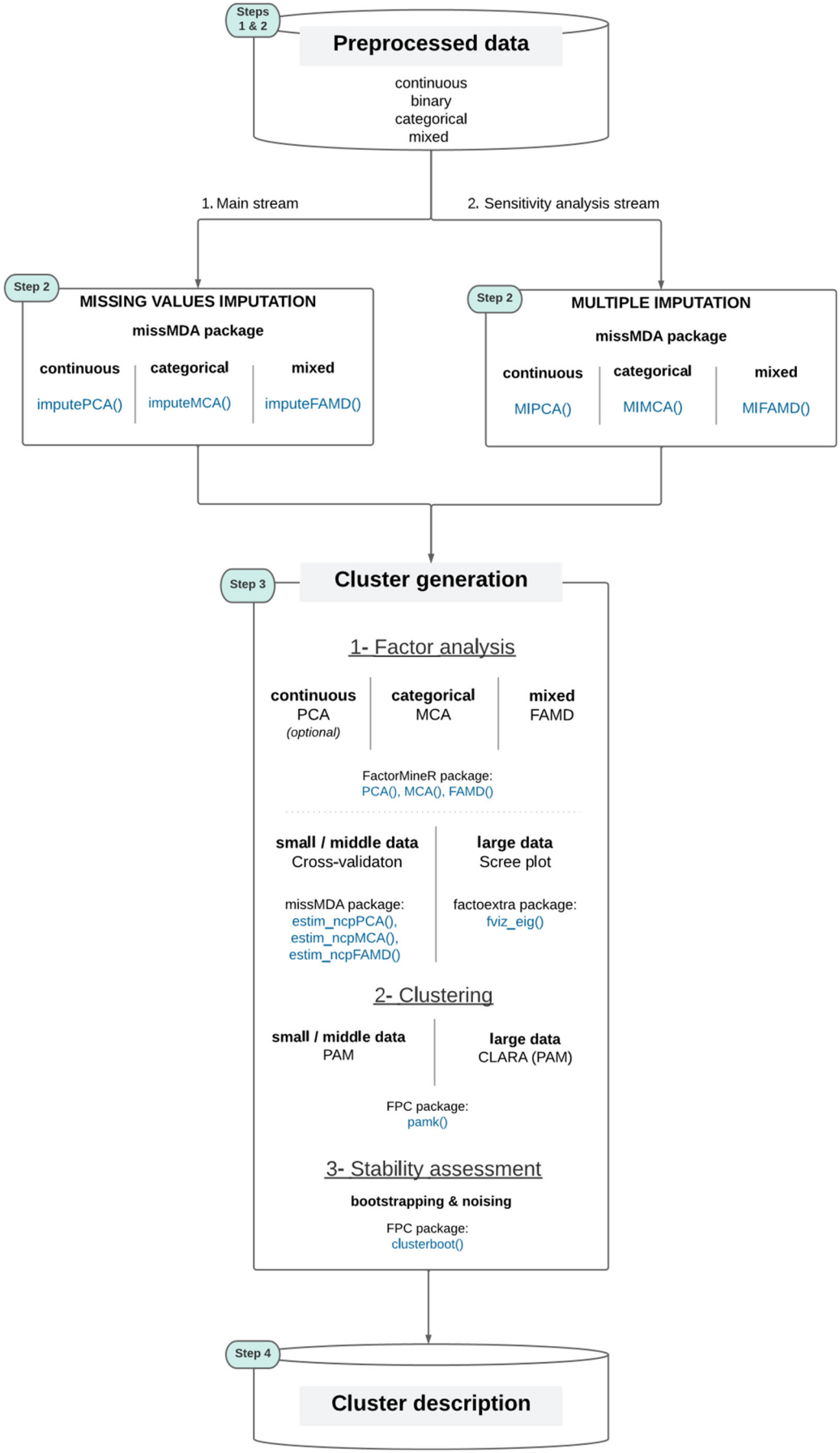
```
correction_benz = function(eig, K){
###
#Function to correct MCA eigenvalues using Benzecri correction
#Arguments:
    #    eig: vector of eigenvalues from MCA()
    #    K: number of qualitative variables (binary or categorical)
#Returns: eigenvalues and variances corrected
###
# Benzecri correction
eig_benz <- rep(0, length(eig))
selection <- eig > 1/K
eig_benz[selection] <- ((K/(K-1))*(eig[selection] - 1/K))^2

# Save results# save results
var_benz <- 100*eig_benz/sum(eig_benz)
df_mca_benz <- cbind(eig_benz, var_benz, cumsum(var_benz))
colnames(df_mca_benz) = c("eig_benz", "%_var_benz", "cum_%_var_benz")

return(df_mca_benz)
}
```

D. Eigenvalues and variances explained with and without Benzecri correction

Dimension	Eigenvalue	Variance explained (in %)	Cumulative variances explained (in %)	Eigenvalue corrected by Benzecri	Variances explained corrected by Benzecri (in %)	Cumulative variances explained corrected by Benzecri (in %)
1	0.29	23.6	23.6	0.04	92.1	92.1
2	0.16	13.3	36.9	< 0.01	7.7	99.7
3	0.12	9.8	46.7	< 0.01	0.2	99.9
4	0.12	9.5	56.1	< 0.01	0.1	100.0
5	0.11	9.0	65.2	–	–	100.0
6	0.11	8.8	74.0	–	–	100.0
7	0.10	8.4	82.4	–	–	100.0
8	0.09	7.6	90.0	–	–	100.0
9	0.07	5.6	95.6	–	–	100.0
10	0.04	3.5	99.0	–	–	100.0
11	0.01	1.0	100.0	–	–	100.0

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2023 doi:10.20944/preprints202205.0215.v3

E. Example of the *Qluster* workflow adapted for handling missing values.



References

Achtert, E., S. *Goldhofer*, H-P. *Kriegel*, E. *Schubert* and A. *Zimek*, "Evaluation of Clusterings -- Metrics and Visual Support," 2012 IEEE 28th International Conference on Data Engineering, 2012, pp. 1285-1288, doi: 10.1109/ICDE.2012.128.

Ahmad, Amir and Shehroz S. *Khan*. "Survey of state-of-the-art mixed data clustering algorithms". In: *IEEE Access* 7 (2019), pp. 31883–31902. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2903568. (visited on 09/09/2021).

Aljalbout, Elie, Vladimir *Golkov*, Yawar *Siddiqui*, Maximilian *Strobel*, and Daniel *Cremers*. "Clustering with Deep Learning: Taxonomy and New Methods". In: *arXiv:1801.07648 [cs, stat]* (Sept. 13, 2018). arXiv: 1801.07648. URL: http://arxiv.org/abs/1801.07648 (visited on 09/09/2021).

Alqurashi, T. and W., *Wang*. Clustering ensemble method. Int. J. Mach. Learn. & Cyber. 10, 1227–1246 (2019). https://doi.org/10.1007/s13042-017-0756-7

Altman, Naomi and Martin *Krzywinski*. "Clustering". In: *Nature Methods* 14.6 (June 1, 2017). Bandiera _abtest: a Cg type: Nature Research Journals Number: 6 Primary atype: News Publisher: Nature Publishing Group Subject term: Data mining;Data processing Subject term id: data-mining;data-processing, pp. 545–546. ISSN: 1548-7105. DOI: 10.1038/nmeth.4299. URL: https://www.nature.com/articles/nmeth.4299 (visited on 09/09/2021).

Arabie, P. and L. *Hubert*. "Cluster analysis in marketing research". In: *Advanced Methods of Marketing Research* (1994), pp. 160–189. URL: https://ci.nii.ac.jp/naid/10026666029/en/.

Arbelaitz, Olatz, Ibai *Gurrutxaga*, Javier *Muguerza*, Jesus M.´ *Perez´* , and Inigo˜ *Perona*. "An extensive comparative study of cluster validity indices". en. In: *Pattern Recognition* 46.1 (Jan. 2013), pp. 243–256. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2012.07.021. URL: https: //www.sciencedirect.com/science/article/pii/S003132031200338X (visited on 12/21/2021).

Arthur, D., and *Vassilvitskii*, S. "k-means++: the advantages of careful seeding". Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. (2007). pp. 1027-1035.

Audigier, Vincent, Franc¸ois *Husson*, and Julie *Josse*. "A principal components method to impute missing values for mixed data". In: *arXiv:1301.4797 [stat]* (Feb. 19, 2013). arXiv: 1301.4797. URL: http://arxiv.org/abs/1301.4797 (visited on 09/09/2021).

Bandalos, Deborah L. and Meggen R. *Boehm-Kaufman*. "Four common misconceptions in exploratory factor analysis". In: *Statistical and methodological myths and urban legends: Doctrine, verity and fable in the organizational and social sciences*. New York, NY, US: Routledge/Taylor & Francis Group, 2009, pp. 61–87. ISBN: 978-0-8058-6238-6 978-0-8058-6237-9.

Bertsimas, Dimitris, Agni *Orfanoudaki*, and Holly *Wiberg*. "Interpretable clustering: an optimization approach". In: *Machine Learning* 110.1 (Jan. 1, 2021), pp. 89–138. ISSN: 1573-0565. DOI: 10.1007/s10994-020-05896-2. URL: https://doi.org/10.1007/s10994-020-05896-2 (visited on 03/17/2021).

Bezdek, J.C., Robert *Ehrlich*, and William *Full*. "FCM: The fuzzy c-means clustering algorithm". en. In: *Computers & Geosciences* 10.2 (Jan. 1984), pp. 191–203. ISSN: 0098-3004. DOI: 10.1016/0098-3004(84)90020-7. URL: https://www.sciencedirect.com/science/ article/pii/0098300484900207 (visited on 12/05/2021).

Bezdek, J.C. and N.R. *Pal*. "Some new indexes of cluster validity". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 28.3 (June 1998). Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), pp. 301–315. ISSN: 1941-0492. DOI: 10.1109/3477.678624.

Bock, H. H. "On the Interface between Cluster Analysis, Principal Component Analysis, and Multidimensional Scaling". en. In: *Multivariate Statistical Modeling and Data Analysis: Proceedings of the Advanced Symposium on Multivariate Modeling and Data Analysis May 15–16, 1986*. Ed. by H. *Bozdogan* and A. K. *Gupta*. Theory and Decision Library. Dordrecht: Springer Netherlands,1987, pp. 17–34. ISBN: 978-94-009-3977-6. DOI: 10.1007/978-94-009-3977-6_2. URL: https://doi.org/10.1007/978-94-009-3977-6_2 (visited on 12/16/2021).

Bousquet, Philippe Jean, Philippe *Devillier*, Abir *Tadmouri*, Kamal *Mesbah*, Pascal *Demoly*, and Jean *Bousquet*. "Clinical Relevance of Cluster Analysis in Phenotyping Allergic Rhinitis in a Real-Life Study". In: *International Archives of Allergy and Immunology* 166.3 (2015). Publisher: Karger Publishers, pp. 231–240. ISSN: 1018-2438, 1423-0097. DOI: 10.1159/000381339. URL: https://www.karger.com/Article/FullText/381339 (visited on 03/16/2021).

Bro, R., K. *Kjeldahl*, A. K. *Smilde*, and H. A. L. *Kiers*. "Cross-validation of component models: A critical look at current methods". en. In: *Analytical and Bioanalytical Chemistry* 390.5 (Mar. 2008), pp. 1241–1251. ISSN: 1618-2650. DOI: 10.1007/s00216-007-1790-1. URL: https://doi.org/10.1007/s00216-007-1790-1 (visited on 12/05/2021).

Brock, Guy, Vasyl *Pihur*, Susmita *Datta*, and Somnath *Datta*. "clValid: An R Package for Cluster Validation". en-US. In: (Mar. 2008). URL: https://www.jstatsoft.org/article/ view/v025i04 (visited on 12/16/2021). *Buuren*, Stef van and Willem J. *Heiser*. "Clusteringn objects intok groups under optimal scaling of variables". In: *Psychometrika* 54.4 (Sept. 1, 1989), pp. 699–706. ISSN: 1860-0980. DOI: 10.1007/BF02296404. URL: https://doi.org/10.1007/BF02296404 (visited on 09/09/2021).

Calinski´, T. and J *Harabasz*. "A dendrite method for cluster analysis". In: *Communications in Statistics* 3.1 (Jan. 1, 1974). Publisher: Taylor & Francis eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101, pp. 1–27. ISSN: 0090-3272. DOI: 10.1080/03610927408827101. URL: https://www.tandfonline. com/doi/abs/10.1080/03610927408827101 (visited on 09/09/2021).

Caruana, R., *Elhawary*, *M.*, *Nguyen*, N. and C. *Smith*, "Meta Clustering," Sixth International Conference on Data Mining (ICDM'06), 2006, pp. 107-118, doi: 10.1109/ICDM.2006.103.

Cattell, Raymond B. "The Scree Test For The Number Of Factors". In: *Multivariate Behavioral Research* 1.2 (Apr. 1, 1966). Publisher: Routledge eprint: https://doi.org/10.1207/s15327906mbr0102 10, pp. 245–276. ISSN: 0027-3171. DOI: 10.1207/ s15327906mbr0102_10. URL: https://doi.org/10.1207/s15327906mbr0102_10 (visited on 09/09/2021).

Celebi, M Emre. *Partitional clustering algorithms*. Springer, 2014.

Ciampi, Antonio and Yves *Lechevallier*. "Clustering Large, Multi-level Data Sets: An Approach Based on Kohonen Self Organizing Maps". In: *Principles of Data Mining and Knowledge Discovery*. Ed. by Djamel A. *Zighed*, Jan *Komorowski*, and Jan *Zytkow˙* . Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, pp. 353–358. ISBN: 978-3-540-45372-7. DOI: 10.1007/3-540-45372-5_36.

Clausen, Sten Erik. *Applied Correspondence Analysis: An Introduction*. es. Google-Books-ID: FsCXR3uk _0gC. SAGE, June 1998. ISBN: 978-0-7619-1115-9. "Cluster analysis". In: *Wikipedia*. July 18, 2021. URL: https://en.wikipedia.org/w/ index.php?title=Cluster_analysis&oldid=1034255231.

Costa, Patr´ıcio Soares and Nuno *Sousa*. *The Use of Multiple Correspondence Analysis to Explore Associations between Categories of Qualitative Variables in Healthy Ageing*. 2013. URL: https://www.hindawi.com/journals/jar/2013/302163/ (visited on 12/16/2021).

Datta, Susmita and Somnath *Datta*. "Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes". In: *BMC Bioinformatics* 7.1 (Aug. 2006), p. 397. ISSN: 1471-2105. DOI: 10.1186/1471-2105-7-397. URL: https://doi.org/10.1186/1471-2105-7-397 (visited on 12/16/2021).

De Soete, Geert and J. Douglas *Carroll*. "K-means clustering in a low-dimensional Euclidean space". en. In: *New Approaches in Classification and Data Analysis*. Ed. by Edwin *Diday*, Yves *Lechevallier*, Martin *Schader*, Patrice *Bertrand*, and Bernard *Burtschy*. Studies in Classification, Data Analysis, and Knowledge Organization. Berlin, Heidelberg: Springer, 1994, pp. 212–219. ISBN: 978-3-642-51175-2. DOI: 10.1007/978-3-642-51175-2_24.

DeSarbo, Wayne S., Daniel J. *Howard*, and Kamel *Jedidi*. "Multiclus: A new method for simultaneously performing multidimensional scaling and cluster analysis". en. In: *Psychometrika* 56.1 (Mar. 1991), pp. 121–136. ISSN: 1860-0980. DOI: 10.1007/BF02294590. URL: https: //doi.org/10.1007/BF02294590 (visited on 12/05/2021).

Desormais, Ileana and Williams *Bryan*. "2018 ESC/ESH Guidelines for the management of arterial hypertension". In: *Journal of Hypertension* (). URL: https://journals.lww.com/ jhypertension/Fulltext/2018/10000/2018_ESC_ESH_Guidelines_for_ the_management_of.2.aspx (visited on 12/16/2021).

Di Franco, Giovanni. "Multiple correspondence analysis: one only or several techniques?" In: *Quality & Quantity* 50.3 (May 1, 2016), pp. 1299–1315. ISSN: 1573-7845. DOI: 10.1007/s11135-015-0206-0. URL: https://doi.org/10.1007/s11135-015-0206-0 (visited on 09/09/2021).

Do, Chuong B. and Serafim *Batzoglou*. "What is the expectation maximization algorithm?" In: *Nature Biotechnology* 26.8 (Aug. 2008). Bandiera abtest: a Cg type: Nature Research Journals Number: 8 Primary atype: Reviews Publisher: Nature Publishing Group, pp. 897–899. ISSN: 1546-1696. DOI: 10.1038/nbt1406. URL: https://www.nature.com/articles/ nbt1406 (visited on 09/09/2021).

Drennan, Robert D. *Statistics for Archaeologists: A Common Sense Approach*. en. Google-BooksID: wZEmzgEACAAJ. Springer US, Mar. 2010. ISBN: 978-1-4419-6071-9.

Efron, B. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1 (Jan. 1979). Publisher: Institute of Mathematical Statistics, pp. 1–26. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1176344552. URL: https://projecteuclid.org/ journals/annals-of-statistics/volume-7/issue-1/BootstrapMethods-Another-Look-at-the-Jackknife/10.1214/aos/1176344552. full (visited on 09/09/2021).

Efron, B. and R. J. *Tibshirani*. *An Introduction to the Bootstrap*. Google-Books-ID: gLlpIUxRntoC. CRC Press, May 15, 1994. 456 pp. ISBN: 978-0-412-04231-7.

Esnault, Cyril, May-Line *Gadonna*, Maxence *Queyrel*, Alexandre *Templier*, and Jean-daniel *Zucker*. "Q-Finder: An Algorithm for Credible Subgroup Discovery in Clinical Data Analysis - An Application to the International Diabetes Management Practice Study." In: *Frontiers in Artificial Intelligence* 3 (2020), p. 559927. DOI: 10.3389/frai.2020.559927

Ester, Martin, Hans-Peter *Kriegel*, Jorg¨ *Sander*, and Xiaowei *Xu*. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, Aug. 1996, pp. 226–231. (Visited on 12/05/2021).

Estivill-Castro, Vladimir. "Why so many clustering algorithms: a position paper". In: *ACM SIGKDD Explorations Newsletter* 4.1 (June 1, 2002), pp. 65–75. ISSN: 1931-0145. DOI: 10.1145/568574.568575. URL: https://doi.org/10.1145/568574.568575 (visited on 09/09/2021).

Fahad, Adil, Najlaa *Alshatri*, Zahir *Tari*, Abdullah *Alamri*, Ibrahim *Khalil*, Albert Y. *Zomaya*, Sebti *Foufou*, and Abdelaziz *Bouras*. "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis". In: *IEEE Transactions on Emerging Topics in Computing* 2.3 (Sept. 2014). Conference Name: IEEE Transactions on Emerging Topics in Computing, pp. 267–279. ISSN: 2168-6750. DOI: 10.1109/TETC.2014.2330519.

Fahrmeir, Ludwig, Thomas *Kneib*, Stefan *Lang*, and Brian *Marx*. "Categorical Regression Models". In: *Regression: Models, Methods and Applications*. Ed. by Ludwig *Fahrmeir*, Thomas *Kneib*, Stefan *Lang*, and Brian *Marx*. Berlin, Heidelberg: Springer, 2013, pp. 325–347. ISBN: 978-3642-34333-9. DOI: 10.1007/978-3-642-34333-9_6. URL: https://doi.org/10. 1007/978-3-642-34333-9_6 (visited on 09/09/2021).

Fisher, Douglas H. "Knowledge Acquisition Via Incremental Conceptual Clustering". In: *Machine Learning* 2.2 (Sept. 1, 1987), pp. 139–172. ISSN: 1573-0565. DOI: 10.1023/A:1022852608280. URL: https://doi.org/10.1023/A:1022852608280 (visited on 09/09/2021).

Foss, Alexander H. and Marianthi *Markatou*. "kamila: Clustering Mixed-Type Data in R and Hadoop". en. In: *Journal of Statistical Software* 83 (Feb. 2018), pp. 1–44. ISSN: 1548-7660. DOI: 10.18637/jss.v083.i13. URL: https://doi.org/10.18637/jss.v083.i13 (visited on 12/05/2021).

Franti¨ , Pasi, Sami *Sieranoja*, Katja *Wikstrom¨* , and Tiina *Laatikainen*. "Clustering Diagnoses From 58 Million Patient Visits in Finland Between 2015 and 2018". EN. In: *JMIR Medical Informatics* 10.5 (May 2022). Company: JMIR Medical Informatics Distributor: JMIR Medical Informatics Institution: JMIR Medical Informatics Label: JMIR Medical Informatics Publisher: JMIR Publications Inc., Toronto, Canada, e35422. DOI: 10.2196/35422. URL: https://medinform. jmir.org/2022/5/e35422 (visited on 05/12/2022).

Gordon. *Classification - 2nd Edition - A.D. Gordon - Routledge Book*. 1999. URL: https://www. routledge.com/Classification/Gordon/p/book/9780367399665 (visited on 12/15/2021).

Green, Paul E. and Abba M. *Krieger*. "A Comparison of Alternative Approaches to Cluster-Based Market Segmentation". en. In: *Market Research Society. Journal.* 37.3 (May 1995). Publisher: SAGE Publications, pp. 1–19. ISSN: 0025-3618. DOI: 10.1177/147078539503700302. URL: https://doi.org/10.1177/147078539503700302 (visited on 12/16/2021).

Greenacre, Michael. *Correspondence Analysis in Practice*. 2nd ed. New York: Chapman and Hall/CRC, May 2007. ISBN: 978-0-429-14614-5. DOI: 10.1201/9781420011234.

Theory and Applications of Correspondence Analysis. en. Google-Books-ID: LsPaAAAAMAAJ. Academic Press, 1984. ISBN: 978-0-12-299050-2.

Greenacre, Michael and Jorg *Blasius*, eds. *Multiple Correspondence Analysis and Related Methods*. New York: Chapman and Hall/CRC, June 2006. ISBN: 978-0-429-14196-6. DOI: 10.1201/9781420011319.

Greene, D., A. *Tsymbal*, N. *Bolshakova* and P. *Cunningham*, "Ensemble clustering in medical diagnostics," Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems, 2004, pp. 576-581, doi: 10.1109/CBMS.2004.1311777.

Guha, Sudipto, Rajeev *Rastogi*, and Kyuseok *Shim*. "CURE: an efficient clustering algorithm for large databases". In: *ACM SIGMOD Record* 27.2 (June 1998), pp. 73–84. ISSN: 0163-5808. DOI: 10.1145/276305.276312. URL: https://doi.org/10.1145/276305.276312 (visited on 12/05/2021).

Rock: A robust clustering algorithm for categorical attributes - ScienceDirect. URL: https:// www.sciencedirect.com/science/article/abs/pii/S0306437900000223 (visited on 12/15/2000).

Halkidi, Maria, Yannis *Batistakis*, and Michalis *Vazirgiannis*. "On Clustering Validation Techniques". In: *Journal of Intelligent Information Systems* 17.2 (Dec. 1, 2001), pp. 107–145. ISSN: 1573-7675. DOI: 10.1023/A:1012801612483. URL: https://doi.org/10.1023/ A:1012801612483 (visited on 09/09/2021).

Handl, Julia, Joshua *Knowles*, and Douglas B. *Kell*. "Computational cluster validation in postgenomic data analysis". In: *Bioinformatics* 21.15 (Aug. 2005), pp. 3201–3212. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti517. URL: https://doi.org/10.1093/ bioinformatics/bti517 (visited on 12/16/2021).

Hennig, Christian. "Cluster validation by measurement of clustering characteristics relevant to the user". In: *arXiv:1703.09282 [stat]* (Sept. 8, 2020). arXiv: 1703.09282. URL: http:// arxiv.org/abs/1703.09282 (visited on 09/09/2021).

Cluster-wise assessment of cluster stability - ScienceDirect. 2007. URL: https://www. sciencedirect.com/science/article/abs/pii/S0167947306004622?via%3Dihub (visited on 09/09/2021).

"Dissolution point and isolation robustness: Robustness criteria for general cluster analysis meth-ods". In: *Journal of Multivariate Analysis* 99.6 (July 1, 2008), pp. 1154–1176. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2007.07.002. URL: https://www.sciencedirect.com/ science/article/pii/S0047259X07000942 (visited on 09/09/2021).

"How Many Bee Species? A Case Study in Determining the Number of Clusters". In: *Data Analysis, Machine Learning and Knowledge Discovery*. Ed. by Myra *Spiliopoulou*, Lars *SchmidtThieme*, and Ruth *Janning*. Studies in Classification, Data Analysis, and Knowledge Organization. Cham: Springer International Publishing, 2014, pp. 41–49. ISBN: 978-3-319-01595-8. DOI:10.1007/978-3-319-01595-8_5.

Hennig, Christian. *How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification - Hennig - 2013 - Journal of the Royal Statistical Society: Series C (Applied Statistics) - Wiley Online Library*. 2013. URL: https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1467-9876.2012.01066.x (visited on 09/09/2021).

Herawan, Tutut and Ali Seyed *Shirkhorshidi*. *Big Data Clustering: A Review — SpringerLink*. Springer Link. 2014. URL: https://link.springer.com/chapter/10.1007/9783-319-09156-3_49 (visited on 03/16/2021).

Huang, Zhexue. "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining," in: *In Research Issues on Data Mining and Knowledge Discovery*, 1997, pp. 1–8.

"Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Val-ues". en. In: *Data Mining and Knowledge Discovery* 2.3 (Sept. 1998), pp. 283–304. ISSN: 1573-756X. DOI: 10.1023/A:1009769707641. URL: https://doi.org/10.1023/A: 1009769707641 (visited on 12/05/2021).

Hwang, Heungsun, Hec *Montreal´* , William R. *Dillon*, and Yoshio *Takane*. "An Extension of Multiple Correspondence Analysis for Identifying Heterogeneous Subgroups of Respondents". In: *Psychometrika* 71.1 (Mar. 1, 2006), pp. 161–171. ISSN: 1860-0980. DOI: 10.1007/s11336-004-1173-x. URL: https://doi.org/10.1007/s11336-004-1173-x (visited on 09/09/2021).

Hwang, Heungsun and Yoshio *Takane*. *Generalized Constrained Canonical Correlation Analysis: Multivariate Behavioral Research: Vol 37, No 2*. 2010. URL: https://www.tandfonline.com/doi/abs/10.1207/S15327906MBR3702_01 (visited on 12/16/2021).

Jain, Anil. "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters*. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR) 31.8 (June 1, 2010), pp. 651–666. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2009.09.011. URL: https://www.sciencedirect.com/science/article/pii/S0167865509002323 (visited on 03/16/2021).

Jin, Xin and Jiawei *Han*. "K-Medoids Clustering". In: *Encyclopedia of Machine Learning*. Ed. by Claude *Sammut* and Geoffrey I. *Webb*. Boston, MA: Springer US, 2010, pp. 564–565. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_426. URL: https://doi.org/ 10.1007/978-0-387-30164-8_426 (visited on 09/09/2021).

Josse, Julie, Marie *Chavent*, Benot *Liquet*, and Franc¸ois *Husson*. "Handling Missing Values with Regularized Iterative Multiple Correspondence Analysis". en. In: *Journal of Classification* 29.1 (Apr. 2012), pp. 91–116. ISSN: 1432-1343. DOI: 10.1007/s00357-012-9097-0. URL: https://doi.org/10.1007/s00357-012-9097-0 (visited on 12/16/2021).

Josse, Julie, Jer´ omeˆ *Pages`* , and Franc¸ois *Husson*. "Multiple imputation in principal component analysis". In: *Advances in Data Analysis and Classification* 5.3 (Oct. 1, 2011), pp. 231–246. ISSN: 1862-5355. DOI: 10.1007/s11634-011-0086-7. URL: https://doi.org/10. 1007/s11634-011-0086-7 (visited on 09/09/2021).

Kamoshida, Ryota and Fuyuki *Ishikawa*. "Automated Clustering and Knowledge Acquisition Support for Beginners". In: *Procedia Computer Science*. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020 176 (Jan. 1, 2020), pp. 1596–1605. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.09.182. URL: https://www.sciencedirect.com/science/article/pii/S1877050920320846 (visited on 09/09/2021).

Kaufman, Leonard and Peter J. *Rousseeuw*. *Finding Groups in Data: An Introduction to Cluster Analysis*. en. Google-Books-ID: YeFQHiikNo0C. John Wiley & Sons, Sept. 2009. ISBN: 978-0470-31748-8.

"Partitioning Around Medoids (Program PAM)". en. In: *Finding Groups in Data*. Section: 2 eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470316801.ch2. John Wiley & Sons, Ltd, 1990, pp. 68–125. ISBN: 978-0-470-31680-1. DOI: 10.1002/9780470316801. ch2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2 (visited on 12/05/2021).

Kaushik, Manju and Bhawana *Mathur*. "Comparative Study of K-Means and Hierarchical Clustering Techniques". In: *Internatioonal journal of Software and Hardware Research in Engineering* 2 (June 1, 2014), pp. 93–98. *Keim*, Daniel A. and Alexander *Hinneburg*. *An efficient approach to clustering in large multimedia databases with noise — Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. 1998. URL: https://dl.acm.org/doi/10.5555/3000292. 3000302 (visited on 12/15/2021).

Kiselev, Vladimir Yu, Tallulah S. *Andrews*, and Martin *Hemberg*. "Challenges in unsupervised clustering of single-cell RNA-seq data". In: *Nature Reviews Genetics* 20.5 (May 2019). Number: 5 Publisher: Nature Publishing Group, pp. 273–282. ISSN: 1471-0064. DOI: 10.1038/s41576-018-0088-9. URL: https://www.nature.com/articles/s41576-018-0088-9 (visited on 03/16/2021).

Kleinberg, Jon M. "An Impossibility Theorem for Clustering." In: *Advances in Neural Information Processing Systems* (2002).

Lange, Tilman, Volker *Roth*, Mikio L. *Braun*, and Joachim *Buhmannand*. "Stability-Based Validation of Clustering Solutions". In: *Neural Computation* 16.6 (June 1, 2004), pp. 1299–1323. ISSN: 0899-7667. DOI: 10.1162/089976604773717621. URL: https://doi.org/10. 1162/089976604773717621 (visited on 03/16/2021).

Le Roux, Brigitte and Henry *Rouanet*. *Multiple Correspondence Analysis*. Google-Books-ID: GWsHakQGEHsC. SAGE, 2010. 129 pp. ISBN: 978-1-4129-6897-3.

Lee, Young, *Lee,* Eun and Byeong, *Park*. (2012). Principal component analysis in very high-dimensional spaces. Statistica Sinica. 22. 10.5705/ss.2010.149.

Li, Nan and Longin Jan *Latecki*. "Affinity Learning for Mixed Data Clustering." In: *IJCAI* (2017).

Lorenzo-Seva, Urbano. "Horn's Parallel Analysis for Selecting the Number of Dimensions in Correspondence Analysis". In: *Methodology* 7.3 (Jan. 2011). Publisher: Hogrefe Publishing, pp. 96–102. ISSN: 1614-1881. DOI: 10.1027/1614-2241/a000027. URL: https: //econtent.hogrefe.com/doi/10.1027/1614-2241/a000027 (visited on 12/16/2021).

Maaten, L.d and G. *Hinton. "*Visualizing data using t-SNE". In: *J. Mach. Learn. Res., 9* (2008), pp. 2579-2605

MacQueen, J. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* 5.1 (Jan. 1967). Publisher: University of California Press, pp. 281–298. URL: https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematicalstatistics-and-probability/Proceedings-of-the-Fifth-BerkeleySymposium-on-Mathematical-Statistics-and/chapter/Some-methodsfor-classification-and-analysis-of-multivariate-observations/ bsmsp/1200512992 (visited on 12/05/2021).

McCallum, Andrew, Kamal *Nigam*, and Lyle H. *Ungar*. "Efficient clustering of high-dimensional data sets with application to reference matching". In: *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. Boston, Massachusetts, United States: ACM Press, 2000, pp. 169–178. ISBN: 1-58113-233-6. DOI: http://doi.acm.org/10.1145/347090.347123.

McCane, Brendan and Michael *Albert*. "Distance functions for categorical and mixed variables". en. In: *Pattern Recognition Letters* 29.7 (May 2008), pp. 986–993. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.01.021. URL: https://www.sciencedirect.com/ science/article/pii/S0167865508000524 (visited on 12/05/2021).

McInnes, L., J. *Healy* and *J. Melville.* "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv,* 1802 (2018), p. 03426

Meila˘, Marina. "Comparing clusterings—an information based distance". In: *Journal of Multivariate Analysis* 98.5 (May 1, 2007), pp. 873–895. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2006.11.013. URL: https://www.sciencedirect.com/science/article/ pii/S0047259X06002016 (visited on 09/09/2021).

Milligan, Glenn W. and Martha C. *Cooper*. "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2 (June 1, 1985), pp. 159–179. ISSN: 1860-0980. DOI: 10.1007/BF02294245. URL: https://doi.org/10.1007/BF02294245 (visited on 09/09/2021).

Mitsuhiro, Masaki and Hiroshi *Yadohisa*. "Reduced k-means clustering with MCA in a lowdimensional space". In: *Computational Statistics* 30.2 (June 1, 2015), pp. 463–475. ISSN: 1613-9658. DOI: 10.1007/s00180-014-0544-8. URL: https://doi.org/10.1007/ s00180-014-0544-8 (visited on 09/09/2021).

Mittal, Mamta, Lalit M *Goyal*, Duraisamy Jude *Hemanth*, and Jasleen K *Sethi*. "Clustering approaches for high-dimensional databases: A review". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.3 (May 2019), e1300.

Murtagh, Fionn. *Correspondence Analysis and Data Coding with Java and R*. en. Google-Books-ID: wIK7p0 _NyzoC. CRC Press, May 2005. ISBN: 978-1-4200-3494-3.

Nagpal, Arpita, Arnan *Jatain*, and Deepti *Gaur*. "Review based on data clustering algorithms". In: *2013 IEEE Conference on Information Communication Technologies*. 2013 IEEE Conference on Information Communication Technologies. Apr. 2013, pp. 298–303. DOI: 10.1109/CICT.2013.6558109.

Ng, Raymond and Jiawei *Han*. "CLARANS: a method for clustering objects for spatial data mining". In: *IEEE Transactions on Knowledge and Data Engineering* 14.5 (Sept. 2002). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 1003–1016. ISSN: 1558-2191. DOI: 10.1109/TKDE.2002.1033770.

"Efficient and Effective Clustering Methods for Spatial Data Mining". In: *VLDB*. 1994.

Nietto, Paulo Rogerio and Maria do *Carmo Nicoletti*. "Estimating the Number of Clusters as a Preprocessing Step to Unsupervised Learning". In: *Intelligent Systems Design and Applications*. Ed. by Ana Maria *Madureira*, Ajith *Abraham*, Dorabela *Gamboa*, and Paulo *Novais*. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2017, pp. 25–34. ISBN: 978-3-319-53480-0. DOI: 10.1007/978-3-319-53480-0_3.

Nishisato, Shizuhiko. *Analysis of Categorical Data: Dual Scaling and its Applications*. en. Publication Title: Analysis of Categorical Data. University of Toronto Press, Nov. 2019. DOI: 10.3138/9781487577995. URL: https://www.degruyter.com/document/doi/10.3138/ 9781487577995/html (visited on 12/05/2021).

Obembe, Olawole and Jelili *Oyelade*. *Data Clustering: Algorithms and Its Applications*. IEEE Xplore. 2019. URL: https://ieeexplore.ieee.org/document/8853526 (visited on 03/16/2021).

Ortega, Francisco B., Carl J. *Lavie*, and Steven N. *Blair*. "Obesity and Cardiovascular Disease". In: *Circulation Research* 118.11 (May 27, 2016). Publisher: American Heart Association, pp. 1752–1770. DOI: 10.1161/CIRCRESAHA.115.306883. URL: https://www.ahajournals.org/doi/full/10.1161/circresaha.115.306883 (visited on 03/23/2021).

Oyelade, J et al. "Data Clustering: Algorithms and Its Applications," 2019 19th International Conference on Computational Science and Its Applications (ICCSA), 2019, pp. 71-81, doi: 10.1109/ICCSA.2019.000-1.

Pagès, Jerome. *Analyse factorielle de donnees mixtes´* . 2004. URL: http://www.numdam.org/article/RSA_2004__52_4_93_0.pdf (visited on 03/19/2021).

Pagès, Jerome and Francois *Husson*. *Exploratory Multivariate Analysis by Example Using R 2nd Edition - F*. 2017. URL: https://www.routledge.com/ExploratoryMultivariate-Analysis-by-Example-Using-R/Husson-Le-Pages/p/ book/9780367658021 (visited on 12/16/2021).

Pasi Franti¨ , Mohammad Rezaei. *Can the Number of Clusters Be Determined by External Indices?* 2020. URL: https://ieeexplore.ieee.org/document/9090190 (visited on 05/12/2022).

Rezaei, Mohammad and Pasi *Franti¨* . "Set Matching Measures for External Cluster Validity". In: *IEEE Transactions on Knowledge and Data Engineering* 28.8 (Aug. 2016). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 2173–2186. ISSN: 1558-2191. DOI: 10.1109/TKDE.2016.2551240.

Rousseeuw, Peter J. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". en. In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. ISSN: 0377-0427. DOI: 10.1016/0377-0427(87)90125-7. URL: https://www. sciencedirect.com/science/article/pii/0377042787901257 (visited on 12/16/2021).

Saint Pierre, Aude, Joanna *Giemza*, Isabel *Alves*, Matilde *Karakachoff*, Marinna *Gaudin*, Philippe *Amouyel*, Jean-Franc¸ois *Dartigues*, Christophe *Tzourio*, Martial *Monteil*, Pilar *Galan*, Serge *Hercberg*, Iain *Mathieson*, Richard *Redon*, Emmanuelle *Genin´* , and Christian *Dina*. "The genetic history of France". In: *European Journal of Human Genetics* 28.7 (July 2020). Bandiera abtest: a Cg type: Nature Research Journals Number: 7 Primary atype: Research Publisher: Nature Publishing Group Subject term: Genetics;Population genetics Subject term id: genetics;populationgenetics, pp. 853–865. ISSN: 1476-5438. DOI: 10.1038/s41431-020-0584-1. URL: https://www.nature.com/articles/s41431-020-0584-1 (visited on 09/09/2021).

Saisubramanian, Sandhya, Sainyam *Galhotra*, and Shlomo *Zilberstein*. "Balancing the Tradeoff Between Clustering Value and Interpretability". In: *arXiv:1912.07820 [cs, stat]* (Jan. 30, 2020). arXiv: 1912.07820. URL: http://arxiv.org/abs/1912.07820 (visited on 03/17/2021).

Sculley, D. "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. WWW '10. New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 1177–1178. ISBN: 978-1-60558-799-8. DOI: 10.1145/1772690.1772862. URL: https://doi.org/10.1145/1772690.1772862 (visited on 12/05/2021).

Sheikholeslami, Gholamhosein, Surojit *Chatterjee*, and Aidong *Zhang*. *WaveCluster: A MultiResolution Clustering Approach for Very Large Spatial Databases — Semantic Scholar*. 1998. URL: https://www.semanticscholar.org/paper/WaveCluster%5C%3A-A-Multi-Resolution-Clustering-Approach-Sheikholeslami-Chatterjee/ f0015f0e834a84699a9b83c6c9af33acdac05069 (visited on 12/16/2021).

Sieranoja, Sami and Pasi *Franti¨* . "Adapting k-means for graph clustering". en. In: *Knowledge and Information Systems* 64.1 (Jan. 2022), pp. 115–142. ISSN: 0219-3116. DOI: 10.1007/s10115-021-01623-y. URL: https://doi.org/10.1007/s10115-021-01623-y (visited on 05/12/2022).

"Fast and general density peaks clustering". en. In: *Pattern Recognition Letters* 128 (Dec. 2019), pp. 551–558. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2019.10.019. URL: https://www.sciencedirect.com/science/article/pii/S0167865519303009 (visited on 05/12/2022).

Testa, Damien and Laurent *Chiche*. *Unsupervised clustering analysis of data from an online community to identify lupus patient profiles with regards to treatment preferences - Damien Testa, Noemie´ Jourde-Chiche, Julien Mancini, Pasquale Varriale, Lise Radoszycki, Laurent Chiche, 2021*. URL: https://journals.sagepub.com/doi/10.1177/09612033211033977 (visited on 11/15/2021).

Tibshirani, Robert, Guenther *Walther*, and Trevor *Hastie*. "Estimating the number of clusters in a data set via the gap statistic". en. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2 (2001). eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/14679868.00293, pp. 411–423. ISSN: 1467-9868. DOI: 10.1111/1467-9868.00293. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293 (visited on 11/15/2021).

Torgerson W.S. *"Multidimensional scaling: I. Theory and method Psychometrika"*, 17 (1952), pp. 401-419

Van de Velden, Michel, A. *Iodice D' Enza*, and F. *Palumbo*. "Cluster Correspondence Analysis". In: (Oct. 1, 2014). Number: EI 2014-24. URL: https://repub.eur.nl/pub/77010 (visited on 09/09/2021).

Vellido, Alfredo. "The importance of interpretability and visualization in machine learning for applications in medicine and health care". In: *Neural Computing and Applications* 32.24 (Dec. 1, 2020), pp. 18069–18083. ISSN: 1433-3058. DOI: 10.1007/s00521-019-04051-w. URL: https://doi.org/10.1007/s00521-019-04051-w (visited on 03/17/2021).

Wang, Wei, Jiong *Yang*, and Richard *Muntz*. *STING : A Statistical Information Grid Approach to Spatial Data Mining*. 1997.

Warwick, K. M. and L. *Lebart*. "Multivariate descriptive statistical analysis (correspondence analysis and related techniques for large matrices)". en. In: *Applied Stochastic Models and Data Analysis* 5.2 (1989). eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asm.3150050207, pp. 175–175. ISSN: 1099-0747. DOI: 10.1002/asm.3150050207. URL: https:// onlinelibrary.wiley.com/doi/abs/10.1002/asm.3150050207 (visited on 12/05/2021).

Windgassen, Sula, Rona *Moss-Morris*, Kimberley *Goldsmith*, and Trudie *Chalder*. "The importance of cluster analysis for enhancing clinical practice: an example from irritable bowel syndrome". In: *Journal of Mental Health* 27.2 (Mar. 4, 2018). Publisher: Routledge eprint: https://doi.org/10.1080/09638237.2018.1437615, pp. 94–96. ISSN: 0963-8237. DOI: 10.1080/09638237.2018.1437615. URL: https://doi.org/10.1080/09638237.2018.1437615 (visited on 03/16/2021).

Wiwie, Christian, Jan *Baumbach*, and Richard *Rottger¨* . "Comparing the performance of biomedical clustering methods". In: *Nature Methods* 12.11 (Nov. 2015). Bandiera abtest: a Cg type: Nature Research Journals Number: 11 Primary atype: Research Publisher: Nature Publishing Group Subject term: Computational biology and bioinformatics;Databases Subject term id: computationalbiology-and-bioinformatics;databases, pp. 1033–1038. ISSN: 1548-7105. DOI: 10.1038/ nmeth.3583. URL: https://www.nature.com/articles/nmeth.3583 (visited on 09/09/2021).

Xu, Rui and Donald C. *Wunsch*. "Clustering Algorithms in Biomedical Research: A Review". In: *IEEE Reviews in Biomedical Engineering* 3 (2010). Conference Name: IEEE Reviews in Biomedical Engineering, pp. 120–154. ISSN: 1941-1189. DOI: 10.1109/RBME.2010.2083647.

Yang, Jiawei, Susanto *Rahardja*, and Pasi *Franti¨* . "Mean-shift outlier detection and filtering". en. In: *Pattern Recognition* 115 (July 2021), p. 107874. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2021.107874. URL: https://www.sciencedirect.com/science/article/ pii/S0031320321000613 (visited on 05/12/2022).

Zhang, Tian, Raghu *Ramakrishnan*, and Miron *Livny*. "BIRCH: an efficient data clustering method for very large databases". In: *ACM SIGMOD Record* 25.2 (June 1996), pp. 103–114. ISSN: 01635808. DOI: 10.1145/235968.233324. URL: https://doi.org/10.1145/235968. 233324 (visited on 12/05/2021).

Zhao, Ying and George *Karypis*. *Comparison of Agglomerative and Partitional Document Clustering Algorithms*. Section: Technical Reports. MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE, Apr. 17, 2002. URL: https://apps.dtic.mil/sti/ citations/ADA439503 (visited on 09/09/2021).

Zhou, Fang L., Hirotaka *Watada*, Yuki *Tajima*, Mathilde *Berthelot*, Dian *Kang*, Cyril *Esnault*, Yujin *Shuto*, Hiroshi *Maegawa*, and Daisuke *Koya*. "Identification of subgroups of patients with type 2 diabetes with differences in renal function preservation, comparing patients receiving sodium-glucose co-transporter-2 inhibitors with those receiving dipeptidyl peptidase-4 inhibitors, using a supervised machine-learning algorithm (PROFILE study): A retrospective analysis of a Japanese commercial medical database". en. In: *Diabetes, Obesity and Metabolism* 21.8 (2019). eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/dom.13753, pp. 1925–1934. ISSN: 1463-1326. DOI: 10.1111/dom.13753. URL: https://onlinelibrary.wiley.com/ doi/abs/10.1111/dom.13753 (visited on 11/15/2021).

Zwick, William R. and Wayne F. *Velicer*. "Comparison of five rules for determining the number of components to retain". In: *Psychological Bulletin* 99.3 (1986). Place: US Publisher: American Psychological Association, pp. 432–442. ISSN: 1939-1455. DOI: 10.1037/0033-2909.99.3.432.