LIVES Working Paper 2024/104

Sequence Analysis for Large Databases

MATTHIAS STUDER, ROJIN SADEGHI, LOUIS TOCHON

RESEARCH PAPER
http://dx.doi.org/10.12682/lives.2296-1658.2024.104
ISSN 2296-1658

UNIL | Université de Lausanne
UNIVERSITÉ DE GENÈVE
LIVES
SWISS CENTRE OF EXPERTISE IN LIFE COURSE RESEARCH

Abstract

This article develops and reviews methods for the creation of sequence analysis typologies in large databases. The creation of sequence analysis typologies relies on the computation of distances between all observations, which quickly becomes intractable with large databases, even with modern computers. We start by discussing the CLARA algorithm before extending it with methods recently proposed for sequence analysis. The strengths of the approaches are assessed using simulations, which further allows drawing practical guidelines. Next, we discuss three approaches to measure the quality of the clustering without computing all distances. The first is based on representative sequences (i.e., medoids) while the second is based on bootstrapping. We then introduce a third innovative approach based on clustering stability, which further allows assessing the convergence of the clustering algorithm. The methods are illustrated through a study of family trajectories in India with more than 180,000 cases. All the methods are made available in the WeightedCluster R package.

Authors
Matthias Studer, Rojin Sadeghi, Louis Tochon

Authors' Affiliation
(LIVES Center and Institute of Demography and Socioeconomics,
Geneva School of Social Sciences, University of Geneva, Switzerland

Keywords
> Sequence Analysis
> Typologies
> Large Databases
> Clustering Algorithms
> Family Trajectories

Sequence Analysis for Large Databases

Matthias Studer*, Rojin Sadeghi, Louis Tochon

LIVES Center and Institute of Demography and Socioeconomics, Geneva School of Social Sciences, University of Geneva, Switzerland

Abstract

This article develops and reviews methods for the creation of sequence analysis typologies in large databases. The creation of sequence analysis typologies relies on the computation of distances between all observations, which quickly becomes intractable with large databases, even with modern computers. We start by discussing the CLARA algorithm before extending it with methods recently proposed for sequence analysis. The strengths of the approaches are assessed using simulations, which further allows drawing practical guidelines. Next, we discuss three approaches to measure the quality of the clustering without computing all distances. The first is based on representative sequences (i.e., medoids) while the second is based on bootstrapping. We then introduce a third innovative approach based on clustering stability, which further allows assessing the convergence of the clustering algorithm. The methods are illustrated through a study of family trajectories in India with more than 180,000 cases. All the methods are made available in the `WeightedCluster R` package.

*Electronic address: `matthias.studer@unige.ch`

1 Introduction

Since its introduction in the social sciences by Abbott and Forrest (1986), Sequence Analysis (SA) has been increasingly used to study trajectories. It is considered a key method for life-course research (e.g., Shanahan 2000; Mayer 2009; Buchmann and Kriesi 2011; Piccarreta and Studer 2019; Liefbroer 2019; Liao and Fasang 2020; Liao et al. 2022). The standard use of SA revolves around the creation of a typology of trajectories with cluster analysis (Gauthier et al. 2014). This typology allows the identification of recurrent patterns or, in other words, typical successions of states through which the trajectories run. By doing so, it provides a holistic perspective on the trajectories. However, despite its growing popularity in life-course research, the standard SA procedure struggles to handle large databases (i.e., more than 40,000 cases at the time of writing the article), as the memory and computational requirements become overwhelming. This article provides a framework for dealing with these limitations.

Large longitudinal databases are becoming increasingly common, coming from large sample surveys, comparative international longitudinal surveys or administrative data (e.g. Losa et al. 2014; Studer et al. 2015; von Gunten et al. 2019). They hold great potential for life-course research for several reasons. First, and most obviously, a large number of observations increase the power of any statistical analyses.

Second, some of these databases allow focusing on specific, often infrequent, subgroups of interest and compare them to the general population. For instance, Landoes (2022) used a large administrative database to study educational paths of asylum seekers in Switzerland, highlighting the detrimental impact of temporary residential permits.

Third, several of these large databases come from international surveys allowing adopting a comparative perspective. Such an analytical framework requires a sufficient number of cases in each country to identify country-specific patterns, generally resulting in a large overall sample. For instance, several studies have relied

on the Survey of Health Ageing and Retirement in Europe (SHARE) (Börsch-Supan et al. 2013) to compare the life courses in 28 European countries.

Finally, a large database offers a unique opportunity to identify atypical trajectories. These trajectories might reveal particularly unusual, vulnerable or at-risk patterns, and are not easily identified in standard smaller surveys as they tend to be infrequent. For instance, in school-to-work transition studies, the identification of individuals following atypical, often "at-risk," trajectories is of central interest. At the same time, these atypical patterns are generally uncommon, and might therefore not be observed in smaller surveys.

To the best of our knowledge, two strategies have been employed so far to use SA with large databases. First, some studies rely on a random subsample of the data (see, for instance, Lorentzen et al. 2018). However, by using a subsample, most of the large-database advantages listed above are lost. Second, Pesando et al. (2021) proposed to overcome this limitation by "extending" the typology identified in a subsample to the overall sample. While the strategy is appealing, it also has its limitations. Indeed, the resulting typology might depend too strongly on the subsample. Moreover, it might fail to identify atypical or context-specific trajectories, which might not be sufficiently represented in the subsample.

The goal of this article is to propose a framework to create SA typologies in large databases that overcomes these limitations. We do so by introducing the Clustering Large Applications (CLARA) algorithm (Kaufman and Rousseeuw 1990; Ng and Han 2002; Schubert and Rousseeuw 2019) to SA, before extending it to handle the three clustering approaches relevant for SA identified by Helske et al. (2023). Each of these approaches have their own strengths and weaknesses, making them particularly relevant for specific applications. First, crisp (also named hard) clustering classifies each observation (or sequence) in exactly one type of trajectories. This is by far the most used set of methods in SA. While the results might be easier to interpret, it might fail to describe trajectories in-between types and might oversimplify the information when the clustering structure is weak.

Second, in fuzzy (also named soft) clustering, each observation is assigned with an estimated membership strength to each cluster (see Studer 2018, for a presentation of the methods with SA). By doing so, it can efficiently describe in-between types trajectories. Third, the "representativness" method further aims to describe outliers or unclassifiable sequences, and is shown to be more efficient when these trajectories need to be identified (Helske et al. 2023).

In the second part of the article, we assess the performances of the proposed clustering algorithms using simulations, which further allows us to provide recommendations on their parameter values. We then discuss the computational burden of the distance measures used in SA, as it also becomes an issue when working with a large database, and draw recommendations regarding the choice of the distance measure. Finally, we introduce and develop several methods to assess the quality of a clustering in large databases, which is a key step to choose the number of groups.

2 Sequence Clustering in Large Databases

The creation of typologies in SA generally follows three steps (e.g. Studer 2013). First, the trajectories are coded as sequences of states. Second, the trajectories are compared to one another using a dissimilarity measure (see Studer and Ritschard 2016, for a review). Third, the dissimilarities between each pair of sequences are used to create a typology of the trajectories with cluster analysis and compute its statistical quality (Studer 2013).

This standard procedure raises computational issues when the database has a large number of cases. Indeed, it requires to store in memory and to compute the dissimilarities between all pairs of sequences, i.e., $N(N-1)/2$ values with N the number of cases. This rapidly becomes intractable as sample size increases, even with modern computers, for two reasons. First, the computation of the dissimilarities themselves might be overwhelming. We further discuss this issue in section 4. Second, most cluster algorithms require to store all the dissimilarities in

memory, which is also rapidly intractable. [1] These issues concern the computation of the typology (i.e., cluster analysis), and of the cluster quality indices, which are used to choose the number of groups, or assess the statistical quality of a typology.

2.1 CLARA: Clustering LaRge Applications

The CLARA algorithm was proposed by Kaufman and Rousseeuw (1990) and features among the first clustering algorithms designed for large databases.[2] The general idea of the algorithm is to cluster a random subsample of the data before "extending" this clustering to the whole dataset. By doing so, it limits the overall computation burden and avoids storing the whole distance matrix. Contrary to the procedure of Pesando et al. (2021), the dependence on the random subsample is reduced by repeating the whole procedure several times and keeping only the best solution.

More formally, CLARA works as follows. Let $d(i, j)$ be a distance measure between sequences i and j, N the size of the full dataset, n the size of the subsample, K the number of groups, and I the number of iterations.

1. A subsample of n sequences is randomly drawn.
2. The distance matrix D_s of size $n(n-1)/2$ between the sampled sequences is computed.
3. The sequences of the subsample are clustered using Partioning Around Medoids (PAM) (Kaufman and Rousseeuw 1990) in a predefined number of groups K.
4. The medoids m_k of each cluster k are identified.
5. The distances between each sequence in the whole dataset and the medoids m_k are computed, resulting in a $N \times K$ matrix.
6. Each sequence is assigned to its closest medoid.

7. The quality of the resulting clustering is computed.

To avoid being trapped in a local maximum, the whole procedure (steps one to seven) is repeated I times independently. These repetitions are called "iterations." The best solution according to the clustering quality used in the last step is kept.

The original formulation of CLARA computes the clustering quality as the sum of distance SD as in PAM (Kaufman and Rousseeuw 1990). The criterion is minimized to favor a clustering solution with high within-cluster homogeneity. It is computed as follows. Let m_k be the medoid of the k^{th} cluster C_k:

$$SD = \sum_{i=k}^{k} \sum_{i \in C_k} d(i, m_k) \tag{1}$$

We further propose two improvements of the CLARA algorithm. First, we aggregate identical sequences, i.e., trajectories following the exact same path. This often occurs when studying trajectories that are constrained by social norms or institutions. When aggregating, we keep a single row for sequences occurring multiple times, and weight it accordingly. We then use the weighted version of the algorithms to cluster the data (Studer 2013). This aggregation is done on the whole dataset, and on the selected subsample. It can drastically reduce the computation and memory burdens. Second, we initialize the PAM algorithm with the solution found by the Ward hierarchical clustering algorithm, which, again, speeds up the computation and might improve the quality of the resulting clustering. We use the optimization proposed by Müllner (2013).

The original CLARA formulation allows clustering large databases using a crisp clustering approach, where each sequence is assigned to a single given type. We now turn to its adaptation for clustering with fuzzy clustering and "representativeness."

2.2 Fuzzy CLARA

Fuzzy clustering aims to take into account that some clusters might overlap, with some sequences lying in-between types. Methodologically, the fuzzy approach

therefore allows a better description when the data is not strongly structured into types, as in many social sciences applications. The approach is also relevant from a life-course perspective, as we can think that some trajectories are the product of multiple influences, resulting in sequences that are a mixture of different types (Studer 2018).

Technically, fuzzy clustering aims to estimate a membership strength or degree u_{ik} of sequence i to each cluster k instead of a categorical variable (D'Urso 2016). FANNY (Kaufman and Rousseeuw 1990) and its extension "Fuzzy Relational Clustering" (Dave and Sen 2002) feature among the most relevant algorithms for SA as they can be used with any distance measures. However, they generally require the computation of the whole distance matrix.

The CLARA algorithm can be extended to fuzzy clustering by replacing the PAM algorithm by its fuzzy counterpart fuzzy c-medoids (FCMdd). As PAM, FCMdd is based on the identification of a representative observation for each type (Krishnapuram et al. 1999). A similar strategy can then be used to extend the clustering from the subsample to the whole dataset based on the identified representatives. However, instead of assigning a sequence to a single type as in PAM, the membership strength u_{ik} of the sequence i to cluster j is computed as follows:

$$u_{ik} = \frac{\left(\frac{1}{d(i,m_k)}\right)^{1/(m-1)}}{\sum_{c=1}^{K}\left(\frac{1}{d(i,m_c)}\right)^{1/(m-1)}} \tag{2}$$

Where $d(i, m_k)$ is the distance between sequence i and representative sequence of cluster k, K the number of groups, and m the *fuzzifier*. The m parameter controls the fuzziness of the resulting clustering. Setting m close to 1 results in an almost crisp clustering, while larger values increase the fuzziness of the partition. According to the simulations reviewed by D'Urso (2016), a value between 1.5 and 2.5 provides the best results. He further recommends using $m = 2$ (as in FANNY) and lowering the value if the algorithm fails to converge. A similar strategy can be

used here.

The evaluation of the clustering quality of an iteration of CLARA should also be adapted by using the objective function of the FCMdd algorithm. This objective function reads as follows and can be interpreted as a fuzzy sum of distance to the medoids (Krishnapuram et al. 1999):

$$SD_f = \sum_{i=1}^{n} \sum_{k}^{K} u_{ik}^m d(i, m_k) \tag{3}$$

To improve the estimation, the FCMdd algorithm, as implemented by De Cáceres et al. (2010), is initialized using a combination of FANNY and Ward clustering.

2.3 Representativeness

Helske et al. (2023) propose to use the *representativness* of each cluster instead of a single categorical variable to account for the clustering. This approach allows taking into account how well each sequence is described by the clustering. By doing so, atypical sequences, far from all sequences types can also be identified. The representativeness R_{ik} of sequence i to cluster k is computed using the following formula:

$$R_{ik} = 1 - \frac{d(i, m_k)}{d_{max}} \tag{4}$$

With d_{max} a constant set to the highest observed distance. Setting this constant requires to compute all the distances. However, it can be avoided as it can be set theoretically to the maximum possible distance. Furthermore, as a constant, its exact value has a low impact on most subsequent statistical analyses relying on it.

The implementation is here straightforward, as the *representativness* is computed after the clustering. Furthermore, it only makes use of the distance to the medoids, which are already computed within the CLARA algorithm.

2.4 Conclusion

In this section, we presented the CLARA algorithm and two extensions allowing using the "representativeness" and fuzzy clustering approaches. This algorithm works by clustering subsamples before extending the typology to the whole dataset. To avoid sample dependence of the results, the operation is repeated several times. It therefore has two parameters on top of the usual ones, such as the number of groups or the fuzziness m parameter, that need to be set by the user: n the size of the random subsample, and I the number of iterations.

3 Simulations

To evaluate the performance of CLARA for SA, we conducted an extensive set of simulations. These simulations allow us to compare the results of CLARA with PAM, a standard cluster algorithm used in SA. Aside from evaluating the quality of the results, these simulations further aims to draw guidelines on the parameters values. We start by presenting the results for the crisp clustering, before turning to the fuzzy approach.

3.1 Family Trajectory in India

The simulations are based on a dataset of women's family trajectories in India using the fourth wave of the National Family Health Survey, which is part of the Demographic and Health Survey. The trajectories are then coded using the following states between 12 and 35 years old: no event, sex out of union, formal marriage, married without children, married with one, two, three or more than four children. Our final sample consists of 188,144 women with fully observed trajectories. The trajectories are compared using optimal matching of spells with constant substitution costs, indel cost set to half this value and an expansion cost of 0.25. This distance measure is sensitive to the time spent in each state and the ordering of the states (Studer and Ritschard 2016).

3.2 Crisp Clustering

The aim of the simulations is to identify the minimal number of iterations and subsample size to obtain a clustering with a similar or better quality than PAM. They work as follows. Twenty thousand sequences are randomly drawn from our illustrative data set on family trajectories. These sequences are clustered using PAM, which serves as a benchmark. We then cluster the same data using CLARA, and compare the quality of the two clusterings using the average distance to the medoids $\frac{SD}{N}$, with SD defined according to equation (1).[3] A negative value indicates that CLARA outperforms PAM. Since the "representativness" approach ultimately relies on crisp clustering, the results of these simulations also evaluate this approach.

The simulations were run using a sample size n of 100, 500, 1,000, 5,000 or 10,000, and a number of groups varying between 2 and 20. The whole operation was repeated two hundred times to ensure the stability of the results.

Figure 1 compares the quality of the clustering in eight groups (y-axis) when using PAM and CLARA with different numbers of iterations (x-axis) and subsample size (panel). The figure allows drawing several interesting conclusions. First, a single iteration is risky. Indeed, we observe a high dispersion of the performances with a single iteration. Furthermore, the quality of this single iteration might be worse than PAM. Second, the performances increase with the number of iterations, but only slightly after 200.

Third, the most important differences in performances are linked to sample sizes. Generally speaking, the larger the better. For eight groups, a sample size of at least 1,000 observations is enough. This is well over the $40 + 2 \cdot k$ recommendation of Kaufman and Rousseeuw (1990). However, a larger sample size is required for a higher number of groups. Finally, and importantly, the CLARA algorithm regularly outperforms PAM. This can be explained by the fact that CLARA searches over a larger solution space than PAM if the number of iterations is high enough.

Figure 2 presents the performance of the algorithms for a varying number of

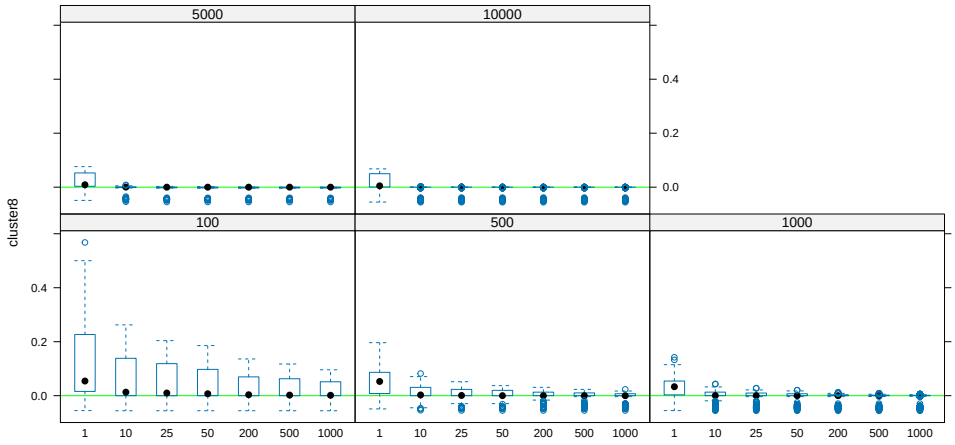


Figure 1: Differences between the quality of CLARA and PAM (y-axis) for varying numbers of iterations (x-axis) and subsample size (panels) for a clustering in 8 groups.

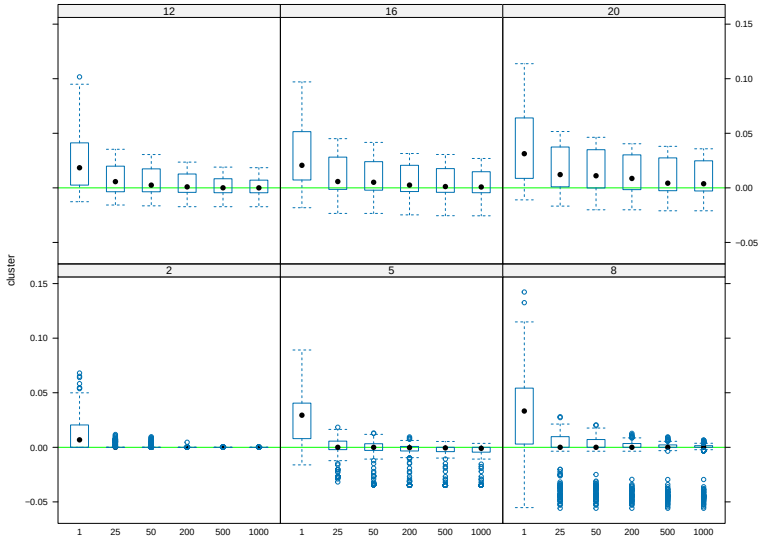


Figure 2: Differences between the quality of CLARA and PAM (y-axis) for varying numbers of iterations (x-axis) and numbers of groups (panels) when the sample size is 1,000.

groups when using a subsample size of 1,000. Figure 7 in appendix A provides the same figure for subsamples of size 10,000. Both figures highlight that a higher

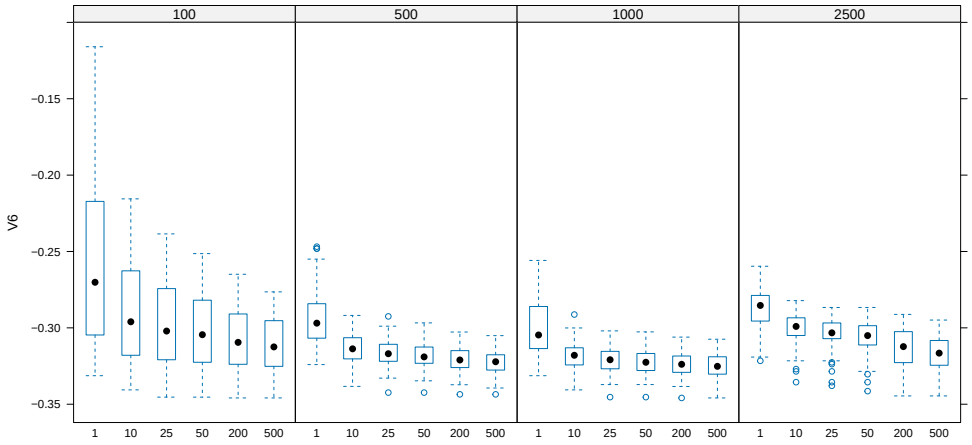


Figure 3: Difference between the quality of fuzzy CLARA and FCMdd (y-axis) for a varying number of iterations (x-axis) and subsample sizes (panels) for a clustering in 8 groups.

number of groups requires more iterations. While 200 iterations were enough for 8 groups, 500 iterations might improve the results for more than 10 groups. One thousand iterations only lead to a tiny improvement with more than 16 groups in some of the simulations.

3.3 Fuzzy Clustering

We used the same approach to evaluate the quality of the clustering results. However, for computational reasons, we used a maximum of 600 iterations, between 2 and 15 groups, and a sample size of 100, 500, 1,000 or 2,500 sequences. As in Helske et al. (2023), the fuzziness parameter m was set to 1.4 to ensure convergence in all simulations. Generally speaking, fuzzy clustering requires fewer groups to describe the data, as it also describes hybrid types.

Figure 3 compares FCMdd and fuzzy CLARA using the same strategy as for crisp clustering. It leads to a quite different conclusion. First, all values are negative, meaning that fuzzy CLARA generally performs better than FCMdd. Second, the performance shows a local maximum around for 1,000 sequences, and

decreases with a subsample of 2,500. These two surprising results indicate that the optimization strategy of FCMdd might be overwhelmed or trapped in a local maximum when too many observations are used. The subsampling strategy of CLARA overcomes this limit.[4] As before, more than one iteration is better, but only slight improvement is observed with more than 200 iterations.

3.4 Guidelines

The aim of the simulation was to assess the relative performance of the algorithms and draw guidelines on the parameters' values. Even if one cannot issue general guidelines based on a single dataset, the results provide interesting insights. For crisp clustering, the required number of iterations and the sample size mostly depend on the expected number of types. A higher number of groups requires more iterations. Furthermore, the sample size is more important than the number of iterations. In other words, one would prefer a larger sample size at the cost of fewer iterations. Practically, using a sample size between 1,000 and 10,000, and 500 hundred iterations provided good results. If one is interested in a more detailed typology, with by definition infrequent types, increasing the subsample size and the number of iterations is safer. We further develop in section 5.4 an indicator to check whether a sufficient number of iterations have been used.

For fuzzy clustering, a subsample size of approximately one thousand provided the best results. This might be sample dependent, but it indicates that a smaller sample size might provide better results. In our simulations, 200 iterations were generally enough with only small improvements with more iterations.

4 Distance Measures

Even if it relies on subsamples, the CLARA algorithm requires the computation of many distances. More precisely, the distances within the subsample as well as

Distance	Complexity	Expected Gain Ratio
Optimal Matching	ℓ^2	1
Optimal Matching of Transition	ℓ^2	1
Optimal matching of Spell	s^2	23.1
SVRspell	s^3	4.3
Hamming	ℓ	24

Table 1: Computational Complexity of the Main Dissimilarity Measures Used in Sequence Analysis, ℓ refers to sequence length and s the number of spells within sequences. The expected gain ratio provides indicative ratio for the illustrative dataset.

the distance between all observations and each medoid should be computed at each iteration, resulting in the computation of $I \times (k \times N + n(n-1)/2)$ distances. The computation of SA distance is more complex than most distances between metric data, such as the Euclidean distance. For this reason, the complexity of the distance calculation should be taken into account when choosing the distance measure.

Table 1 presents the approximate computational complexity following (Elzinga and Studer 2015) of the main distances measures recommended by Studer and Ritschard (2016), namely optimal matching of transitions, optimal matching of spells or the Hamming distance. This complexity depends either on ℓ, the length of the sequences, or on s, the number of spells within the sequences. Following Elzinga and Studer (2015), we further present the expected computational gain ratio for our illustrative application. It is computed as the average number of operations required by a given distance measure divided by this number for standard optimal matching. According to these numbers, the Hamming distance requires approximately 24 (the length of the sequences) times fewer operations than usual optimal matching, resulting in a much faster distance computation.

The difference between these measures mainly depends on the ratio between ℓ and s. When s is smaller than ℓ, as it is almost always the case in social sciences, spell-based and Hamming distances are faster than optimal matching distance. The

optimal matching of spells is even more efficient than Hamming when the number of spells per sequence is lower than $\sqrt{\ell}$, which is often the case when using monthly data, for instance. Otherwise, the Hamming distance features among the most efficient.

The choice of a distance measure should always be made according to the research question, as it defines how the trajectories are compared (Studer and Ritschard 2016). Computational efficiency should therefore not be the primary criteria to choose a distance measure. However, there are often several measures suitable for a given research question. Hamming and Optimal Matching of Spells are covering a wide range of situations depending on their parameters' values. Computational efficiency can then be taken into account as a secondary criterion.

5 Cluster Quality

Once clustering has been computed, one typically computes several cluster quality indices (CQI). These indices can be used to guide the choice of the number of groups or to tone the interpretation of the resulting typology (Studer 2013, 2021). The CQI generally used in SA are based on the whole distance matrix and their computation therefore face the same computational and memory issues as the clustering itself. In this section, we discuss three different approaches to evaluate the clustering quality in a large database.

First, one can compute the quality of the typology using bootstraps, i.e., random subsamples of the data. This approach allows for computing the CQI commonly used in SA with a confidence interval to account for subsampling uncertainty.

Second, one can rely on CQI computed using only the distances between each sequence and the medoids, which are already computed within CLARA. In this article, we present some of these CQI and adapt others to do so. We further develop crisp and fuzzy versions of these CQI when they do not exist.

Finally, one can rely on cluster stability to estimate the quality of the typology

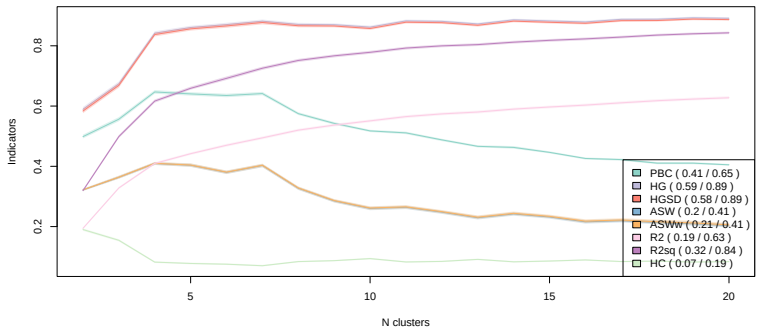


Figure 4: Cluster Quality Indices Between 2 and 20 Groups Estimated by Bootstrap With Corresponding Confidence Intervals.

(Hennig 2007, 2008). The general idea of the approach is as follows. In every CLARA iteration, a distinct typology is created. If a strong clustering structure is found in the data, we should retrieve similar grouping in each iteration. However, if the data are poorly structured, different clustering solutions should pop up at each iteration. The stability of the clustering thus informs on the underlying clustering structure of the data, which is directly linked to the quality of the typology.

5.1 Bootstrapping Cluster Quality Indices

The cluster quality of a given typology can be estimated using a subsample of the data. By doing so, any cluster quality index (CQI) can be used (see Studer 2013, for a review). This is a great advantage, since these indices are already widely used in SA.

This strategy requires us to account for the potential sample dependence of the results, or in other words, the potential estimation error due to the sampling procedure. The null distributions of most CQI are unknown. It is therefore impossible to derive their standard error analytically. We propose here to rely on bootstrapping. To minimize the estimation error, the sampling is stratified by the clustering solution in the highest number of groups.

Figure 4 presents several CQI for different numbers of groups for our sample

application. We used a sample size of 5,000 sequences. The 95% confidence intervals are represented using a lighter polygon. Interestingly, these confidence intervals are very thin, showing that the estimation procedure is precise enough. The bootstrapped CQI indicate a clustering in 4 or 7 groups.

5.2 Medoid-Based Cluster Quality Indices

In this subsection, we present four CQI that can be directly computed from the distances to the medoids including the Davies-Bouldin index (Davies and Bouldin 1979), the Average Medoid Silhouette (Lenssen and Schubert 2022), the XB (Xie and Beni 1991) and the PBM index (Pakhira et al. 2004). While some of these CQI are readily available, others require to be adapted. Furthermore, we present or develop, when required, a fuzzy and a crisp clustering version of these CQI using the framework proposed by Sledge et al. (2010).

5.2.1 Davies-Bouldin Index

The Davies-Bouldin (DB) index aims to measure the statistical quality of a clustering by balancing two aspects of each cluster (Davies and Bouldin 1979): The within-cluster homogeneity and the between-cluster separation. It relies on the notion of cluster centers. While the barycenters (i.e., average points) are generally used, medoids can be used as well. The homogeneity W_k of cluster k is measured using the within-cluster average distance to its medoid m_k, $W_k = \frac{1}{N_k} \sum_{i \in C_k} d(i, m_k)$ with N_k the size of the k^{th} cluster. The between-cluster separation $B_{k\ell}$ is measured using the distance between the cluster centers k and ℓ, i.e., the medoids, $B_{k\ell} = d(m_k, m_\ell)$.

The DB index then computes for each cluster the ratio between cluster homogeneity and separation and combines them as follows.

$$DB = \frac{1}{K} \sum_{i=k}^{K} \max_{k \neq \ell} \frac{W_k + W_\ell}{B_{k\ell}} \tag{5}$$

The DB index can be adapted to fuzzy clustering with representatives. Indeed,

the measure of the between-cluster separation $B_{k\ell}$ remains unchanged. The within-cluster homogeneity should now account for the gradual membership u_{ik} of each sequence i to cluster k. This can be achieved by weighting the average distance W_k by the membership strength (Sledge et al. 2010):

$$W_k = \frac{\sum_i^N u_{ik} d(i, m_k)}{\sum_i^N u_{ik}} \tag{6}$$

The DB index should be minimized to obtain well-separated homogeneous clusters.

5.2.2 XB Index

The Xie-Beni (XB) index is one of the most used fuzzy CQI (Xie and Beni 1991). It can also be employed for crisp clustering. More generally, any fuzzy CQI can also evaluate crisp partitions by using a membership matrix filled with 0 and 1. As the DB index, the XB index balances within-cluster homogeneity, measured again using the weighted distances to cluster centers, and the worst between-cluster separation, measured using the minimal distances between two cluster centers. It is computed with the following formula:

$$XB = \frac{\sum_i^N \sum_k^K u_{ik}^m d(i, m_k)}{N * \min_{k \neq \ell} d(m_k, m_\ell)} \tag{7}$$

The XB index should then be minimized. A low value indicates homogeneous clusters that are clearly separated.

5.2.3 PBM Index

Named after its authors, the PBM index (Pakhira et al. 2004) explicitly balances three aspects: parsimony, homogeneity and separation. It relies on the notion of barycenters (i.e., average points), which we replace here by the medoids. It is computed as follows:

$$PBM = \Big(\frac{1}{K} \times \frac{E_1}{E} \times D\Big)^2 \tag{8}$$

With K the number of groups, $E = \sum_i^N \sum_k^K u_{ik} d(i, m_k)$ to measure homogeneity, and $D = \max_{k,\ell} d(m_k, m_\ell)$ to measure cluster separation. E_1 is a constant, which captures the no-cluster homogeneity. However, since finding the overall medoid is a complicated task in large databases, and it has no impact on the resulting choice of a number of groups, we arbitrarily set $E_1 = 1$ in our implementation of the index.

The PBM index should be maximized to identify parsimonious typologies, with homogeneous and well-separated clusters.

5.2.4 Average Medoid Silhouette

The average medoid silhouette (AMS) (Lenssen and Schubert 2022) aims to simplify the computation of the well-known average silhouette width index (Kaufman and Rousseeuw 1990). This is achieved by using distances to medoids instead of averages of distances to other sequences. Also named simplified silhouette width, this index was shown to be almost as efficient as its original version (Hruschka et al. 2006).

Conceptually, the AMS index measures the coherence, at the observation level, of the overall clustering. This is achieved by comparing the distance of each sequence i to the medoids of its own cluster a_i and the distance to the medoid of the closest other cluster b_i. The coherence is then measured with the silhouette $s_i = \frac{a_i - b_i}{\max(a_i, b_i)}$. The final AMS index is given by the average of the individual silhouette values ($AMS = \frac{1}{N} \sum s_i$).

Campello and Hruschka (2006) proposed a fuzzy extension of this index named fuzzy silhouette (FS), which reads as follows:

$$FS = \frac{\sum_{i=1}^N (u_{ia} - u_{ib})^\alpha s_i}{\sum_{i=1}^N (u_{ia} - u_{ib})^\alpha} \tag{9}$$

With $u_i a$ the highest cluster membership of sequence i, u_{ib} the second highest one, and α a user-defined parameter. The aim of fuzzy clustering is to take into

account hybrid-type sequences, which are heavily penalized by the silhouette. The α parameter controls this penalization by allowing weighting the average of the individual silhouettes using the difference between the two highest membership strengths. When $\alpha = 0$ or when the clustering is crisp, FS equals the AMS index. For fuzzier clustering, hybrid sequences are less taken into account as α increases. Campello and Hruschka (2006) propose to use $\alpha = 1$ by default.

The AMS and the FS values range between -1 and 1, with 1 indicating a perfectly coherent clustering. According to our experiments, the AMS values tend to be higher than the average silhouette width. The interpretation thresholds proposed by Kaufman and Rousseeuw (1990) should therefore not be used with these indices.

5.3 Conclusion

We presented four CQI to measure the quality of a fuzzy or a crisp clustering in a large database. Their properties and interpretation are summarized in Table 2.

Name	Interval	Min/Max	Interpretation
Davies-Bouldin (DB)	$[0; +\inf[$	Min	Well-separated homogeneous clusters measured at the cluster level
Xie-Beni (XB)	$[0; +\inf[$	Min	Well-separated homogeneous clusters measured as the worst overall case
PBM	$]0; +\inf[$	Max	Parsimonious well-separated homogeneous clusters overall best case
PBMK	$]0; +\inf[$	Max	PBM less penalized by complexity
Average Medoid Fuzzy Silhouette (AMS)	$[-1; 1]$	Max	Well-separated homogeneous clusters measured at the observational level

Table 2: Summary of Medoid-Based Cluster Quality Indices

Conceptually, they measure cluster quality differently and might all be of interest in some situations. As most of the presented CQI, the DB index balances within-cluster homogeneity and the between-cluster separation. Importantly, these

aspects are measured at the cluster level and then combined using the average. The XB index uses the same two aspects. However, they are measured using the overall within-cluster homogeneity and the worst separation between two clusters. By focusing on the worst separation, it is expected to heavily penalize higher number of groups, as adding a new cluster, typically results in lower distances between cluster centers. The PBM index adds an explicit penalization of the number of groups. This penalization can be lowered resulting in the PBMK index. Finally, the AMS and FS indices balance homogeneity and separation at the observation level. Furthermore, the fuzzy version of the index penalizes less hybrid sequences, which are generally better described by fuzzy clustering.

Figure 5 presents the values of these indices for our sample application. These indices are typically used to choose the number of groups of the resulting typology (Studer 2013). The DB and the AMS indices opt for the 4 and 7 cluster solutions. The PBM index, which explicitly penalizes complexity, favors 2 or 4 groups, but local maxima are found for 7 and 11 clusters. The PBMK index orders the same solutions differently, penalizing less complexity as expected. Finally, the XB index leads to the choice of the 2 or 4 clusters solutions. As expected, this index heavily penalizes complexity, as it strongly increases with the number of clusters.

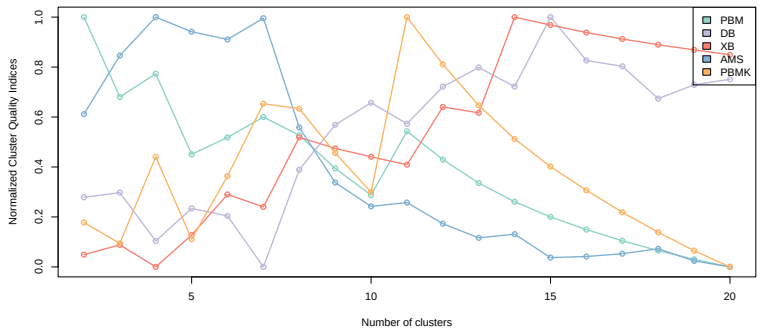


Figure 5: Values of the DB, ASM, PBM, PBMK, and XB indices according to the number of clusters.

5.4 Clustering Stability

The stability of a clustering method can be used to estimate the quality of the resulting typology, but also informs on the underlying clustering structure of the data (Hennig 2007, 2008). Clustering stability is generally estimated using subsampling procedures such as bootstraps. We propose to rely on the CLARA subsampling procedure to implement a similar approach. The general idea is as follows. In each iteration of the CLARA algorithm, a distinct typology is created. If a strong clustering structure is found in the data, the CLARA algorithm should retrieve a similar clustering solution at (almost) each iteration. On the contrary, if the data is poorly structured in subgroups, different clustering solutions might arise in each iteration.

The stability is measured by looking at the similarity of clustering solutions (Ben-Hur et al. 2001). Two similarity measures can be used: the Jaccard Coefficient (JC) (Hennig 2007) and the Adjusted Rand Index (ARI) (Hubert and Arabie 1985). The JC index examines whether the sequences are clustered in the same cluster in the two partitions to define similarity. The ARI further looks at the pairs of objects clustered into different clusters in both partition as a factor of similarity between clusterings.

The stability index is computed using the average similarity between the retained solution (i.e., the best CLARA iteration) and the other iterations. To mitigate the impact of the worst CLARA iterations, the average among the best 20% of the iterations is also computed. Figure 6 presents these values for different numbers of groups. Looking at the trimmed average, solutions between 2 and 6 groups, 8, 9 or 11 groups feature among the most stable. Interestingly, these results are in line with the conclusion drawn using the CQI, except regarding the 7-cluster solution. The latter is found to be much less stable.

Looking at the stability of the CLARA typologies can also inform us on the convergence of the algorithm. If solutions similar to the current best were found several times, we can be more confident that the algorithm explored a solution

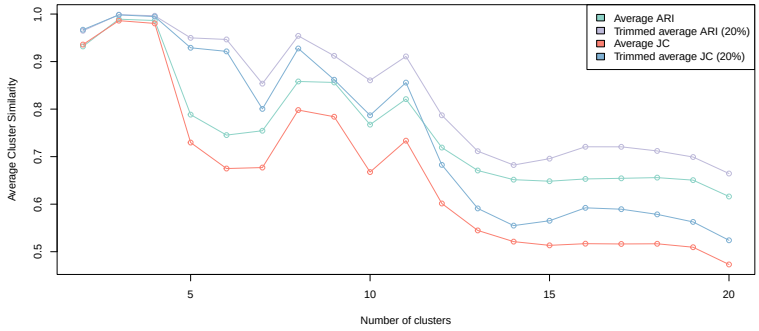


Figure 6: Clustering Stability Among CLARA Iterations Measured With the Adjusted Rand Index and the Jaccard Coefficient.

space large enough. In our simulations, we recorded at each iteration the number of time that a solution similar to the current one was found in previous iterations. The results showed that, on average, having recovered at least five times a solution similar to the current best (ARI$\geq$ 0.8) is strongly associated with outperforming the PAM algorithm (64% vs. 10%). However, a more detailed look at the results showed that the ARI threshold to use depends on the number of groups. An ARI threshold of 0.9 should be favored for a fewer number of groups, while 0.7 should be used for a higher one.

Figure 8 in the Appendix plots this information by counting the number of times that an iteration leads to a solution similar to the best one. We used the three ARI thresholds to consider a clustering as similar to the best one: strong (ARI$\geq$ 0.9), good (ARI$\geq$ 0.8) and weak (ARI$\geq$ 0.7). Similarly, we used three JC thresholds, which also have theoretical foundations (see Hennig 2008, for a full discussion): strong (JC$\geq$ 0.8), weak (JC$\geq$ 2/3) and minimal (JC$\geq$ 0.5).

We observe an important drop in the number of recoveries above 11 groups. If a very detailed description of the trajectories is needed, we might therefore want to increase the iterations. However, the number of recoveries with ARI$\geq$ 0.8 is always greater than 5.

The stability of the clustering usefully complements the CQI by providing

information on the strength of the underlying clustering structure, but also to assess whether a sufficient number of iterations were used.

5.5 Conclusion

We reviewed three approaches to measure the quality of the clustering, each having their own strengths and weaknesses. The bootstrapping approach allows using the "standard" CQI, which are well known within the SA community. This is an advantage, but it requires further computations. However, these indices are not readily available for fuzzy clustering. Medoid-based CQI are directly computed within CLARA and can be computed for fuzzy clustering. However, they are less well known. Finally, the stability approach informs us on the underlying clustering structure of the data, and on the convergence of the algorithm. It is therefore complementary to the previous approaches.

6 Clustering Sequences in Large Data With `R`

In this section, we illustrate the use of the CLARA algorithm through a practical example: the clustering of transition to adulthood trajectories in Switzerland. The data are made publicly available in the `TraMineR` package (Gabadinho et al. 2011). The presented CLARA algorithm and the different strategies to evaluate the clustering quality in large databases are made available in the `WeightedCluster R` library (Studer 2013). We start by loading the required library.

```
library(WeightedCluster)
```

6.1 Data Preparation

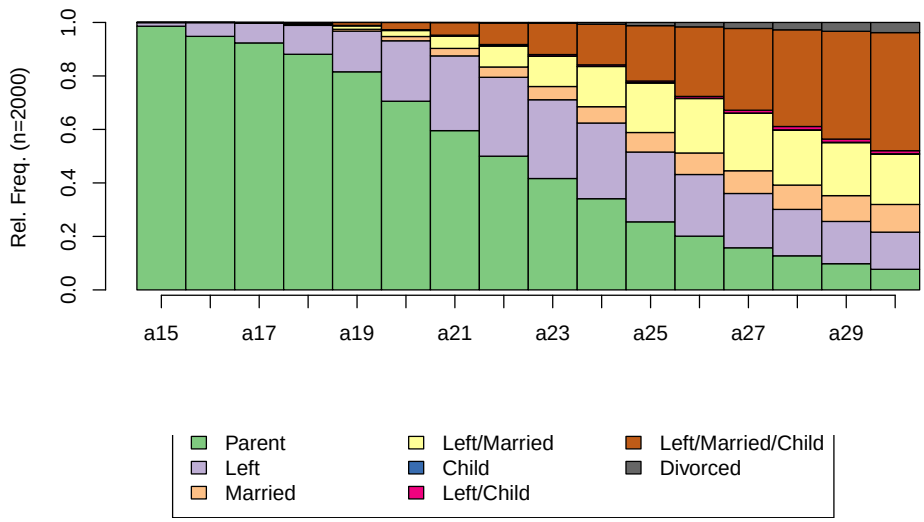
First, we need to prepare the data by creating a state sequence object using the `seqdef` command (see Gabadinho et al. 2011, for a full introduction). This object stores all the information about our trajectories, including the data and associated

characteristics, such as state labels. During this step, we further define proper state labels as the original data are coded using numerical values.

```
data(biofam)  #load illustrative data
## Defining the new state labels
statelab <- c("Parent", "Left", "Married", "Left/Married",
    "Child", "Left/Child", "Left/Married/Child", "Divorced")
## Creating the state sequence object,
biofam.seq <- seqdef(biofam[, 10:25], alphabet = 0:7,
    states = statelab)
```

We can now plot our sequences using, for instance, a chronogram.

```
seqdplot(biofam.seq, legend.prop = 0.2)
```



6.2 CLARA Clustering

CLARA for SA is available in the `seqclararange` function. It is used as follows:

```
bfclara <- seqclararange(biofam.seq, R = 100, sample.size = 200,
    kvals = 2:10, seqdist.args = list(method = "LCS"),
    parallel = TRUE, stability = TRUE)
```

The function requires us to specify the sequence to cluster (our `biofam.seq` sequence object), the number of iterations (argument `R`, here set to 100 to avoid long computation time, larger values are recommended) and the subsample size (argument `sample.size`, here again set to the low value of 200 for illustrative purpose). The number of groups in our typology is set using the `kvals` arguments, here between 2 and ten groups solutions. We directly specify a range of values that will be considered later on. Finally, we need to specify how to compute the distance between sequences through the `seqdist.args` argument as a `list` object. All the arguments specified here will be directly passed to the `seqdist` function. Therefore, any distance measures available in `seqdist` can be used here. Finally, we set `stability=TRUE` to estimate the stability of the clusterings among the subsamples.

Setting `parallel=TRUE`, a default parallel back-end is set up using the future framework (Bengtsson 2021). However, any parallel back-end previously defined with the `plan` function will be used when `parallel=FALSE`. The parallel protocol can then be adapted to specific environments, for instance some High Performance Computing (HPC) server relies on specific protocols (MPI,...). As implied by the name, setting `progressbar=TRUE` shows information (and estimated computation time) on the progress of the computations.

6.3 Cluster Quality Indices

The values of the medoid-based CQI are shown when the result is printed, or can be plotted using the `plot` command. When plotting the CQI, standardizing the values makes it easier to identify the best solution (see Studer 2013).

```
bfclara
```

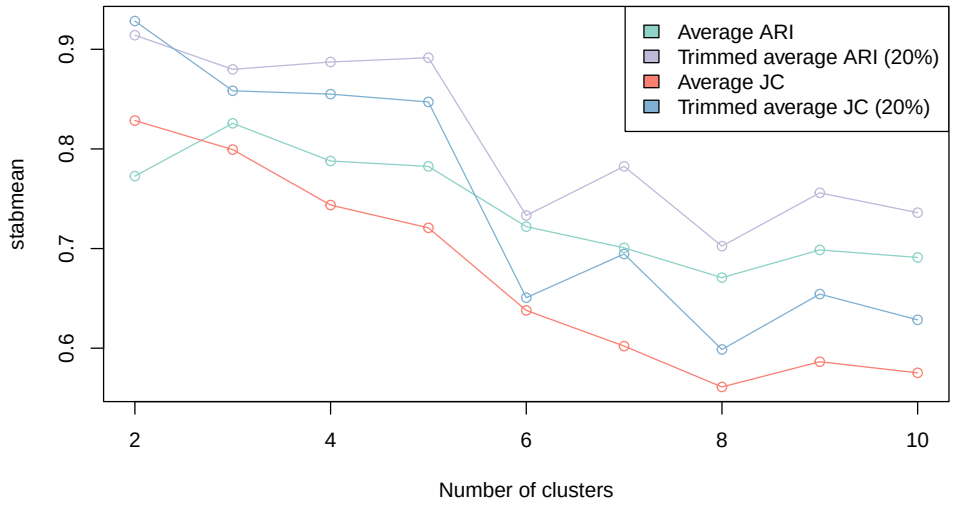
```
##            Avg dist  PBM   DB   XB  AMS ARI>0.8 JC>0.8 Best iter
## cluster2       9.50 0.90 1.02 0.53 0.50      70     76         93
## cluster3       7.73 0.60 1.02 0.55 0.51      62     53         11
## cluster4       6.72 0.67 0.84 0.48 0.54      55     45         39
## cluster5       5.81 0.47 0.72 0.41 0.59      48     32         99
## cluster6       5.26 0.49 0.90 0.53 0.55      23     10         59
## cluster7       4.79 0.43 0.95 0.60 0.55      15      3         76
## cluster8       4.50 0.44 0.95 0.56 0.54       4      1         98
## cluster9       4.23 0.33 0.96 0.53 0.55       4      1         83
## cluster10      4.04 0.55 0.90 0.50 0.56       6      1         14
```

```
plot(bfclara, norm = "range")
```

Except the PBM index, all indices favor a five-cluster solution. The resulting clustering is stored in the `clustering` element of the results. It can for instance be used to represent the sequences in each cluster as follows.

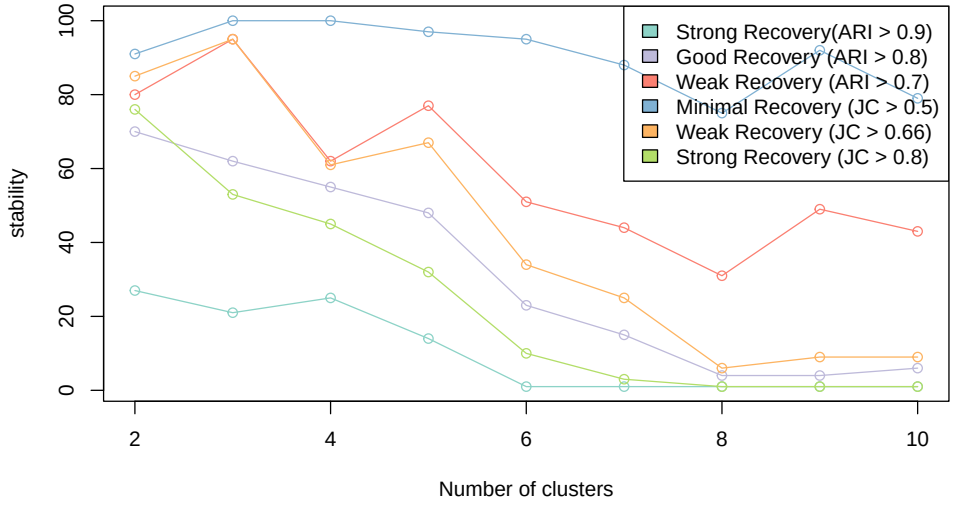
The stability of the clustering can also be plotted using either the average value, or the number of recoveries of a similar solution.

```
plot(bfclara, stat = "stabmean")
```



Here again, the five-cluster solution shows the highest CQI values. However, the absolute number of recoveries is low for more than seven groups. A higher number of iterations is therefore recommended. This is not surprising as we used a low number of iterations.

```
plot(bfclara, stat = "stability")
```



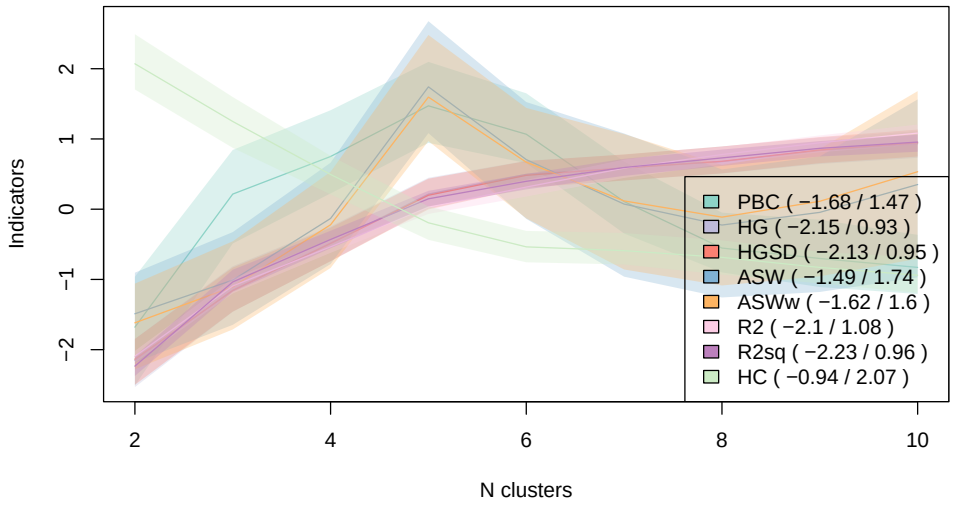
The `bootclustrange` function can be used to bootstrap the cluster quality measures. It has the following arguments. The first object is the clustering to be evaluated, as a `seqclararange` object, a `data.frame` or a vector. Second, the sequences that were used to create the typology. We should further specify the distance measure to use (as before), the number of bootstraps (`R` argument), and the subsample size (here 500). We would generally use higher values for this last two arguments. Finally, the `parallel` and `progressbar` arguments could be used as before.

```
bCQI <- bootclustrange(bfclara, biofam.seq, seqdist.args = list(method = "LCS"),
    R = 100, sample.size = 500, parallel = TRUE)
```

The resulting object can then be printed. The standardized value of the CQI can also be plotted. The results lead to the same conclusion as for the medoid-based CQI.

```
bCQI

##            PBC   HG HGSD  ASW ASWw     CH   R2   CHsq R2sq   HC
## cluster2  0.51 0.63 0.61 0.36 0.36 136.34 0.21 277.47 0.36 0.18
```

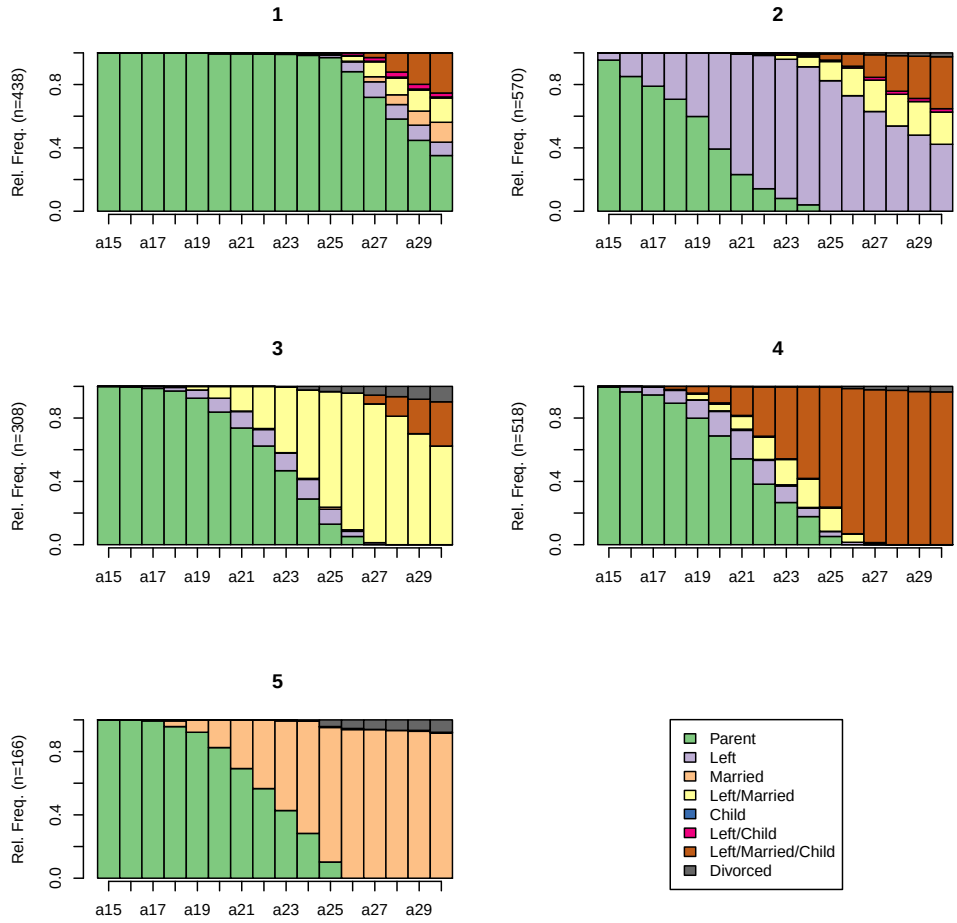
```
## cluster3  0.57 0.73 0.70 0.37 0.37 135.08 0.35 292.57 0.54 0.15
## cluster4  0.59 0.79 0.77 0.39 0.40 124.28 0.43 284.00 0.63 0.12
## cluster5  0.61 0.85 0.84 0.43 0.44 126.37 0.51 318.88 0.72 0.09
## cluster6  0.60 0.88 0.86 0.41 0.42 117.65 0.54 308.84 0.76 0.08
## cluster7  0.57 0.89 0.87 0.40 0.40 113.83 0.58 305.67 0.79 0.08
## cluster8  0.55 0.90 0.88 0.39 0.40 108.14 0.61 296.67 0.81 0.07
## cluster9  0.54 0.91 0.90 0.39 0.40 104.09 0.63 296.15 0.83 0.07
## cluster10 0.54 0.92 0.91 0.40 0.41 100.22 0.65 289.89 0.84 0.06

plot(bCQI, norm = "zscore")
```



6.4 Plotting the Typology

Once a clustering in a given number of groups has been selected, we can plot the sequences by cluster to give a better interpretation.

```
seqdplot(biofam.seq, group = bfclara$clustering$cluster5)
```



6.5 Fuzzy Clustering

By setting `method="fuzzy"` in the `seqclararange` function, the fuzzy version of the algorithm is used. It should be noted that the computations are generally longer than for crisp clustering.

```
bfclaraf <- seqclararange(biofam.seq, R = 100, sample.size = 200,
    kvals = 2:10, method = "fuzzy", seqdist.args = list(method = "LCS"),
    parallel = TRUE)
```

The analyses then follow the same logic. The CQI values can be printed and

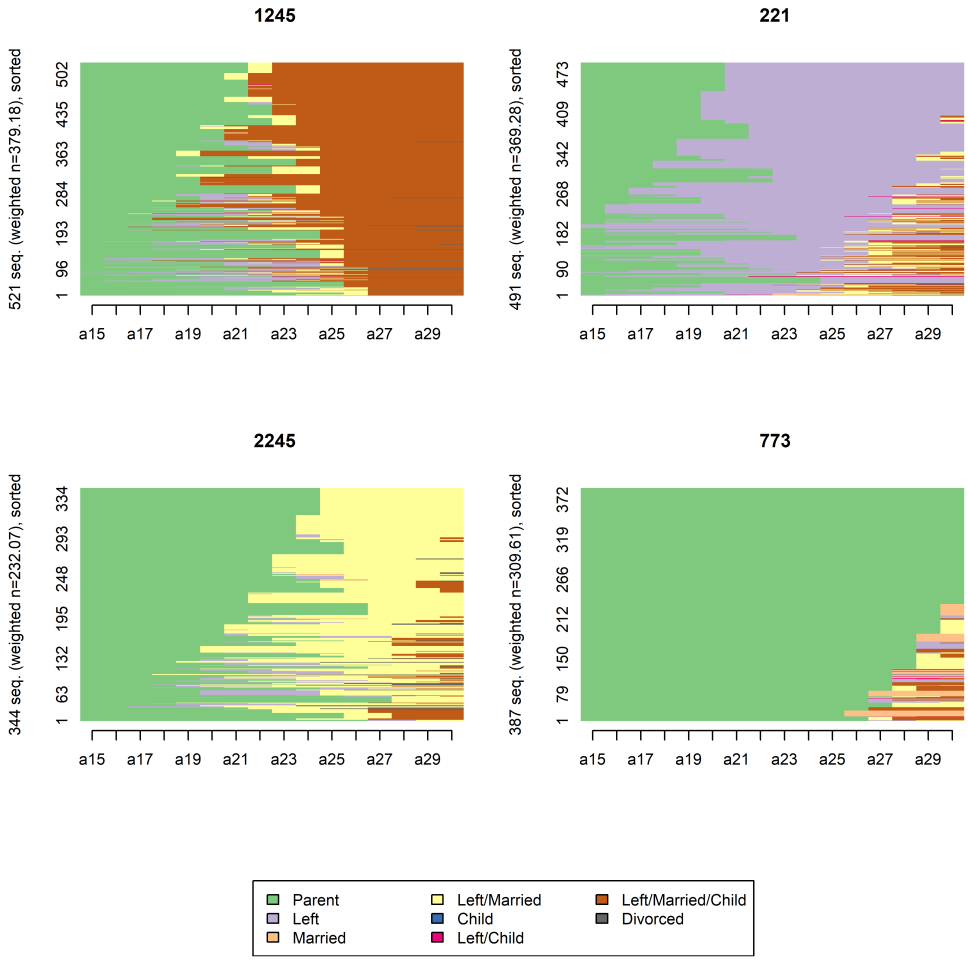
plotted using the same command.

```
bfclaraf

##            Avg dist  PBM   DB   XB  AMS ARI>0.8 JC>0.8 Best iter
## cluster2       8.02 0.69 1.21 0.45 0.62      NA     NA        49
## cluster3       5.96 0.49 1.20 0.43 0.70      NA     NA         4
## cluster4       4.90 0.32 1.28 0.41 0.74      NA     NA        47
## cluster5       4.22 0.22 1.71 0.53 0.74      NA     NA         6
## cluster6       3.70 0.21 1.44 0.37 0.74      NA     NA        37
## cluster7       3.30 0.17 1.53 0.41 0.75      NA     NA         6
## cluster8       3.03 0.14 1.61 0.38 0.74      NA     NA        40
## cluster9       2.81 0.12 1.71 0.47 0.75      NA     NA        31
## cluster10      2.61 0.12 1.94 0.43 0.73      NA     NA        10

plot(bfclaraf, norm = "zscore")
```

The XB and AMS index favor a four or six cluster solution. Several plots are available to describe fuzzy clustering of trajectories, see Studer (2018) and the `fuzzyseqplot` function. Here, we use sequence index plot ordered by membership strength, with typical trajectories of each cluster represented at the top, leaving

aside trajectories with low membership. In this kind of graphic, hybrid trajectories are represented at the bottom.

```
fuzzyseqplot(biofam.seq, group = bfclaraf$clustering$cluster4,
    type = "I", sortv = "membership", membership.threashold = 0.4)
```



7 Conclusion

This article aimed to adapt the CLARA algorithm for SA, and to extend it to clustering approaches recommended by Helske et al. (2023), including fuzzy clustering and representativeness. We further developed three approaches to

measure the quality of the resulting clustering, which faces the same computational issues: bootstrapping "standard" CQI, using Medoid-based CQI, and a stability approach. All these developments are made available in the `WeightedCluster` library (Studer 2013). A step-by-step example on how to use the library is provided in Section 6.

REFERENCES

Abbott, Andrew and John Forrest. 1986. "Optimal Matching Methods for Historical Sequences." *Journal of Interdisciplinary History* 16:471–494.

Ben-Hur, Asa, Andre Elisseeff, and Isabelle Guyon. 2001. "A stability based method for discovering structure in clustered data." In *Biocomputing 2002*. WORLD SCIENTIFIC.

Bengtsson, Henrik. 2021. "A Unifying Framework for Parallel and Distributed Processing in R using Futures." *The R Journal* 13:208.

Börsch-Supan, Axel, Martina Brandt, Christian Hunkler, Thorsten Kneip, Julie Korbmacher, Frederic Malter, Barbara Schaan, Stephanie Stuck, and Sabrina Zuber. 2013. "Data Resource Profile: The Survey of Health, Ageing and Retirement in Europe (SHARE)." *International Journal of Epidemiology* 42:992–1001.

Buchmann, Marlis C. and Irene Kriesi. 2011. "Transition to Adulthood in Europe." *Annual Review of Sociology* 37:481–503.

Campello, R.J.G.B. and E.R. Hruschka. 2006. "A fuzzy extension of the silhouette width criterion for cluster analysis." *Fuzzy Sets and Systems* 157:2858–2875.

Dave, R.N. and S. Sen. 2002. "Robust fuzzy clustering of relational data." *IEEE Transactions on Fuzzy Systems* 10:713–727.

Davies, David L. and Donald W. Bouldin. 1979. "A Cluster Separation Measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1:224–227.

De Cáceres, Miquel, Xavier Font, and Francesc Oliva. 2010. "The management of vegetation classifications with fuzzy clustering: Fuzzy clustering in vegetation classifications." *Journal of Vegetation Science* 21:1138–1151.

D'Urso, Pierpaolo. 2016. "Fuzzy Clustering." In *Handbook of Cluster Analysis*, edited by Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci, pp. 545—573. Chapman & Hall.

Elzinga, Cees H. and Matthias Studer. 2015. "Spell Sequences, State Proximities and Distance Metrics." *Sociological Methods and Research* 44:3–47. published online 9 July 2014.

Gabadinho, Alexis, Gilbert Ritschard, Nicolas S. Müller, and Matthias Studer. 2011. "Analyzing and visualizing state sequences in R with TraMineR." *Journal of Statistical Software* 40:1–37.

Gauthier, Jacques-Antoine, F. Bühlmann, and Philippe Blanchard. 2014. "Introduction: Sequence Analysis in 2014." In *Advances in Sequence Analysis: Theory, Method, Applications*, edited by Philippe Blanchard, F. Bühlmann, and Jacques-Antoine Gauthier, volume 2 of *Life Course Research and Social Policies*. Heidelberg: Springer.

Helske, Satu, Jouni Helske, and Guilherme K. Chihaya. 2023. "From Sequences to Variables: Rethinking the Relationship between Sequences and Outcomes." *Sociological Methodology* .

Hennig, Christian. 2007. "Cluster-wise assessment of cluster stability." *Computational Statistics and Data Analysis* 52:258–271.

Hennig, Christian. 2008. "Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods." *Journal of Multivariate Analysis* 99:1154–1176.

Hruschka, Eduardo R., Ricardo J.G.B. Campello, and Leandro N. de Castro. 2006. "Evolving clusters in gene-expression data." *Information Sciences* 176:1898–1927.

Hubert, L and P Arabie. 1985. "Comparing Partitions." *Journal of Classification* 2:193–218.

Kaufman, L. and P. J. Rousseeuw. 1990. *Finding groups in data. an introduction to cluster analysis*. New York: Wiley.

Krishnapuram, R., A. Joshi, and Liyu Yi. 1999. "A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering." In *FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315)*. IEEE.

Landoes, Aljoscha. 2022. *Inégalités scolaires durant la transition vers l'éducation post-obligatoire en Suisse. L'influence du lieu de résidence et du motif d'immigration*. Ph.D. thesis, University of Geneva.

Lenssen, Lars and Erich Schubert. 2022. *Clustering by Direct Optimization of the Medoid Silhouette*, pp. 190–204. Springer International Publishing.

Liao, Tim F., Danilo Bolano, Christian Brzinsky-Fay, Benjamin Cornwell, Anette Eva Fasang, Satu Helske, Raffaella Piccarreta, Marcel Raab, Gilbert Ritschard, Emanuela Struffolino, and Matthias Studer. 2022. "Sequence analysis: Its past, present, and future." *Social Science Research* 107:102772.

Liao, Tim Futing and Anette Eva Fasang. 2020. "Comparing Groups of Life-Course Sequences Using the Bayesian Information Criterion and the Likelihood-Ratio Test." *Sociological Methodology* 51:44–85.

Liefbroer, Aart C. 2019. "Methodological diversity in life course research: Blessing or curse?" *Advances in Life Course Research* 41:100276.

Lorentzen, Thomas, Olof Bäckman, Ilari Ilmakunnas, and Timo Kauppinen. 2018. "Pathways to Adulthood: Sequences in the School-to-Work Transition in Finland, Norway and Sweden." *Social Indicators Research* 141:1285–1305.

Losa, Fabio B., Maurizio Bigotta, Eric Stephani, and Gilbert Ritschard. 2014. *D'où venons-nous? Que sommes-nous? Où allons-nous? Analyse des parcours professionnels des chômeurs de longue durée en Suisse*. Giubiasco: Ufficio di Statistica, TI.

Mayer, Karl Ulrich. 2009. "New Directions in Life Course Research." *Annual Review of Sociology* 35:413–433.

Müllner, Daniel. 2013. "fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python." *Journal of Statistical Software* 53:1–18.

Ng, R.T. and Jiawei Han. 2002. "CLARANS: a method for clustering objects for spatial data mining." *IEEE Transactions on Knowledge and Data Engineering* 14:1003–1016.

Pakhira, Malay K., Sanghamitra Bandyopadhyay, and Ujjwal Maulik. 2004. "Validity index for crisp and fuzzy clusters." *Pattern Recognition* 37:487–501.

Pesando, Luca Maria, Nicola Barban, Maria Sironi, and Frank F. Furstenberg. 2021. "A Sequence-Analysis Approach to the Study of the Transition to Adulthood in Low- and Middle-Income Countries." *Population and Development Review* 47:719–747.

Piccarreta, Raffaella and Matthias Studer. 2019. "Holistic analysis of the life course: Methodological challenges and new perspectives." *Advances in Life Course Research* 41:100251. Theoretical and Methodological Frontiers in Life Course Research.

Schubert, Erich and Peter J. Rousseeuw. 2019. "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms." In *Similarity Search and Applications*, pp. 171–187. Springer International Publishing.

Shanahan, Michael J. 2000. "Pathways to Adulthood in Changing Societies: Variability and Mechanisms in Life Course Perspective." *Annual Review of Sociology* 26:pp. 667–692.

Sledge, I J, J C Bezdek, T C Havens, and J M Keller. 2010. "Relational Generalizations of Cluster Validity Indices." *IEEE Transactions on Fuzzy Systems* 18:771–786.

Studer, Matthias. 2013. "WeightedCluster Library Manual: A practical guide to creating typologies of trajectories in the social sciences with R." LIVES Working Papers 24, NCCR LIVES, Switzerland.

Studer, Matthias. 2018. "Divisive Property-Based and Fuzzy Clustering for Sequence Analysis." In *Sequence Analysis and Related Approaches: Innovative Methods and Applications*, edited by Gilbert Ritschard and Matthias Studer, volume 10 of *Life Course Research and Social Policies*, chapter 13, pp. 223–239. Springer.

Studer, Matthias. 2021. "Validating Sequence Analysis Typologies Using Parametric Bootstrap." *Sociological Methodology* 51:290–318.

Studer, Matthias, Sinisa Hadziabdic, and Gilbert Ritschard. 2015. "Analyse des trajectoires des chômeurs en fin de droits dans le Canton de Genève." Rapport final, Institut d'études détudes démographiques et du parcours de vie, Université de Genève.

Studer, Matthias and Gilbert Ritschard. 2016. "What Matters in Differences between Life Trajectories: A Comparative Review of Sequence Dissimilarity Measures." *Journal of the Royal Statistical Society, Series A* 179:481–511.

von Gunten, Luzius, Nora Meister, Philippe Meyer, and Thomas Ruch. 2019. *ML_SoSi: Verläufe im System der Sozialen Sicherheit*. Number 9847916 in Experimental Statistics. Office fédéral de la statistique.

Xie, X.L. and G. Beni. 1991. "A validity measure for fuzzy clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13:841–847.

A Further Results

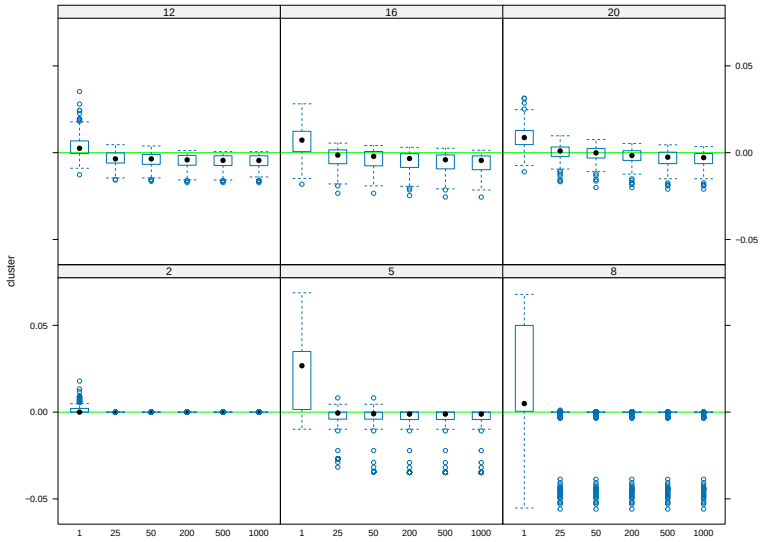


Figure 7: Differences between the quality of CLARA and PAM (y-axis) for varying numbers of iterations (x-axis) and numbers of groups (panels) when the sample size is 10,000.

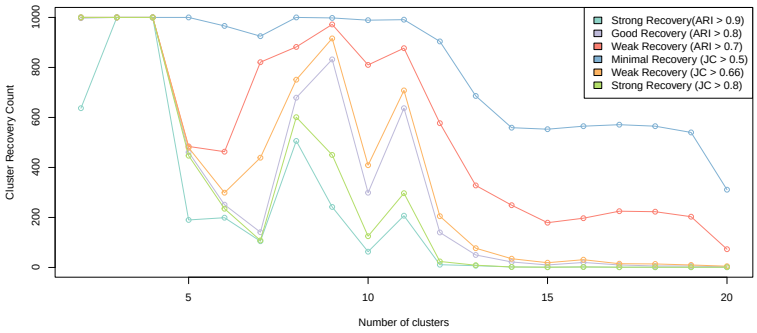


Figure 8: Number of Iterations With a Clustering Similar to the Best One According to Various Ajusted Rand Index (ARI) and Jaccard Coefficient (JC) Thresholds.