Universidade Federal de São Carlos

DComp – Departamento de Computação de Sorocaba

Relatório Técnico
DComp-TR-002/2016

Multiple solutions in cluster analysis: partitions x clusters

Katti Faceli

Tiemi C. Sakata

Sorocaba-SP
Maio/2016

Technical report: DComp-TR-002/2016

Title: Multiple solutions in cluster analysis: partitions x clusters

Authors: Katti Faceli and Tiemi C. Sakata
Contact: katti@ufscar.br and tiemi@ufscar.br
Institution: DComp/CCGT/UFSCar, Sorocaba - SP - Brazil

Funding:

This work was funded by São Paulo Research Foundation (FAPESP). The opinions, hypothesis, conclusions or recommendations contained in this material are the responsibility of the author(s) and do not necessarily reflect FAPESP's vision.

Abstract:

Techniques for obtaining multiple clustering solutions are essential to knowledge extraction in many fields, as they may offer different interpretations of the data. Some of the challenges in discovering multiple clustering solutions refers to obtaining solutions which contain highly differing and high quality clusters, flexibility of approaches regarding different cluster's definitions and the need for benchmark data and a quality assessment methodology. In this paper, we tackle these challenges by (i) providing a methodology for evaluating a set of multiple clustering solutions with respect to domain knowledge available, considering the clusters themselves as solutions instead of the usual partitions; (ii) providing a benchmark for the proposed assessment methodology; (iii) investigating the suitability of techniques based on traditional clustering algorithms for the discovery of high quality clusters considering flexibility regarding different cluster's definitions; and (iv) proposing a simple mechanism for extracting relevant clusters given a collection of partitions. Moreover, besides being used for clustering evaluation, we point out that the clusters themselves could be viewed as the multiple clustering solutions, with great benefits in several aspects of cluster analysis.

keywords: Cluster analysis, Multiple clustering solutions, Cluster evaluation, Alternative clustering

1 Introduction

A great number of applications of cluster analysis can be found today in both academic and commercial areas. Solutions range from the application of traditional clustering algorithms to advanced approaches, which encompasses ensembles and a variety of techniques for obtaining multiple alternative clusterings (Müller et al, 2012). In the scientific research, the main goal of cluster analysis is the extraction of new knowledge from experimental data, giving important insights to advances in knowledge on the field. In commerce/business area, the use of such techniques for data mining may also collaborate significantly to the development of enterprises. In summary, the potential application of clustering techniques is quite broad.

Either on science or business, clustering poses challenges to the data analysts. Clustering techniques are means to explore and verify structures present in the data, by grouping the objects according to some sort of similarity (Jain and Dubes, 1988; Handl et al, 2005; Xu and Wunsch, 2005). The idea is to reveal hidden intrinsic structures with great potential of practical utility for the domain experts.

The majority of the existing clustering approaches aims to find one partition describing the intrinsic structure of a data set. These are the cases, for example, of many traditional algorithms like k-means as well as of many other advanced approaches, such as the clustering ensembles (Strehl and Ghosh, 2002; Monti et al, 2003; Fern and Brodley, 2004; Topchy et al, 2005; Kuncheva et al, 2006; Vega-Pons and Ruiz-Shulcloper, 2011), many evolutionary algorithms for clustering as those described in (Hruschka et al, 2009), and the multiobjective algorithm of Law (Law et al, 2004). Other traditional algorithms find a hierarchy of nested partitions, such as the single-link and complete-link algorithms (Xu and Wunsch, 2005; Jain, 2010). There are also algorithms that search for overlapping clusters, which relax the condition of mutual disjunction in a clustering to obtain soft or fuzzy partitions (Hruschka et al, 2009; Parvin and Minaei-Bidgoli, 2015).

Regardless of the type of structure a clustering algorithm looks for, cluster analysis involves some well-known difficulties derived from the lack of a precise and unique definition of what a cluster is and the nature of data that can present a heterogeneous structure and/or can hide more than one possible structure (Estivill-Castro, 2002; Faceli et al, 2008). These difficulties motivated the development of new techniques that aimed the discovery of a number of alternative heterogeneous structures given a data set. Such availability of alternative structures is important for the domain experts, which are applying cluster analysis for knowledge extraction, as they may offer different interpretations of the data (Handl and Knowles, 2004; Faceli et al, 2008; Müller et al, 2012). As far as we know, most of the approaches dealing with multiple alternative structures considers partition as the structure of interest.

Müller et al. discuss the challenges related to discover multiple clustering solutions (Müller et al, 2012). Among these challenges, we can mention (i) "to provide a processing scheme, which computes multiple clustering solutions that contain high quality clusters and are highly differing to each other" (Müller et al, 2012); (ii) the need of general and flexible approaches allowing the discovery of flexible novel solutions regarding the cluster definition; and (iii) the need for benchmark data and a quality assessment methodology for evaluating multiple clustering solutions. They also highlight the lack of more general techniques tackling several challenges at once.

The traditional clustering algorithms in general consider an homogeneous clustering criterion over the entire feature space and, thus, all clusters recovered will be of the same type (similar in shape or density, for example) (Law et al, 2004; Jiamthapthaksin et al, 2009). On the other hand, these algorithms are very effective in finding the type of clusters they are designed for (Handl and Knowles, 2007). This is evidenced by the large amount of applications that successfully employ such algorithms. In this way, different clustering algorithms, based on different clustering criteria, can be used to build a collection of partitions and provide multiple diverse alternative solutions.

A collection of partitions obtained like this is highly prone to present irrelevant and redundant solutions. There are several more recent clustering approaches that rely on traditional algorithms. They produce a collection of solutions using traditional algorithms and then work on these solutions to build up a more general, concise, and robust result. Many of these techniques are based on the simultaneous optimization of several clustering criteria. MOCK (Multi-Objective Clustering with automatic K-determination) (Handl and Knowles, 2007), MOCLE (Multi-Objective Clustering Ensemble) (Faceli et al, 2009), and IMOCLE (Liu et al, 2012) are examples of such techniques. This type of approach considers partitions produced with traditional algorithms as starting points and an optimization phase to select and explore new solutions. Other techniques like ASA (Automatic Selection Algorithm), aim to select a relevant and diverse subset of solutions, given an initial collection of partitions (Sakata et al, 2010). Traditional algorithms can be used to build this initial collection of partitions.

Redundancy in a collection of partitions can be seen as the presence of a number of solutions that are very similar or even identical. Techniques for finding multiple clustering alternatives always present some mechanism to avoid such redundant solutions. However, we argue that there is another type of redundancy hidden inside partitions. This means that a highly diverse set of partitions can present a great amount of identical clusters. Moreover, we also argue that by relying on the quality of a partition, one can underestimate the quality of the clusters inside it. This is particularly relevant in the multiple solutions context, in which each partition can contain part of the meaningful clusters.

Finally, a partition can be seen as evident and possibly relevant when it is obtained by different algorithms at the same time. Such characteristic was successfully used in ASA for the partitions' selection (Faceli et al, 2010; Sakata et al, 2010). ASA considers the partitions obtained simultaneously by algorithms based on distinct clustering criteria as the most evident ones, and guarantees they are present in the solution set.

Considering all these context, we tackle the mentioned challenges on the discovery of multiple clustering solutions by (i) providing a methodology for evaluating a set of multiple clustering solutions with respect to the domain knowledge available, by considering the clusters themselves as solutions instead of the usual partitions; (ii) investigating the suitability of techniques based on traditional clustering algorithms to discover high quality clusters considering flexibility of cluster's definitions; (iii) proposing a simple mechanism for extracting relevant clusters given a collection of partitions and (iv) providing a benchmark for the proposed assessment methodology. Moreover, besides being used for clustering evaluation, we claim the clusters could be regarded as the multiple clustering solutions themselves. Specifically, the contributions can be detailed as follows:

- We assess the employment of traditional clustering algorithms as a simple strategy for finding multiple alternative partitions by employing several traditional algorithms based on different clustering criteria to build a collection of partitions. Then, we evaluate the quality and diversity of all solutions in this collection, to verify if algorithms that find homogeneous partitions can be used together to provide the mentioned flexibility regarding cluster's definitions.

- We compare a multiobjective clustering and a selection strategies concerning their ability to recover high quality multiple solutions while avoiding redundant and irrelevant solutions.

- We analyze partitions sets in a new fashion, which encompasses exploring the contents of partitions. More specifically, we analyze the alternative partitions collections by breaking them into their clusters components and then producing collections of clusters. We investigate redundancy and quality issues as well as the amount of irrelevant information produced. With this, we prove our arguments that a great amount of redundancy can exist even in a diverse set of partitions and that the quality of a whole set of clusters in a given collection of partitions is underestimated when the evaluation is done solely considering the rigid structures that are the partitions.

- We investigate the usefulness of a cluster evidence in the selection of relevant and high quality clusters. The evidence of a cluster is given by the number of times a cluster appears inside different partitions of a collection. We show that this approach can be as, or more effective than, some advanced clustering techniques, if we take into account the quality of clusters recovered, the number of irrelevant clusters selected, and the simplicity of the approach (low computational cost).

- By comparing the analysis based on partitions and clusters and using the evidence of a cluster to select high quality alternatives, we illustrate the benefits of considering clusters as multiple alternative solutions instead of multiple partitions.

- We organize all data and results, making them available as a benchmark for the type of analysis we propose, namely the Clusters Evaluation Benchmark[1]. The available information encompasses pre-processed data sets, known structures, collections of partitions, collections of clusters, worksheets with the quality of solutions (partitions and clusters), according to the evaluation indices employed.

As our aim is on the way of analyzing and not comparing the techniques themselves, we choose a representative of each technique we have been working with. To achieve all the mentioned goals, this paper is structured as follows. In Section 2, we introduce related studies that had motivated and/or that will be used in our analysis along with the notation we used in our study. A detailed description of the performed experiments is presented in Section 3. Section 4 contains all the analysis performed and a discussion on their implications. Finally, in Section 5, we present a summary of our conclusions.

2 Background and related studies

Many recent and advanced clustering approaches rely on traditional algorithms and/or aim to find multiple alternative solutions. In this section, we briefly describe the approaches used in this paper and/or those that motivated the ideas presented here. We also introduce the terminology and notation we used along the paper:

- $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ is a data set with n objects.

- $c_k \subset X$ and $c_k \neq \emptyset$ is a cluster of X.

- $C = \{(c_i, n_i) | c_i \in C^u, n_i \in \mathbb{Z}^+\}$ is a multiset of clusters, where C^u is the underlying set of C, with nc clusters, and n_i is the multiplicity of the cluster c_i.

- A partition of X in K clusters is a set of clusters $\pi = \{c_1, c_2, ..., c_K\}$, such that $\bigcup_{j=1}^{K} c_j = X$ and $c_j \cap c_l = \emptyset, j, l = 1, ..., K$ and $j \neq l$.

- $\Pi = \{(\pi^i, n_i) | \pi^i \in \Pi^u, n_i \in \mathbb{Z}^+\}$ is a multiset of partitions, where Π^u is the underlying set of Π, and n_i is the multiplicity of the partition π^i.

- The cardinality of a set or a multiset A is denoted by $|A|$.

[1] Clusters Evaluation Benchmark, accessed in 12/10/2015 and available at `http://lasid.sor.ufscar.br/clustersEvaluationBenchmark/`

Henceforth, we will use the term *collection* to denote both sets and multisets.

As previously mentioned, traditional clustering algorithms search for homogeneous partitions. In such type of partition, all clusters are in accordance to the same definition. Thus, one traditional algorithm alone can not contribute with the challenge of producing novel solutions which are flexible regarding cluster definitions. Nevertheless, many advanced approaches have been using these algorithms attempting to produce more general results such as heterogeneous partitions (each cluster is in accordance to a different cluster definition) or partitions sets where heterogeneity are distributed among partitions (each partition can be in accordance to a different criterion). In this paper, we will employ the traditional algorithms Average-link (AL), Centroid-link (CeL), Complete-link (CoL), k-means (KM), Single-link (SL), and Shared Nearest Neighbors (SNN) as a basis for producing the mentioned flexibility.

The first techniques that use traditional algorithms to provide diversity of criteria were the heterogeneous ensembles. In such ensembles, a collection of base partitions Π is produced by running different clustering algorithms. This strategy can be applied alone or combined with other strategies. Then, a function is applied to Π to combine partitions into a single consensus partition. There is a great variety of consensus functions resulting in different ensembles (Vega-Pons and Ruiz-Shulcloper, 2011; Topchy et al, 2005; Kuncheva et al, 2006; Iam-On and Boongoen, 2015). Some examples of heterogeneous ensembles can be seen at (Strehl and Ghosh, 2002; Ayad and Kamel, 2003; Monti et al, 2003; Hu and Yoo, 2004; Fred and Jain, 2006; Gionis et al, 2007; Domeniconi and Al-Razgan, 2009; Chung and Dai, 2014). Albeit such ensembles employ different clustering criteria in the construction of base partitions, all these criteria are considered simultaneously in the construction of the consensus partition. In this way, high quality clusters with regard to one criterion can be diluted by weak clusters when combined. This may lead to an overall poor quality of the consensus partition, although good clusters may be present in base partitions (Law et al, 2004; Piantoni et al, 2015). Moreover, traditional ensembles are not intended to the obtaining of multiple alternative clusterings.

The multiobjective approach of Law (MOL) (Law et al, 2004) is one of the first works on ensembles addressing the problem of finding a heterogeneous partition based in the ability of the traditional clustering algorithms and applying different clustering criteria for different regions of data space. MOL produces a collection of candidate clusters C by running different clustering algorithms. Each algorithm produces a partition π^a of X in K^a clusters, and $C = \cup_a \pi^a$. Then, it applies hill climbing to find the set of target clusters that will compose the final partition. The optimized objective function is composed of a goodness function based on cluster stability (NMI- Normalized Mutual Information) and penalties for dealing with overlapping clusters and objects that remain unassigned. Essentially, this algorithm finds a partition with each cluster that may be according to a different cluster definition, despite failing when the best candidate clusters significantly overlap. As MOL enforces that the result to be one single partition, it is not appropriate for finding multiple alternative clusterings. In the meantime, several ideas employed in MOL have a great potential to be applied in the multiple alternative clusterings scenario.

There is a number of approaches that address the issue of considering multiple clustering criteria and also result in a set of alternative clusterings. Many of these techniques rely on the simultaneous optimization of several clustering criteria. In this paper, we are interested on approaches based on the combined use of traditional single-objective clustering algorithms and MultiObjective Evolutionary Algorithms (MOEAs) to find multiple solutions, as our aim is to investigate the suitability of traditional algorithms as a basis for such task.

The MOEAs are based on the concept of Pareto optimality and, in the context of clustering, will produce a set of partitions Π_S, which represents an approximation of the Pareto optimal set (for short, referred to as Pareto set from now on). The objective functions to be optimized should represent validation indexes able to measure the quality of partitions in different ways, each one being related to a different clustering criterion. For example, the algorithms MOCK (Multi-Objective Clustering with automatic K-determination) (Handl and Knowles, 2007) and MOCLE (Multi-Objective Clustering Ensemble) (Faceli et al, 2009) employ a Pareto-based multiobjective genetic algorithm to simultaneously optimize several clustering criteria. Other similar approaches can be seen in (Saha and Bandyopadhyay, 2010; Kraus et al, 2011; Coelho et al, 2011; Liu et al, 2012; Wahid et al, 2014). We employ MOCLE in this paper, which has shown to be able to find a diverse set of partition, while keeping this set somehow concise (Faceli et al, 2009).

More than an algorithm, MOCLE is a framework used to build multiobjective clustering ensembles (Faceli et al, 2009). It employs several of the same basic ideas of MOCK together with ideas of clustering ensembles. Its general idea is to build a collection of base partitions Π_I by considering different clustering criteria and to optimize this set of solutions with multiple criteria to produce a set of consensus partitions Π_C. Any set of algorithms can be used to produce Π_I, but for better performance, algorithms based on different criteria are recommended. For the optimization, a Pareto-based multiobjective genetic algorithm should be employed with the following components: (i) Π_I as initial population; (ii) an ensemble algorithm as crossover operator; (iii) two or more complementary validation indices as objective functions. No mutation is used. For the crossover, we could use any existing cluster ensemble method suitable for partitions pairs. Several alternatives were investigated for different components.

With ideas similar to MOL, MOC (MultiObjective Clustering) is a framework for multiobjective clustering that aims to recover interesting clusters regarding two or more objectives (criteria) (Jiamthapthaksin et al, 2009). Like the MOEAs mentioned, MOC is based on the concept of Pareto optimality. However, it differs from most multiobjective approaches as “it seeks for good individual clusters maximizing multiple objectives that are integrated into a single clustering by a user-driven post-processing step” (Jiamthapthaksin et al, 2009). Briefly, MOC produces a repository of potentially interesting clusters according to multiple objectives and have a cluster summarization unit that allows the selection of subsets of these clusters, according to user preferences. The repository of clusters is build by running clustering algorithms that support plug-in fitness functions and selecting the best clusters according to the Pareto dominance. This repository will contain only clusters which are good with respect to at least two objectives. The summarization step allows users to query the repository of clusters on different objectives and thresholds. The result is a “final clustering from the viewpoint of a single or a small set of objectives that are of particular interest for a user”(Jiamthapthaksin et al, 2009). For this, they propose an algorithm named MO-Dominance-guided Cluster Reduction algorithm, which selects the clusters considering objectives and thresholds given by the user.

An important issue to consider when searching for multiple solutions is that alternatives must be highly different so that each alternative will be able to provide additional knowledge (Müller et al, 2012, 2015). On the other hand, when a solution can be found by different means (e.g different algorithms based on different criteria), this is a clear sign of evidence and probably of relevance (Faceli et al, 2010; Sakata et al, 2010). The ASA is an algorithm for the selection of partitions, which considers both evidence of partitions (given its easy identification by several different criteria), and diversity of collections of partitions (Faceli et al, 2010).

Given a collection of initial partitions (Π_I), the ASA produces a set of partitions Π_S, working as follows. First, it initializes Π_S with the most evident partitions, that is, partitions $\pi^i \in \Pi_I$ having $n_i > p$, where p is a parameter of the algorithm. In (Faceli et al, 2010), p is the number of algorithms employed to build Π_I. Then, ASA starts an iterative process of (i) discarding partitions of Π_I which are highly similar to all those already selected, and (ii) adding new other distinct partitions to Π_S. Similarity is given by the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). The level of similarity considered for discarding a partition is given by a threshold automatically adjusted by the algorithm. Any clustering strategy could be used to produce Π_I. Nevertheless, to take advantage of the selection of evident partitions aspect, it is necessary to employ different algorithms to produce Π_I.

The success obtained by the techniques based on traditional algorithms motivated us to investigate them as basis for a simple mechanism of producing multiple solutions according to diverse definitions of clusters. The ideas of MOL and MOC also motivated us to investigate the quality of clusters inside a collection of partitions produced with traditional clustering algorithms and other mentioned approaches. In this way, we look at the results of algorithms as if they were a repository of clusters and then, we analyze the extension they encompass the true clusters hidden among several underlying structures of a given data set.

As the multiplicity of a solution in a given collection of partitions has been successfully used in ASA, we also investigated the potential of the same information in the context of clusters. For such, we will consider the selection of clusters with a multiplicity higher than two as an approach for producing a collection of clusters instead of partitions. From here on, we will refer to such approach as Multiplicity Based Cluster Selection (MBCS).

3 Experiments

In this section, we describe the experimental design employed in our investigation. More specifically, we detail the data sets, the procedure to build the collections of partitions and clusters, and finally, the procedure for evaluating the quality of these collections.

3.1 Data sets

In this analysis, we used a total of 37 data sets with differences in size (number of objects and dimensionality), shape, definitions of cluster, and domain area. Among them, 15 are artificially produced, representing different properties of interest, and 17 are real data from several domains like Medicine and Bioinformatics. Moreover, 15 of these artificial and real data sets present multiple alternative true partitions. These data sets were obtained from several sources and will be briefly described in the following.

Each of these data sets have a set of np^{TP} different true partitions (or known structures) $\Pi_{TP} = \{\pi^1, \pi^2, ..., \pi^{np^{TP}}\}$. As we are interested in evaluating the amount of information discovered using clustering strategies, we considered the recovery of clusters individually. For this, we break Π_{TP} into their clusters components thus producing a multiset of clusters $C_{TP} = \bigcup_{\pi^{tp} \in \Pi_{TP}} \pi^{tp}$.

Table 1 summarizes the main characteristics of these data sets. In this table, n is the number of objects, d is dimensionality (number of attributes), np^{TP} is the number of true partitions, $K^{\pi^j \in \Pi_{TP}}$ is the amount of clusters of each $\pi^j \in \Pi_{TP}$, and nc^{TP} is the number of distinct clusters in C_{TP}. Considering all data sets, we have a total of 62 true partitions and 251 true clusters.

Table 1: Data sets characteristics

Type	**Data set**	n	d	np^{TP}	$K^{\pi^j \in \Pi_{TP}}$	nc^{TP}
Artificial	`atom`	800	3	1	2	2
	`ds2c2sc13`	588	2	3	2, 5, 13	19
	`ds3c3sc6`	905	2	2	3, 6	8
	`ds4c2sc8`	485	2	2	2, 8	10
	`engyTime`	4096	2	1	2	2
	`gaussian`	60	600	1	3	3
	`hepta`	212	3	1	7	7
	`lsun`	400	2	1	3	3
	`monkey`	4000	2	4	8,5,3,2	14
	`simulated6`	60	600	1	6	6
	`spiralsquare`	2000	2	2	2, 6	8
	`target`	770	2	1	6	6
	`tetra`	400	3	1	4	4
	`twoDiamonds`	800	2	1	2	2
	`wingNut`	1016	2	1	2	2
Real	`armstrong`	72	1081	2	2,3	4
	`chowdary`	104	182	1	2	2
	`contractions`	98	27	1	2	2
	`dyrskjot`	40	1203	1	3	3
	`eTongueSugar`	375	6	2	2,3	5
	`glass`	214	9	3	2, 5, 6	9
	`golub`	72	3571	4	2, 3, 2, 4	10
	`gordon`	181	1626	1	2	2
	`iris`	150	4	1	3	3
	`laryngeal1`	213	16	1	2	2
	`laryngeal2`	692	16	1	2	2
	`laryngeal3`	353	16	2	2,3	4
	`libras`	360	90	2	8,15	21
	`lung`	197	1000	1	4	4
	`miRNAcancer`	218	217	6	3, 20, 4, 9, 2, 2	40
	`respiratory`	85	17	1	2	2
	`segmentation`	2310	19	1	7	7
	`su`	174	1571	1	10	10
	`voice3`	238	10	2	2,3	4
	`voice9`	428	10	2	2,9	10
	`weaning`	302	17	1	2	2
	`yeoh`	248	2526	2	2, 6	7

Artificial Data Sets

Data sets atom, engyTime, hepta, lsun, target, tetra, twoDiamonds and wingNut belong to the Fundamental Clustering Problems Suite[2] (FCPS), which is an elementary benchmark for clustering algorithms (Ultsch, 2005). These data sets were designed to reveal benefits and shortcomings of traditional algorithms, including their adequacy for finding diverse types of clusters. For this, FCSP presents data sets with a good diversity of cluster's definitions. Each algorithm can successfully identify the structure of some of these data sets, while failing with others.

Data sets gaussian and simulated6 contain high dimensional data simulating gene expression data. They are available as supplementary material[3] of (Monti et al, 2003).

Data sets ds2c2sc13, ds3c3sc6, ds4c2sc8, spiralsquare and monkey were designed by the authors to explore (i) the diversity of types of clusters in heterogeneous structures and (ii) the availability of multiple solutions. For such, each of them contain at least two structures representing different refinement levels of the same information. Moreover, at least one of the structures of each data set is heterogeneous. Data sets ds2c2sc13, ds3c3sc6, ds4c2sc8 and spiralsquare were previously described in (Faceli et al, 2010). The spiralsquare data set was constructed from two data sets described in (Handl and Knowles, 2004). Data set monkey is used in this paper for the first time. Figure 1 illustrates the data set monkey with its known structures. These data sets are available at the website of the Laboratory of Intelligent and Distributed Systems of the Department of Computing, UFSCar, as part of the Clusters Evaluation Benchmark.

The remainder data sets contain real data. In these, different structures correspond to different known classifications of data. Thus, we assume the known classifications follow some of the used clustering criteria. However, a classification could be unrelated to a clustering criterion. This would lead to a low performance for all clustering techniques.

Real Data Sets

Data sets contractions, laryngeal1, laryngeal2, laryngeal3, respiratory, voice3, voice9 and weaning regard to data from medical domain and were made available by the Pattern Recognition group of School of Computer Science, Bangor University[4]. Details on specific publications concerning these data sets are available together with the data.

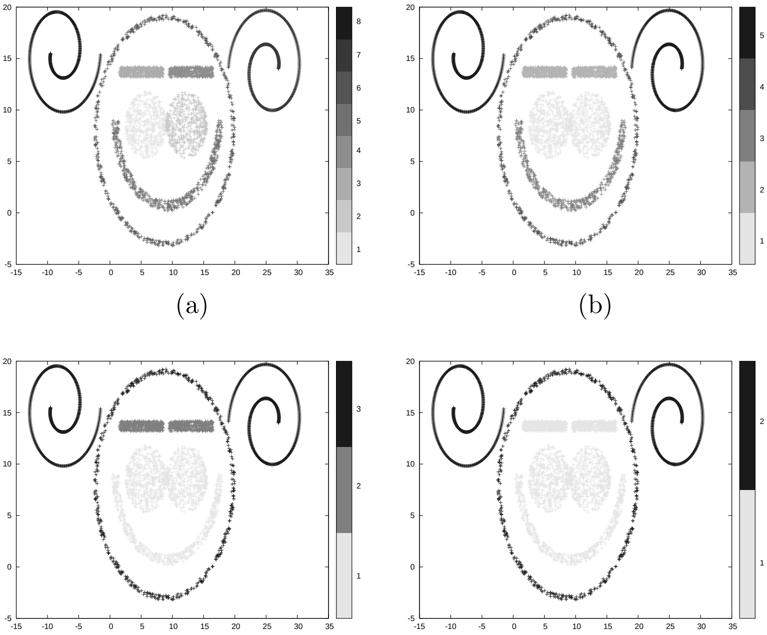
Data sets glass, iris, libras and segmentation were obtained from the UCI Machine Learning Repository[5] (Newman et al, 1998).

[2] Fundamental Clustering Problems Suite, accessed in 04/14/2015 and available at http://www.uni-marburg.de/fb12/datenbionik/data?language_sync=1

[3] Supplementary material of (Monti et al, 2003), accessed in 04/14/2015 and available at http://www.broadinstitute.org/cgi-bin/cancer/publications/view/87

[4] Real medical data sets, accessed in 04/14/2015 and available at http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm

[5] UCI Machine Learning Repository, accessed in 04/14/2015 and available at http://archive.ics.uci.edu/ml/



(a) (b)

(c) (d)

Figure 1: True structures of `monkey` data set

Data sets `armstrong`, `chowdary`, `dyrskjot`, `golub`, `gordon`, `lung`, `miRNAcancer`, `su` and `yeoh` are from bioinformatics domain and were originally described in (Golub et al, 1999; Bhattacharjee et al, 2001; Armstrong et al, 2002; Su et al, 2001; Yeoh et al, 2002; Gordon et al, 2002; Dyrskjøt et al, 2003; Lu et al, 2005; Chowdary et al, 2006). Here, we used the same version of the data sets we employed in (Faceli et al, 2010).

Data set `eTongueSugar` was built as a combination of the data sets of the E-Tongue Sugar Collections v.1[6], described in (Sakata et al, 2012). The `eTongueSugar` data set refers to sugar quality assessment. The attributes refers to measurements automatically collected using an electronic tongue sensor. The original data sets contained the same type of data, but with different types of sample preparation. For one data set, the pH of the samples was controlled and for the other it wasn't. In the `eTongueSugar` data set, we mixed samples with and without pH control. In this way, the data set `eTongueSugar` contains two alternative structures, one distinguishing the samples with and without pH control, and another concerning sugar type: Organic, VHP (Very High Polarization) or VVHP (Very Very High Polarization). This version of the data set is available only as part of the Clusters Evaluation Benchmark.

3.2 Partitions and Clusters Obtaining

For the analysis we made, we have produced three sets of partitions: Π_{BAlg}, produced with Basic Algorithms; Π_{MOCLE}, produced with MOCLE; and Π_{ASA}, produced with ASA.

[6] E-Tongue Sugar Collections v.1, accessed in 12/10/2015 and available at `http://www.dcomp.sor.ufscar.br/talmeida/etonguesugar/index.html`

The more diverse algorithms are employed, the higher the chances of producing a more diverse set of partitions and clusters. To explore diverse numbers of clusters is fundamental. For data sets hiding several alternative structures, it is natural for these alternatives to present different numbers of clusters. Moreover, even considering data sets with one single true structure, it is known that the best clustering obtained with an algorithm does not always contain the same number of clusters of the true structure. By considering this, to produce Π_{BAlg}, we run conceptually different clustering algorithms with several parameters settings. The algorithms employed were AL, CeL, CoL, KM, SL, and SNN. Generally speaking, AL, CeL, CoL, and KM look for compact clusters while SL and SNN produce connected clusters. AL, CeL, CoL, KM, and SL are traditional and largely employed clustering algorithms (Jain and Dubes, 1988). SNN is a more recent technique that robustly deal with high dimensionality, noise, and outliers (Ertöz et al, 2002).

To allow flexibility and increase the amount of information that can be extracted, we have varied the parameters for each algorithm to produce partitions with the number of clusters $K \in [K^{min}, K^{max}]$, where $K^{min} = 2$ and $K^{max} = 2 \max_{\pi^j \in \Pi_{TP}} K^{\pi^j}$.

For hierarchical algorithms (AL, CeL, CoL, and SL), we generated and cut the hierarchies to produce one partition for each value of K. For KM, to minimize the occurrence of suboptimal solutions, we ran the algorithm 30 times for each K with a random choice of initial centroids. Among all 30 partitions produced for a given K, we selected the partition with the lowest squared error for Π_{BAlg}. For SNN, we ran the algorithm with several values for its parameters and then selected the partitions having K in the interval of interest to compose Π_{BAlg}. The parameters values was the same as the ones used in (Faceli et al, 2007), that is NN being of 2%, 5%, 10%, 20%, 30% and 40% of n, *topic* and *merge* of 0, 0.2, 0.4, 0.6, 0.8 and 1, the default value for the parameter *strong*, and the value 0 for the parameters *noise* and *label*.

We employed the software Cluster 3.0[7](Hoon et al, 2004) to run the algorithms AL, CeL, CoL, KM, and SL. To run SNN, we used the implementation of its authors, sent to us upon our request.

To produce Π_{MOCLE}, we ran the version of MOCLE[8] implemented with NSGA-II and with MCLA as the crossover operator. We also used the CON and DEV validation indices as the objective functions and the set of partitions Π_{BAlg} as the initial population. The current version of MOCLE allows the variation of two parameters: G, which is the number of generations the genetic algorithm will run and L, a percentage on the data set size, based in which the number of nearest neighbors necessary for calculating the connectivity will be determined. After a few empirical tests with different values for parameters L and G, we decided to employ the values 2.5% and 100, respectively.

Finally, for producing Π_{ASA}, we ran the algorithm ASA[9] by feeding it with Π_{BAlg}. For the parameter p, we used value 2.

To evaluate the obtaining of clusters, we broke the sets of partitions and produced the corresponding sets of clusters, as we did with the true sets of partitions. That is, for each Π_S (Π_{BAlg}, Π_{MOCLE} or Π_{ASA}) we produced their corresponding set of clusters

[7] Cluster 3.0, accessed in 12/10/2015 and available at
`http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm`

[8] MOCLE, accessed in 12/10/2015 and available at
`http://lasid.sor.ufscar.br/mocleproject/`

[9] ASA, accessed in 12/10/2015 and available at
`http://lasid.sor.ufscar.br/asaproject/`

$C_S = \bigcup_{\pi^s \in \Pi_s} \pi^s$.

Moreover, to analyze the selection of clusters as an alternative to some advanced clustering techniques, we produced a set of clusters (C_{MBCS}), by applying the MBCS (Multiplicity Based Cluster Selection) directly in the collection of clusters C_{BAlg}. In this way, the C_{MBCS} contains all the distinct clusters in C_{BAlg} that presented a multiplicity equal or greater than 2 ($n_i \geq 2$).

3.3 Methodology for Evaluating the Quality of Results

To evaluate the quality of partitions sets, we used the Adjusted Rand index (ARI), which measures the similarity between two partitions, π^i and π^j (Jain and Dubes, 1988; Hubert and Arabie, 1985; Milligan and Cooper, 1986). The $ARI(\pi^i, \pi^j)$ is given by Equation 1, where (1) $n_{ij} = |c_i \cap c_j|$, $c_i \in \pi^i$ and $c_j \in \pi^j$; (2) $n_{i\cdot}$ indicates the number of objects in the cluster c_i; (3) $n_{\cdot j}$ indicates the number of objects in the cluster c_j; (4) n is the total number of objects; and (5) $\binom{a}{b}$ is the binomial coefficient $\frac{a!}{b!(a-b)!}$. This index results in values from -1 to 1, with the value 1 indicating a perfect agreement between the partitions and values near 0 or negatives corresponding to cluster agreement found by chance.

$$ARI(\pi^i, \pi^j) = \frac{\sum_{i=1}^{|\pi^i|} \sum_{j=1}^{|\pi^j|} \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^{|\pi^i|} \binom{n_{i\cdot}}{2} \sum_{j=1}^{|\pi^j|} \binom{n_{\cdot j}}{2}}{\frac{1}{2}[\sum_{i=1}^{|\pi^i|} \binom{n_{i\cdot}}{2} + \sum_{j=1}^{|\pi^j|} \binom{n_{\cdot j}}{2}] - \binom{n}{2}^{-1} \sum_{i=1}^{|\pi^i|} \binom{n_{i\cdot}}{2} \sum_{j=1}^{|\pi^j|} \binom{n_{\cdot j}}{2}} \tag{1}$$

Considering each data set X, the corresponding sets of true partitions, Π_{TP}, and a collection of partitions Π_S produced by Π_{BAlg}, Π_{MOCLE} or Π_{ASA}, we calculate the ARI between each pair of partitions π^{tp} and π^s, where $\pi^{tp} \in \Pi_{TP}$ and $\pi^s \in \Pi_S$. Then, for each $\pi^{tp} \in \Pi_{TP}$, we selected the partition π^s with the largest ARI.

Similarly, for evaluating the quality of the individual clusters, we considered the set of true clusters C_{TP} and a collection of clusters C_S produced by C_{BAlg}, C_{MOCLE} or C_{ASA}. In order to evaluate the similarity between two clusters c_i and c_j, we employed the proportion of the objects present in both clusters, given by Equation 2. For each pair of clusters c_{tp} and c_s, where $c_{tp} \in C_{TP}$ and $c_s \in C_S$, we calculate the Intersection Degree of two clusters, $InD(c_{tp}, c_s)$. Then, for each $c_{tp} \in C_{TP}$, we selected the cluster c_s with the largest InD.

$$InD(c_i, c_j) = \frac{|c_i \cap c_j|}{|c_i \cup c_j|} \tag{2}$$

4 Analysis and Discussion

Figure 2 illustrates the total number of partitions produced with each technique for all data sets, comparing to the number of true structures (line in the graphic). The graphic shows the great amount of information produced with the techniques in comparison to the amount of relevant information hidden in data (true partitions). This gives rise to three main questions: (i) How much of the information obtained represent redundant information?; (ii) All the relevant information are indeed recovered?; (iii) How much of the information obtained really represents relevant information?

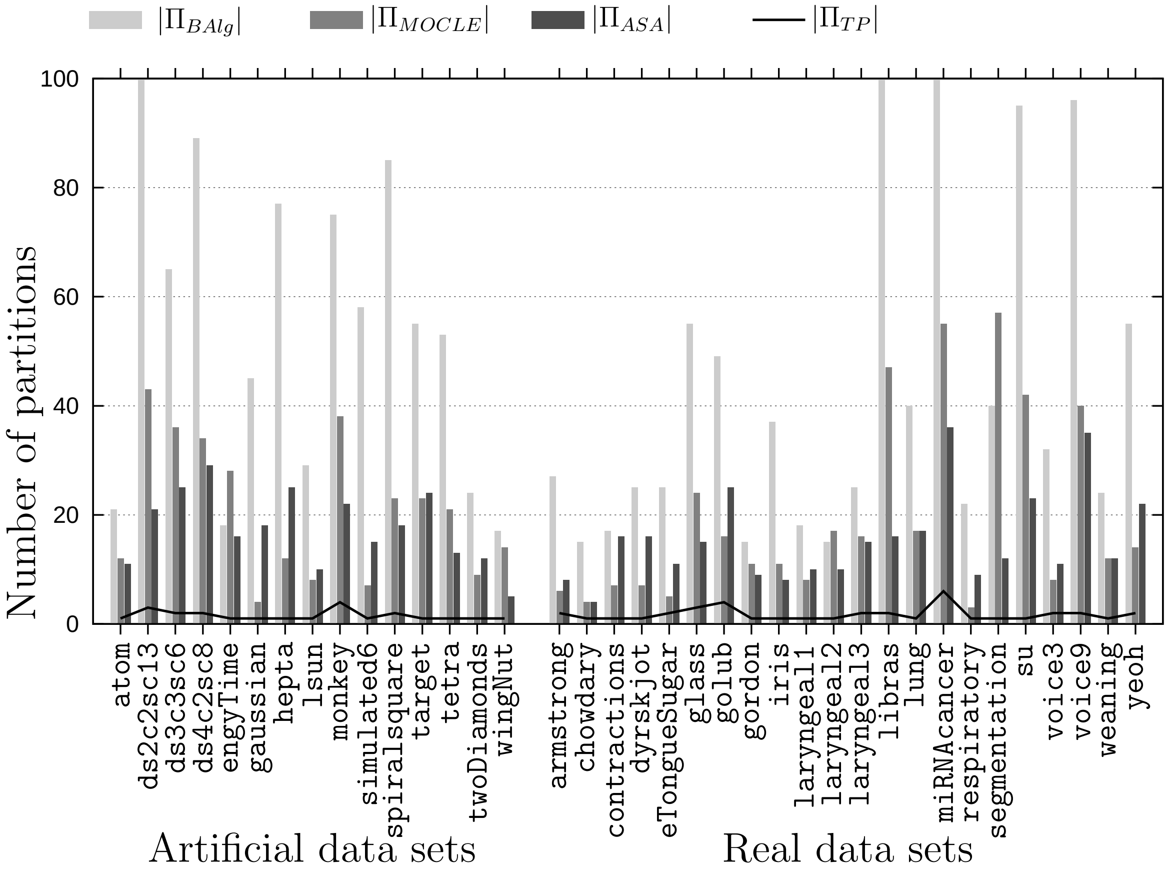


Figure 2: Number of partitions produced by each technique. For data sets `ds2c2sc13`, `libras` and `miRNAcancer`, $|\Pi_{BAlg}| > 100$.

Because of their nature, the solutions of MOCLE and ASA are not redundant. That is, partitions they produce/select are different from each other. On the other hand, different algorithms (or one algorithm with different settings of parameters) can produce identical or very similar partitions. Thus, Π_{BAlg} can present redundant partitions. Figure 3 illustrates the number of partitions in Π_{BAlg} in comparison to the number of distinct partitions (Π^u_{BAlg}). As can be observed, the redundancy in Π_{BAlg} exists, but it is not too high.

If we turn our attention to the clusters inside partitions, we can see that the number of distinct clusters present in partitions is indeed much smaller than the total number of clusters they present. This means the information present inside partitions can be redundant even for true partitions, in cases in which more than one underlying structure exists. To illustrate this issue, we quantify the redundancy of clusters by what we call redundancy degree of a collection of clusters. Given a collection of clusters C_S (C_{BAlg}, C_{MOCLE} or C_{ASA}), redundancy degree is given by $1 - |{C^u}_S|/|C_S|$. The smaller the value, the smaller the number of replicated clusters. Figure 4 presents the redundancy degree of each collection of clusters, including true clusters C_{TP}, represented by a line in the graphic.

Observing Figure 4, we can see that, for almost all cases, all techniques present some degree of redundancy. Even in the collection of true clusters, there are cases of redundancy. For example, this can be due to partitions containing clusters that represent subdivisions of clusters from other partitions. In the case of BAlg, all but two data sets

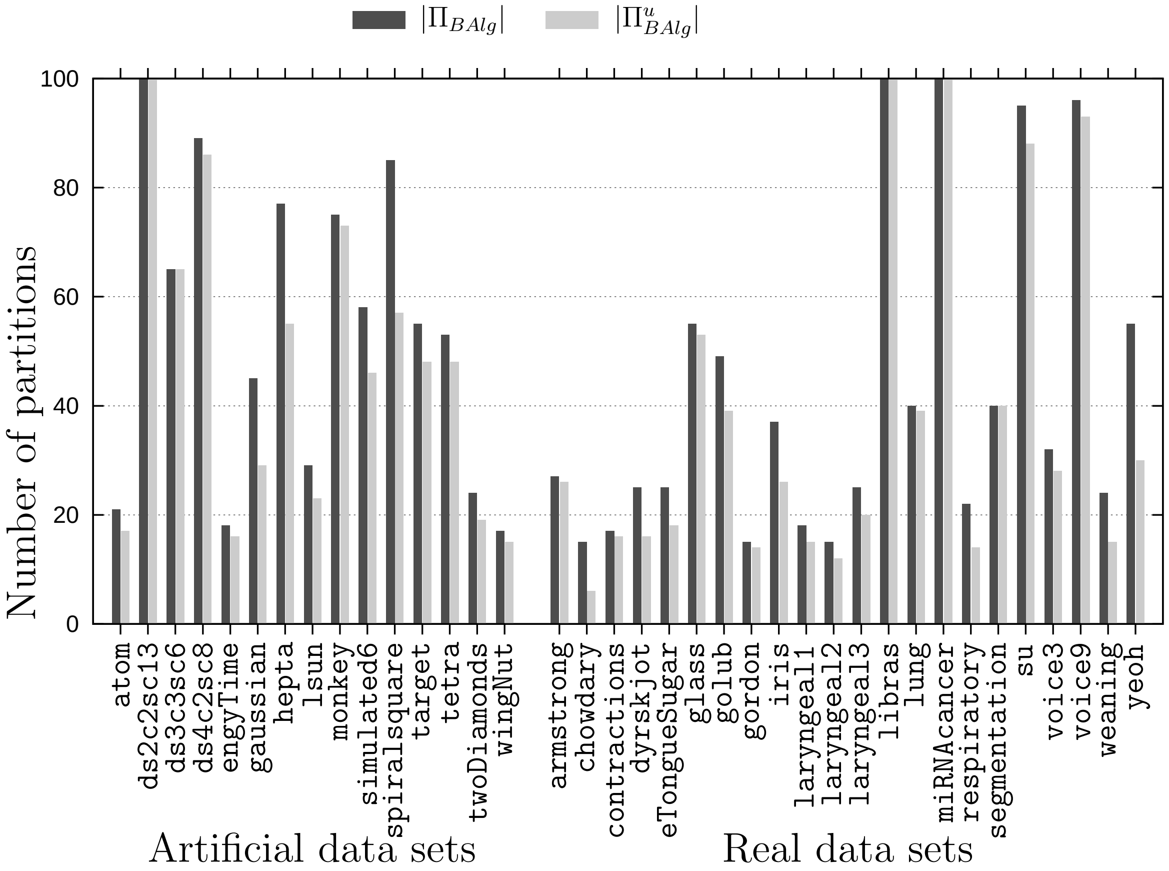


Figure 3: Number of partitions ($|\Pi_{BAlg}|$) and of distinct partitions ($|\Pi^{u}_{BAlg}|$) produced by traditional algorithms.

presented a redundancy degree of at least 0.4, which means that more than 40% of the clusters are repeated. Even in the cases of MOCLE and ASA, where partitions are not redundant, the clusters are. In more than 50% of data sets, more than 50% (redundancy degree of 0.5) of the clusters obtained with MOCLE and more than 30% (redundancy degree of 0.3) of the clusters obtained with ASA are redundant. In some cases, the redundancy achieves almost 90% of the clusters. In average, around 62% of the clusters in BAlg, 49% of the clusters obtained with MOCLE, 33% of the clusters obtained with ASA and 5% of the clusters in the true partitions are redundant.

Turning the attention to the quality of solutions, we presented the percentage of true partitions (Figure 5) and true clusters (Figure 7) recovered with each strategy. In Figure 5, we present the percentage of fully recovered partitions ($ARI = 1$) and a percentage of recovery including partitions retrieved only partially ($ARI > 0.7$), and, in Figure 7, we present the percentage of fully recovered clusters ($InD = 1$) and a percentage of recovery including clusters retrieved only partially ($InD > 0.7$).

Observing Figure 5, we can see that no more than 16% of true partitions was fully recovered by all techniques. If we consider an approximated recovery, no more than 40% of partitions was retrieved. The higher level of recovery was achieved by the traditional algorithms (BAlg). We expected ASA would loose information as it only works by selecting partitions among those in the initial collection. However, we did not expect the lost of information to occur in the case of MOCLE. Oppositely, we expected the amount of information recovered would increase over the initial collection, as MOCLE can produce

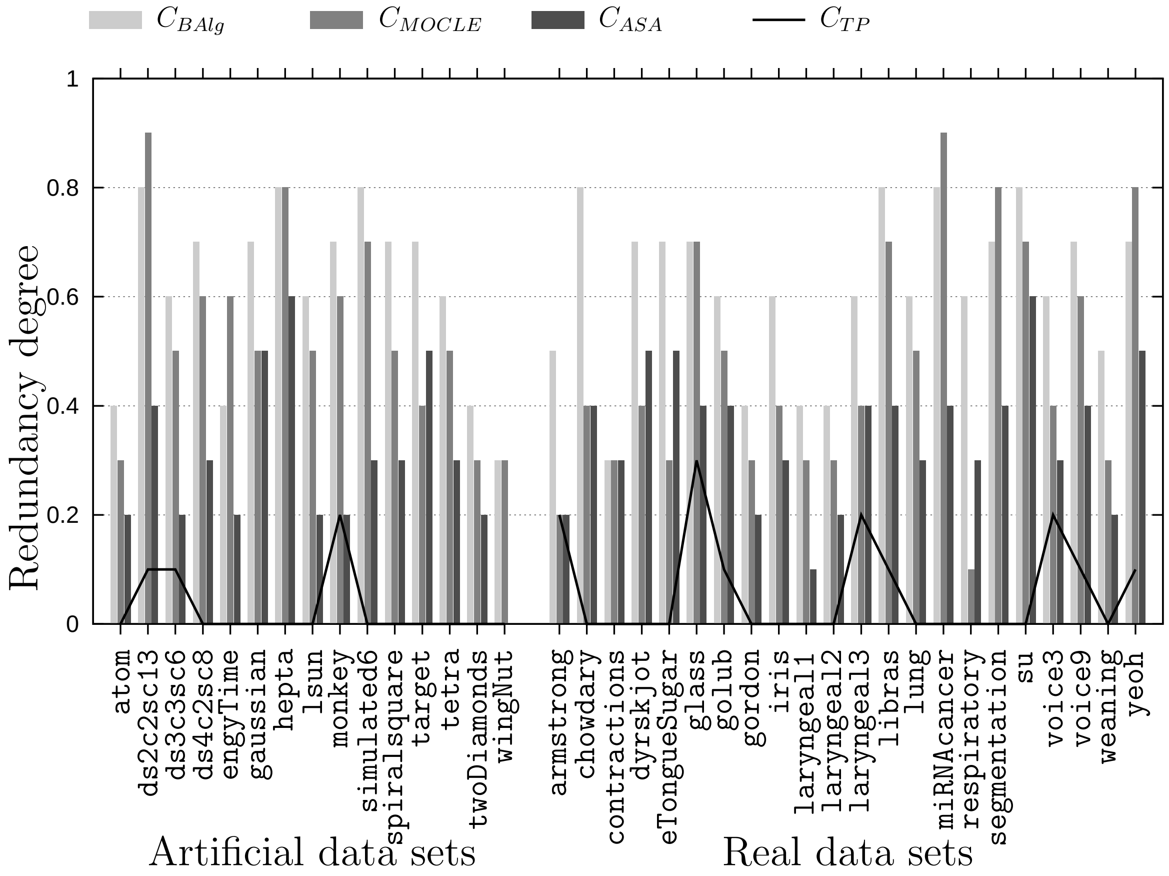


Figure 4: Redundancy degree for each collection of clusters.

new partitions. The loss in MOCLE's case ended up being higher than for ASA, either considering full or approximate partitions.

Figure 6 represents one detail of Figure 5. It depicts the percentage of fully or approximately recovered partitions ($ARI > 0.7$) detailed for each data set. In this figure, it is evident that the recovery of true structures are concentrated in artificial data sets. None of the partitions was integrally recovered ($ARI = 1$) with real data sets (data not shown). And, considering an approximate recovery, in only four out of 22 real data sets, the strategies recovered some of the partitions.

Although all techniques produced a high number of partitions, as observed in Figure 2, most of them are of no interest. That is, they do not represent true partitions hidden in the data, as observed in Figures 5 and 6.

With this analysis, we show that the recovery of whole partitions is hardly achieved, mainly for real data. However, we wanted to check if we could see a better picture if we looked at the clusters hidden in the partitions. For this, we analyzed the collections of clusters produced from collections of partitions provided by each technique (BAlg, MOCLE and ASA). At the same time, we analyze the potential of applying MBCS by comparing C_{MBCS} to the other collections of clusters.

In Figure 7, it is possible to observe that approximately 30% of the clusters was retrieved integrally ($InD = 1$) by all techniques. The worst recovery was achieved with MOCLE, while the best was achieved with BAlg. MBCS (which means the simple selection of clusters that appeared twice in C_{BAlg}) provided the retrieval of the same clusters as those recovered by ASA. Considering a partial recovery ($InD > 0.7$), 50% to 60% of

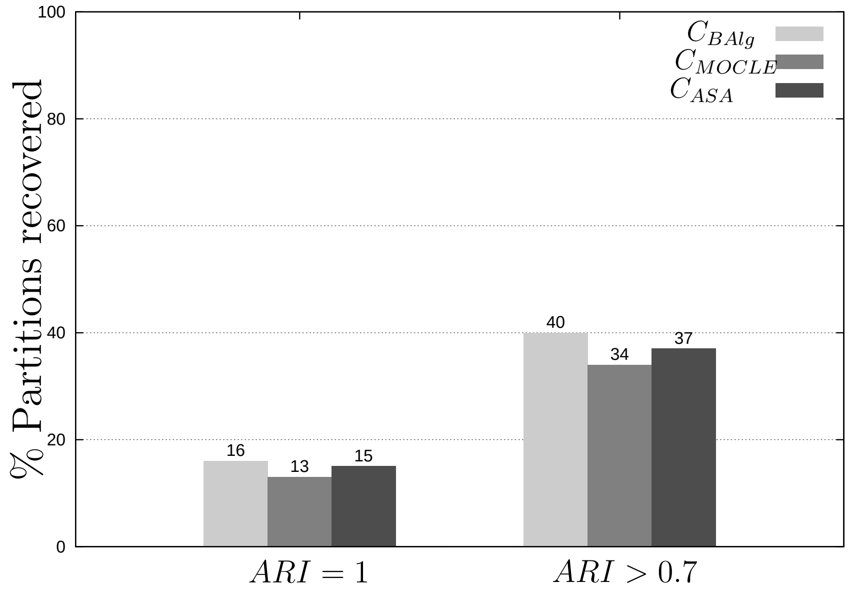


Figure 5: Percentage of true partitions recovered by each technique.

the true clusters were recovered. Again, the best recovery was achieved with BAlg (61% of the true clusters) and the worst was achieved with MOCLE.

Figure 8 represents the detailing of Figure 7, considering the approximate recovery detailed for each data set. As for the scenario of partitions, the recovery of true clusters was better for artificial data sets. All techniques recovered 100% of the clusters for 9 out of 15 artificial data sets. In the other six cases, at least 60% of the clusters was recovered by all techniques (except in two MBCS cases). Nevertheless, for the real data sets, a very different picture can be seen. For eight out of the 22 real data sets, all techniques recovered at least 50% of true clusters (again with 2 exceptions in MBCS) and for 18 out of 22 real data sets, the strategies recovered some of the clusters (contrasting with four cases in the scenario of partitions).

This represent a much better picture than the one that we saw in the partitions scenario, showing the partitions analysis indeed underestimate the amount of useful information hidden inside the partitions. Moreover, it indicates that the simple selection of the clusters identified repeatedly by traditional algorithms (MBCS) may lead to the identification of relevant information.

Finally, to compare the amount of irrelevant information produced by these techniques, Figure 9 presents the ratio between the total number of distinct clusters (by summing up all data sets) obtained with each technique and the total number of distinct true clusters. It is possible to see that the BAlg obtained about 18 times more distinct clusters than the exiting true clusters. By selecting the clusters of BAlg with multiplicity of at least two (MBCS), we only get 8 times more clusters than the true ones, which was the smallest amount of irrelevant clusters obtained.

In summary, none of the strategies was able to recover all clusters, specially in real scenarios.

BAlg and MOCLE were the only strategies which in fact produces solutions. MOCLE starts with partitions produced in BAlg and is supposed to select the good ones as well as to produce new better partitions. However, with rare exceptions, MOCLE was not able to find more information than that present in the initial sets (BAlg) or even selected

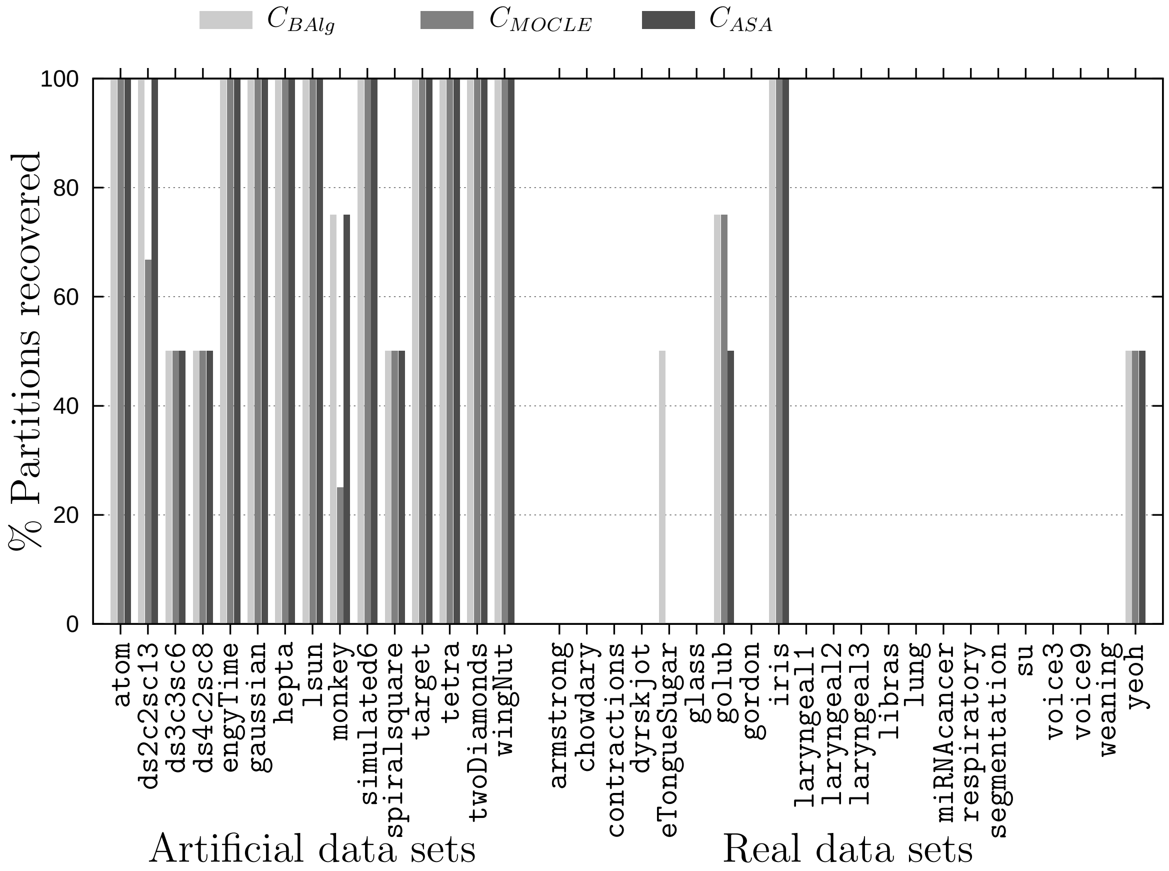


Figure 6: Percentage of true partitions recovered for each data set.

with ASA or MBCS (either in the context of partitions or clusters). In this way, BAlg showed to be the most useful to produce relevant information.

This, together with the employment of data sets with clusters known to be based on different definitions of cluster, show that the combined use of different types of traditional clustering algorithms can be used as a simple strategy for finding flexible multiple solutions regarding clusters definitions. This is much more evident when we look at clusters as the solutions instead of partitions.

As alternatives to select solutions, ASA and MBCS were not able to maintain all the information present in the initial collection of partitions. On the other hand, at the cost of a small loss of relevant information, a significant reduction in irrelevant information produced was achieved together with a reduction in the computational cost required to produce the results. While ASA's complexity is $\mathcal{O}(C^2n^2)$, MBCS's is $\mathcal{O}(C^2n)$. MOCLE is far more complex as its computational cost depends on a large number of different tasks that need to be performed and include many other factors as the dimensionality of the data set and the multi-objective genetic algorithm variables.

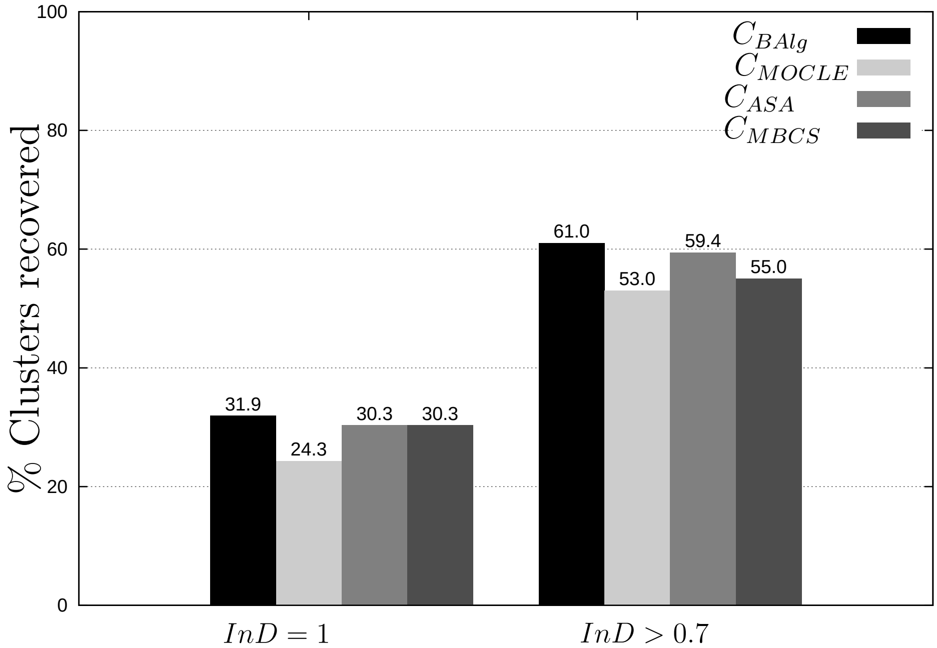


Figure 7: Percentage of true clusters recovered by each technique

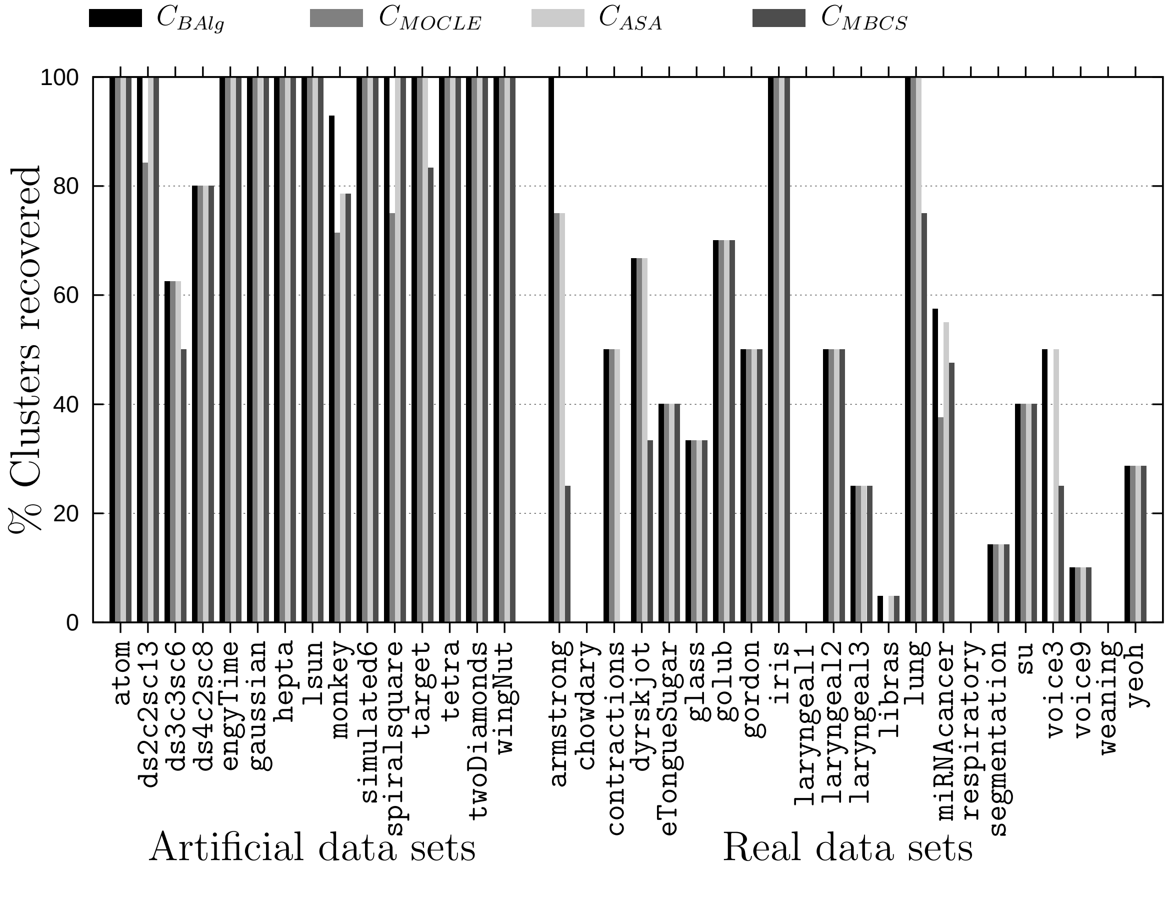


Figure 8: Percentage of true clusters recovered for each data set

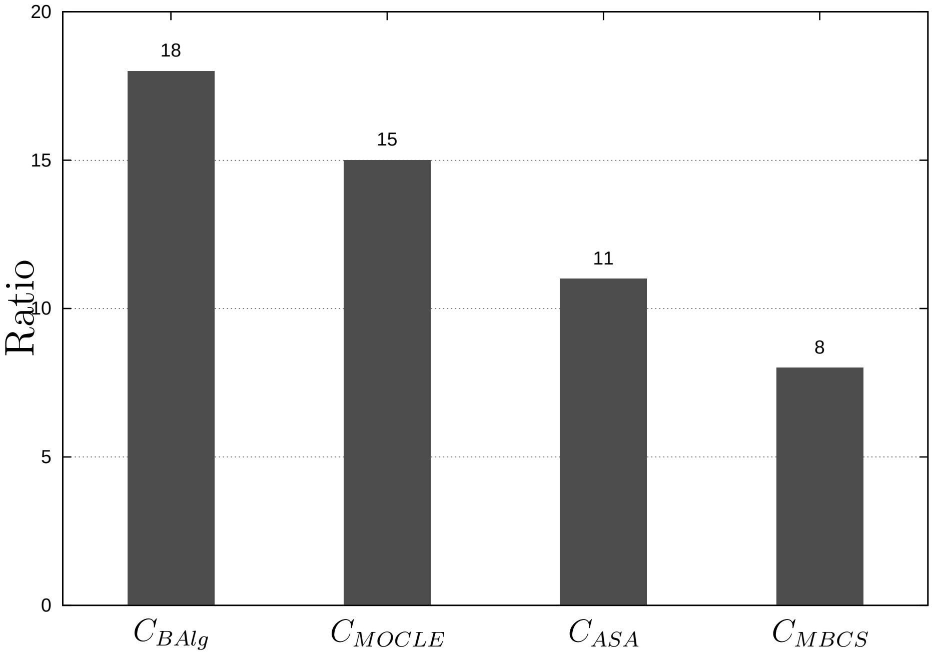


Figure 9: Ratio between obtained clusters and true clusters.

5 Conclusion

In this paper, we provided a methodology for evaluating a set of multiple clustering solutions by considering the clusters themselves as solutions instead of partitions. For such, we rely on three types of techniques for obtaining multiple partitions based on traditional clustering algorithms (multiple complimentary traditional algorithms, multi-objective clustering ensembles, and partition's selection approaches). Then, we proceeded a comparison of the two ways of analyzing multiple solutions: the traditional one by comparing partitions and the analysis of clusters obtained regardless of the partitions they originally belonged. By doing so, we showed that (i) even a diverse set of partitions can have a great amount of redundant information in their clusters, and more importantly, that (ii) the quality of the information extracted is quite underestimated when evaluating them by the partitions analysis.

By focusing on obtaining multiple clusters, we illustrated how different types of traditional clustering algorithms can be used jointly to produce high quality clusters based on different definitions at a cost of generating a great amount of redundant and irrelevant clusters. Multiobjective clustering and strategies of selection relying on partitions diminish the amount of redundant and irrelevant solutions obtained while add an extra high computational cost for processing the initial solutions. Moreover, they lost part of the information obtained by traditional algorithms.

Finally, we showed that the evidence of a cluster can be used as a simple and effective strategy for selecting the most relevant clusters. In fact, we are considering that redundant clusters may indeed represent relevant clusters. In summary, the strategy consists of (i) obtaining a collection of partitions by different types of traditional clustering algorithms, (ii) breaking the obtained partitions into their clusters components and (iii) selecting the redundant clusters as the multiple solutions. We showed that such strategy was able to produce more concise sets of relevant clusters than other more complex approaches relying on partitions.

References

Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, Sallan SE, Lander ES, Golub TR, Korsmeyer SJ (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics 30(1):41–47

Ayad H, Kamel M (2003) Finding natural clusters using multiclusterer combiner based on shared nearest neighbors. In: Proceedings of international workshop on multiple classifier systems, pp 166–175

Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, Ladd C, Beheshti J, Bueno R, Gillette M, Loda M, Weber G, Mark EJ, Lander ES, Wong W, Johnson BE, Golub TR, Sugarbaker DJ, Meyerson M (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma sub-classes. In: Proc. Natl. Acad. Sci. USA, vol 98, pp 13,790–13,795

Chowdary D, Lathrop J, Skelton J, Curtin K, Briggs T, Zhang Y, Yu J, Wang Y, Mazumder A (2006) Prognostic gene expression signatures can be measured in tissues collected in RNAlater preservative. Journal of Molecular Diagnostics 8(1):31–39

Chung CH, Dai BR (2014) A fragment-based iterative consensus clustering algorithm with a robust similarity. Knowledge and Information Systems 41(3):591–609

Coelho A, Fernandes E, Faceli K (2011) Multi-objective design of hierarchical consensus functions for clustering ensembles via genetic programming. Decision Support Systems 51(4):794–809, URL `http://www.sciencedirect.com/science/article/pii/S0167923611000431`

Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: methods and analysis. ACM Transactions on Knowledge Discovery from Data 2(4):1–40

Dyrskjøt L, Thykjaer T, Kruhøffer M, Jensen JL, Marcussen N, Hamilton-Dutoit S, Wolf H, Orntoft TF (2003) Identifying distinct classes of bladder carcinoma using microarrays. Nature Genetics 33(1):90–96

Ertöz L, Steinbach M, Kumar V (2002) A new shared nearest neighbor clustering algorithm and its applications. In: Proceedings of the Workshop on Clustering High Dimensional Data and its Applications, 2nd SIAM International Conference on Data Mining (SDM'2002), pp 105–115

Estivill-Castro V (2002) Why so many clustering algorithms - a position paper. SIGKDD Explorations 4(1):65–75

Faceli K, Carvalho A, Souto M (2007) Multi-objective clustering ensemble. International Journal of Hybrid Intelligent Systems, Special Issue: Ensemble and Integration Approaches, Selected papers contributed to the HIS-NCEI06 conference 4(3):145–156

Faceli K, Carvalho ACFLF, de Souto MCP (2008) Cluster ensemble and multi-objective clustering methods. In: Verma B, Blumenstein M (eds) Pattern Recognition Technologies and Applications: Recent Advances, IGI Global, pp 325–343

Faceli K, Souto MCP, de Araújo DSA, Carvalho ACFLF (2009) Multi-objective clustering ensemble for gene expression data analysis. Neurocomputing 72(13-15):2763–2774, URL `http://dx.doi.org/10.1016/j.neucom.2008.09.025`

Faceli K, Sakata T, de Souto M, de Carvalho A (2010) Partitions selection strategy for set of clustering solutions. Neurocomputing 73(16-18):2809–2819, URL `http://www.sciencedirect.com/science/article/pii/S092523121000281X`

Fern XZ, Brodley CE (2004) Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the Twenty First International Conference on Machine Learning (ICML'2004), ACM Press, New York, NY, USA, p 36, DOI http://doi.acm.org/10.1145/1015330.1015414

Fred ALN, Jain AK (2006) Learning pairwise similarity for data clustering. In: Proceedings of international conference on pattern recognition, pp 925–928

Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. ACM Transactions on Knowledge Discovery from Data 1(1):4–ex

Golub TR, D K Slonim and PT, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh M, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537

Gordon GJ, Jensen RV, Hsiao LL, Gullans SR, Blumenstock JE, Ramaswamy S, Richards WG, Sugarbaker DJ, Bueno R (2002) Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research 62(17):4963–4967

Handl J, Knowles J (2004) Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPSYSBIO-2004-02, UMIST, Manchester

Handl J, Knowles J (2007) An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation 11(1):56–76

Handl J, Knowles J, Kell D (2005) Computational cluster validation in post-genomic data analysis. Bioinformatics 21(15):3201–3212

Hoon MJL, Imoto S, Nolan J, Miyano S (2004) Open source clustering software. Bioinformatics 20(9):1453–1454

Hruschka E, Campello R, Freitas A, Carvalho A (2009) A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man and Cybernetics Part C - Applications and Reviews 39:133–155

Hu X, Yoo I (2004) Cluster ensemble and its applications in gene expression analysis. In: Proceedings of Asia-Pacific Bioinformatics Conference, pp 297–302

Hubert LJ, Arabie P (1985) Comparing partitions. Journal of Classification 2:193–218

Iam-On N, Boongoen T (2015) Comparative study of matrix refinement approaches for ensemble clustering. Machine Learning 98(1-2):269–300

Jain A, Dubes R (1988) Algorithms for Clustering Data. Prentice Hall

Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31(8):651–666, award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 19th International Conference in Pattern Recognition (ICPR)

Jiamthapthaksin R, Eick CF, Vilalta R (2009) A framework for multi-objective clustering and its application to co-location mining. In: Lecture Notes in Computer Science, vol 5678, pp 188–199

Kraus JM, Müssel C, Palm G, Kestler HA (2011) Multi-objective selection for collecting cluster alternatives. Computational Statistics 26(2):341–353

Kuncheva LI, Hadjitodorov ST, Todorova LP (2006) Experimental comparison of cluster ensemble methods. In: Proceedings of FUSION 2006, pp 105–115

Law M, Topchy A, Jain AK (2004) Multiobjective data clustering. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2, pp 424–430

Liu R, Liu Y, Li Y (2012) An improved method for multi-objective clustering ensemble algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation, pp 1–8

Lu J, Getz G, Miska EA, Alvarez-Saavedra EA, Lamb J, Peck D, Sweet-Cordero A, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvitz HR, Golub TR (2005) MicroRNA expression profiles classify human cancers. Nature 435:834–838

Milligan GW, Cooper MC (1986) A study of the comparability of external criteria for hierarchical cluster analysis. Multivariate Behavorial Research 21:441–458

Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52(1-2):91–118

Müller E, Günnemann S, Färber I, Seidl T (2012) Discovering multiple clustering solutions: Grouping objects in different views of the data. In: Proceedings of 10th IEEE International Conference on Data Mining

Müller E, Assent I, Günnemann S, Seidl T (2015) Multiclust special issue on discovering, summarizing and using multiple clusterings. Machine Learning 98(1-2):1–5, editorial

Newman D, Hettich S, Blake C, Merz C (1998) UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html` [Acessado em 06/07/2006], university of California, Irvine, Dept. of Information and Computer Sciences

Parvin H, Minaei-Bidgoli B (2015) A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm. Pattern Analysis and Applications 18(1):87–112

Piantoni J, Faceli K, Sakata TC, Pereira JC, de Souto MC (2015) Impact of base partitions on multi-objective and traditional ensemble clustering algorithms. In: Arik S, Huang T, Lai WK, Liu Q (eds) Neural Information Processing, Lecture Notes in Computer Science, vol 9489, Springer International Publishing, pp 696–704, DOI 10.1007/978-3-319-26532-2_77, URL http://dx.doi.org/10.1007/978-3-319-26532-2_77

Saha S, Bandyopadhyay S (2010) A new multiobjective clustering technique based on the concepts of stability and symmetry. Knowledge and Information Systems 23(1):1–27, DOI 10.1007/s10115-009-0204-4, URL http://dx.doi.org/10.1007/s10115-009-0204-4

Sakata T, Faceli K, De Souto M, De Carvalho A (2010) Improvements in the partitions selection strategy for set of clustering solutions. In: Proceedings of the 11th Brazilian Symposium on Neural Networks, (SBRN'2010), pp 49–54, URL http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=5715212

Sakata T, Faceli K, Almeida T, Riul A, Steluti W (2012) The assessment of the quality of sugar using electronic tongue and machine learning algorithms. In: 11th International Conference on Machine Learning and Applications (ICMLA), vol 1, pp 538–541

Strehl A, Ghosh J (2002) Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research (JMLR) 3:583–617

Su AI, Welsh JB, Sapinoso LM, Kern SG, Dimitrov P, Lapp H, Schultz PG, Powell SM, Moskaluk CA, Frierson HF, Hampton GM (2001) Molecular classification of human carcinomas by use of gene expression signatures. Cancer Research 61(20):7388–7393

Topchy A, Jain AK, Punch W (2005) Clustering ensembles: models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12):1866–1881

Ultsch A (2005) Clustering with som: U*c. In: Workshop on Self-Organizing Maps, Paris, France, pp 75–82

Vega-Pons S, Ruiz-Shulcloper J (2011) A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence 25(3):337–372

Wahid A, Gao X, Andreae P (2014) Multi-view clustering of web documents using multi-objective genetic algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation, pp 2625–2632

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3):645–678

Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui CH, Evans WE, Naeve C, Wong L, Downing JR (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2):133–143