Journal of Machine Learning Research 21 (2020) 1-54 Submitted 7/18; Revised 5/20; Published 6/20

Kernel-estimated Nonparametric Overlap-Based Syncytial Clustering

Israel A. Almodóvar-Rivera israel.almodovar@upr.edu
Department of Biostatistics and Epidemiology
University of Puerto Rico at Medical Science Campus
San Juan, PR 00936-5067, USA

Ranjan Maitra maitra@iastate.edu
Department of Statistics
Iowa State University
Ames, IA, 50011-1090, USA

Editor: Miguel Á. Carreira-Perpiñán

Abstract

Commonly-used clustering algorithms usually find ellipsoidal, spherical or other regular-structured clusters, but are more challenged when the underlying groups lack formal structure or definition. Syncytial clustering is the name that we introduce for methods that merge groups obtained from standard clustering algorithms in order to reveal complex group structure in the data. Here, we develop a distribution-free fully-automated syncytial clustering algorithm that can be used with k-means and other algorithms. Our approach estimates the cumulative distribution function of the normed residuals from an appropriately fit k-groups model and calculates the estimated nonparametric overlap between each pair of clusters. Groups with high pairwise overlap are merged as long as the estimated generalized overlap decreases. Our methodology is always a top performer in identifying groups with regular and irregular structures in several datasets and can be applied to datasets with scatter or incomplete records. The approach is also used to identify the distinct kinds of gamma ray bursts in the Burst and Transient Source Experiment 4Br catalog and the distinct kinds of activation in a functional Magnetic Resonance Imaging study.

Keywords: BATSE, DEMP, DEMP+, DBSCAN*, density peaks algorithm, GRB, GSL-NN, k-clips, k-means, k_m-means, kernel density estimation, KNOB-SynC, MixModCombi, MGHD, MSAL, overlap, PGMM, SDSS, spectral clustering, TiK-means

1. Introduction

Cluster analysis (Ramey, 1985; McLachlan and Basford, 1988; Kaufman and Rousseuw, 1990; Everitt et al., 2001; Melnykov and Maitra, 2010; Xu and Wunsch, 2009; Bouveyron et al., 2019) is an unsupervised learning method that partitions datasets into distinct groups of homogeneous observations. Finding such structure in the absence of group information can be challenging but is important in many applications, such as taxonomical classification (Michener and Sokal, 1957), market segmentation (Hinneburg and Keim, 1999), software management (Maitra, 2001) and so on. As such, a number of methods, ranging from the heuristic (Johnson, 1967; Everitt et al., 2001; Jain and Dubes, 1988; Forgy, 1965; MacQueen, 1967; Kaufman and Rousseuw, 1990) to the more formal, model-based (Titterington et al., 1985; McLachlan and Peel, 2000; Melnykov and Maitra, 2010; McNicholas, 2016; Bouveyron et al., 2019) approaches have been proposed and implemented.

Most common clustering algorithms, whether model-agnostic methods like k-means (MacQueen, 1967; Hartigan and Wong, 1979; Lloyd, 1982) or model-based approaches such as Gaussian mixture models (Fraley and Raftery, 2002; Melnykov and Maitra, 2010) yield clusters with regular dispersions or structure. For instance, the k-means algorithm is geared towards finding homogeneous spherical clusters or spherically-dispersed groups of equal radius. Such algorithms are not designed to find general-shaped or structured groups, therefore, many additional approaches have been suggested to identify irregularly-shaped groups (see, for example, Dhillon et al., 2004; Fred and Jain, 2005; von Luxburg, 2007; Baudry et al., 2010; Hennig, 2010; Melnykov, 2016; Peterson et al., 2018). Kernel k-means clustering (Dhillon et al., 2004) enhances the k-means algorithm by using a kernel function $\phi(\cdot)$ that nonlinearly maps the original (input) space to a higher-dimensional feature space where it may be possible to linearly separate clusters that were not linearly separable in the original space. Spectral clustering (von Luxburg, 2007) uses k-means on the first few eigenvectors of a Laplacian of the similarity matrix of the data. Both methods need the number of clusters to be provided: in the case of spectral clustering, von Luxburg (2007) suggests estimating this number as the one with the highest gap between successive eigenvalues.

A separate set of approaches modifies the distribution of the mixture components in model-based clustering (MBC) by replacing the commonly-used multivariate Gaussian component with other more general distributions. Some of these approaches simply add dimension reduction in the form of factor models (Ghahramani and Hinton, 1997; McNicholas and Murphy, 2008) through parsimonious Gaussian mixture models (PGMM). More generally, Franczak et al. (2014) propose MBC using a mixture of asymmetric shifted Laplace distributions (MixSAL) while Browne and McNicholas (2015) suggest using a mixture of generalized hyperbolic distributions (MixGHD). These approaches more fully exploit MBC but can be CPU intensive and are somewhat limited in capturing complex structures.

Evidence accumulation clustering or EAC (Fred and Jain, 2005) combines results from multiple runs of the k-means algorithm with the underlying rationale that each partitioning provides independent evidence of structure that is then extricated by cross-tabulating the relative frequencies (out of the multiple partitionings) that each observation pair is in the same group. This relative frequency table serves as a similarity matrix for hierarchical clustering: however, implementation of this method can be computationally demanding in terms of CPU speed and memory. Stuetzle and Nugent (2010) developed a nonparametric clustering approach under the premise that each group corresponds to a mode of the estimated multivariate density of the observations. The high-density modes are located and hierarchically clustered with dissimilarity between two modes calculated in terms of the lowest density or number of common points in each mode's domain of attraction. The "density-based spatial clustering algorithm of applications with noise" (DBSCAN) algorithm (Ester et al., 1996) groups together points in high-density regions while identifying points in low-density regions as outliers. A refinement (DBSCAN*, by Campello et al., 2013) follows the same principle but classifies so-called border observations as outliers. Both algorithms depend on the minimum cluster size and *reachability distance*, and also on a cut-off to determine the border and outlying observations. The authors suggest setting this cutoff at the knee of a plot of the k-nearest neighbor distances of the observations. In a similar vein, Rodriguez and Laio (2014) developed a fast Density Peaks (DP) algorithm to determine cluster centers and find outliers while considering the local density of each observation. DP uses the estimated multivariate density in order to classify observations into outliers and does not rely on an explicit cut-off value, but other parameters need to be subjectively specified or estimated to graphically decide on the number of groups. These methods all rely on density estimates and are not immune from the ravages of the curse of dimensionality.

More recent work (Baudry et al., 2010; Hennig, 2010; Melnykov, 2016; Peterson et al., 2018) proposed merging groups found using MBC or k-means. Such methods fall into the category of what we introduce in this paper as syncytial clustering algorithms, because they yield a cluster structure resembling a *syncytium*, a term that in cell biology refers to a multi-nucleated mass of cytoplasm inseparable into individual cells and that can arise from multiple fusions of uninuclear cells. Syncytial clustering algorithms are similar in that they merge or fuse groups that originally corresponded to mixture model components or k-means or other regular-structured groups. Resulting partitions have groups with potentially multiple well-defined and structured sub-groups. We outline a few such algorithms next.

MBC is premised on the idea of a one-to-one correspondence between a mixture component of given density form and group. Such injective mapping assumptions are not always tenable so some authors (Baudry et al., 2010; Hennig, 2010; Melnykov, 2016) model each group as a mixture of (one or more) components. Operationally, we have a syncytial clustering framework where identified mixture components that are not very distinct from each other are merged (Baudry et al., 2010; Hennig, 2010; Melnykov, 2016) into a cluster. Baudry et al. (2010) successively merge mixture component pairs that result in the highest change in entropy, continuing for as long as the entropy increases. This method, abbreviated here as MMC, is implemented in the R (R Development Core Team, 2018) package RMixModCombi (Baudry and Celeux, 2014). Hennig (2010) developed the *directly estimated misclassification probabilities* (DEMP) algorithm to identify candidate components for merging. The author argued that the best measure of group similarity should relate to the classification probability and so proposed that clusters with the highest pairwise misclassification probabilities be merged. The DEMP+ method (Melnykov, 2016) mimics DEMP but replaces the misclassification probabilities of DEMP with the overlap measure of Maitra and Melnykov (2010) for Gaussian mixture components. DEMP+ uses Monte Carlo simulation to determine pairwise overlap between merged components and uses thresholds on the maximum pairwise overlap to determine termination. The sliding threshold was empirically suggested to be chosen to be inversely related to dimension.

The MBC algorithms offer a principled approach to the partitioning of observations into groups but are more demanding in CPU time and perhaps unnecessary to use when the objective is simply to find the most appropriate grouping with no particular dogma regarding shape or structure and where using k-means as a starting point for an initial clustering may be a fairly plausible but faster alternative. Perhaps recognizing this aspect, Melnykov (2016) contended that DEMP+ can be applied to k-means output by assuming equal mixing proportions and homogeneous spherical dispersions in the mixture model. The basis for this assertion is the framing of the k-means algorithm of Lloyd (1982) as a Classification Expectation-Maximization (CEM) Algorithm (see Fraley and Raftery, 1998, for details). But k-means clustering makes hard assignments of each observation and, indeed, most commonly-used statistical software programs, such as R (R Development Core Team, 2018) use the efficient Hartigan and Wong (1979) algorithm that handles computations quite differently and sparingly than Lloyd (1982). In this vein, Peterson et al. (2018) provided the K-mH algorithm to merge poorer-separated k-means groups. Such groups are identified as per an easily-computed index that uses normal theory with spherical dispersion assumptions. However, the K-mH algorithm has a large number of settings and parameters: using default values and rules-of-thumb provided by the authors, we have found that this method performs well in many datasets but not as well in many others. Therefore, it would be worth investigating other syncytial clustering algorithms that use k-means groupings for clustering efficiency while also reducing the need to tune multiple parameter settings.

A separate issue is the impact of Gaussian mixture model assumptions in methods such as DEMP+ when applied to regular-structured groups found using, say, the multivariate t-mixture or other appropriate models. A nonparametric method not taking recourse to such distributional assumptions would be desirable in addressing this shortcoming. This paper therefore proposes the Kernel-estimated Nonparametric Overlap-Based Syncytial Clustering (KNOB-SynC) algorithm that successively merges groups from a well-optimized k-means solution until some objective and nonparametric data-driven cluster overlap measure vanishes or is no longer reduced. This measure is calibrated through the generalized overlap (Maitra, 2010; Melnykov and Maitra, 2011; Melnykov et al., 2012) calculated using smooth estimation of the cumulative distribution function (CDF) developed in Section 2. Our algorithm is illustrated and comprehensively evaluated in Section 3. Although motivated using k-means, the method is general enough to apply to the output of other partitioning algorithms, such as clustering using the Mahalanobis (1936) distance, or in scenarios with scatter (Maitra and Ramler, 2009) or incomplete records (Lithio and Maitra, 2018). Section 4 also applies our methodology to two interesting settings: in the first case, we identify the differents kinds of gamma ray bursts in the most recent Burst and Transient Source Experiment (BATSE) 4Br catalog. Our second application uses KNOB-SynC to identify activation from single replications of a functional Magnetic Resonance Imaging (fMRI) study obtained from a right-hand finger tapping experiment performed by a right-hand-dominant male. We find our results to both be interpretable and with greater reproducibility than current methods. The paper concludes with some discussion. An appendix provides mathematical proofs for our derived theoretical properties of smooth estimation of the CDF using asymmetric kernel density estimation and detailed graphical illustrations of experimental performance on two-dimensional (2D) datasets and numerical summaries of performance on all datasets.

2. Methodological Development

2.1. Problem Setup

Let $\boldsymbol{\Xi} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$ be a random sample of n p-dimensional observations, with each

$$\boldsymbol{X}_i \sim \prod_{c=1}^{C} [f_c(\boldsymbol{x})]^{\zeta_{ic}}, \tag{1}$$

where C is the number of groups, $\zeta_{ic} = \mathcal{I}_{(\boldsymbol{X}_i \in \boldsymbol{\mathcal{C}}_c)}$ with $\mathcal{I}_{(\mathcal{Z})} = 1$ if $\mathcal{Z}$ holds and 0 otherwise, $f_c(\boldsymbol{x})$ is the cluster-specific density of an observation in the cth cluster and $\boldsymbol{\mathcal{C}}_c$ is the set of observations in the sample from that group. Our specification in (1) refers to a hard clustering framework: we marginally obtain a mixture model if we specify independent identical multinary prior distributions on each ζ_{ic}. Our objective is to estimate ζ_{ic}s (equivalently, $\boldsymbol{\mathcal{C}}_c$s) for each $c = 1, 2, \ldots, C$ with C possibly unknown. We also assume that for each $c = 1, 2, \ldots, C$, the density $f_c(\boldsymbol{x})$ for any $\boldsymbol{X}_i \in \boldsymbol{\mathcal{C}}_c$ (*i.e.* $\zeta_{ic} = 1$) can be further described by

$$f_c(\boldsymbol{x}) = \prod_{k=1}^{k_c} [h(\|\boldsymbol{x} - \boldsymbol{\mu}_k^{\boldsymbol{\mathcal{C}}_c}\|)]^{\zeta_{ik}^{\boldsymbol{\mathcal{C}}_c}}, \tag{2}$$

where $h(\cdot)$ is defined on the positive half of the real line so that $h(\|\boldsymbol{x}\|)$ is a zero-centered density in $\mathbb{R}^p$ with spherical level hyper-surfaces. This means that each group in the dataset can be further decomposed into multiple homogeneous spherically-dispersed subgroups, and $\zeta_{ik}^{\boldsymbol{\mathcal{C}}_c} = 1$ if $\boldsymbol{X}_i$ is in $\boldsymbol{\mathcal{C}}_c$ and in the kth subgroup inside $\boldsymbol{\mathcal{C}}_c$, and zero otherwise. That is, we can model $\boldsymbol{X}_i \in \boldsymbol{\Xi}$ as $\boldsymbol{X}_i \sim \prod_{c=1}^{C} \prod_{k=1}^{k_c} [h(\|\boldsymbol{x} - \boldsymbol{\mu}_k^{\boldsymbol{\mathcal{C}}_c}\|)]^{\zeta_{ik}^{\boldsymbol{\mathcal{C}}_c}}$, or equivalently as

$$\boldsymbol{X}_i \sim \prod_{k=1}^{K} [h(\|\boldsymbol{x} - \boldsymbol{\mu}_k^{\circ}\|)]^{\zeta_{ik}^{\circ}}, \tag{3}$$

where ζ_{ik}° and $\boldsymbol{\mu}_k^{\circ}$ for $k = 1, 2, \ldots, K$ are renumerations, respectively, of all the $\zeta_{ik}^{\boldsymbol{\mathcal{C}}_c}$ and $\boldsymbol{\mu}_k^{\boldsymbol{\mathcal{C}}_c}$ for $k = 1, 2, \ldots, k_c, c = 1, 2, \ldots, C$. Therefore, $K = \sum_{c=1}^{C} k_c$, $\zeta_{ic} = \sum_{k=1}^{k_c} \zeta_{ik}^{\boldsymbol{\mathcal{C}}_c}$ for $c = 1, 2, \ldots, C$ and $\sum_{k=1}^{K} \zeta_{ik}^{\circ} \equiv \sum_{c=1}^{C} \sum_{k=1}^{k_c} \zeta_{ik}^{\boldsymbol{\mathcal{C}}_c} = 1$ (however, both K and C are also unknown). The reformulation of (1) in terms of (3) means that the k-means algorithm (Forgy, 1965; Lloyd, 1982; Hartigan and Wong, 1979) can be employed along with cluster-selection methods (for example, Krzanowski and Lai, 1988; Sugar and James, 2003; Maitra et al., 2012) to obtain a first-pass clustering of the dataset where the observations are partitioned into an estimated number ($\hat{K}$) of homogeneous spherically-dispersed groups. Our proposal is to develop methods for identifying the supersets of these k-means (homogeneous spherical) groups to obtain the clusters $\{\boldsymbol{\mathcal{C}}_c; c = 1, 2, \ldots, C\}$ with C also needing to be estimated. These supersets will reveal the general-shaped clustering structure in the data.

From the $\hat{K}$-groups solution, define the ith residual ($i = 1, 2, \ldots, n$) as

$$\hat{\boldsymbol{\epsilon}}_i = \boldsymbol{X}_i - \sum_{k=1}^{\hat{K}} \hat{\boldsymbol{\mu}}_k^{\circ} \hat{\zeta}_{ik}^{\circ}; \tag{4}$$

where $\hat{\boldsymbol{\mu}}_k^{\circ}$ is the multivariate mean vector of the observations in the kth group and $\hat{\zeta}_{ik}^{\circ} = \mathcal{I}_{(\boldsymbol{X}_i \in \text{ } k\text{th } k\text{-means group})}$. From (4), we obtain the normed residuals, that is, we obtain

$$\hat{\Psi}_i = \sqrt{\hat{\boldsymbol{\epsilon}}_i' \hat{\boldsymbol{\epsilon}}_i} = \|\boldsymbol{X}_i - \sum_{i=1}^{\hat{K}} \hat{\zeta}_{ik}^{\circ} \hat{\boldsymbol{\mu}}_k^{\circ}\| \tag{5}$$

for $i = 1, 2, \ldots, n; k = 1, 2, \ldots, \hat{K}$. These $\hat{\Psi}_1, \hat{\Psi}_2, \ldots \hat{\Psi}_n$ may be viewed as a random sample with density function $h(\cdot)$ and CDF $H(\cdot)$ and having support in $[0, \infty)$. We now provide methods for estimating $H(\cdot)$ under assumptions of a smooth CDF.

2.2. Smooth estimation of the CDF of the normed residuals

We first introduce a smooth estimator for an univariate CDF. Let $Y_1, Y_2, \dots, Y_n$ be a random sample having CDF $H(\cdot)$ and probability density function (PDF) $h(\cdot)$. The natural and most common estimator is the empirical CDF (ECDF) defined as

$$\hat{H}_n(y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(Y_i \leq y). \tag{6}$$

It is easy to see that $\hat{H}_n(y)$ is an unbiased estimator of $H(y)$, that is, $\mathbb{E}[\hat{H}_n(y)] = H(y)$. Further, it converges almost surely to the true CDF $H(\cdot)$. However, the ECDF is a step function for any n and so inappropriate for a smooth continuous CDF, even though it is a smooth function in the limit as $n \to \infty$ (Silverman, 1986). An alternative *kernel estimator* (Rosenblatt, 1956; Parzen, 1962; Silverman, 1986; Wand and Jones, 1995) for $H(\cdot)$ replaces the indicator function in (6) by its smooth cousin. Strictly speaking, kernel density estimation is most often employed in nonparametric contexts (Silverman, 1986) but can also be extended to smooth CDF estimation by integrating over the domain of the kernel. Let $G(y) = \int_{-\infty}^{y} K(u)du$ be the CDF of a kernel function $K(\cdot)$. The kernel CDF estimator is then defined as

$$\hat{H}(y; b) = \frac{1}{n}\sum_{i=1}^{n} G\left(\frac{y - Y_i}{b}\right), \tag{7}$$

where b is the *bandwidth* or the *smoothing parameter*. Equation (7) makes the popular assumption of a symmetric kernel, the most common examples of which are the Gaussian and Epanechnikov (Epanechnikov, 1969; Azzalini, 1981; Reiss, 1981) kernels. However, using a symmetric kernel when the support of the distribution is not on the entire real line (as is the case with our normed residuals) causes weights to be assigned outside the domain of the observations, resulting in boundary bias (Bouezmarni and Scaillet, 2005). So Chen (2000) proposed using an asymmetric kernel in (7) based on the gamma density, with behavior similar to the Gaussian kernel and a comparable rate of convergence in terms of the mean squared error. However, Chen (2000)'s estimator is not a valid density for finite sample sizes (Jeon and Kim, 2013) so we consider the Reciprocal Inverse Gaussian (RIG) kernel density estimator (Scaillet, 2004)

$$\hat{h}(y; b) = \frac{1}{n}\sum_{i=1}^{n} K\left(y; Y_i, b\right), \tag{8}$$

with $K(y; Y_i, b) = \frac{1}{\sqrt{2\pi b Y_i}} \exp\{-\frac{1}{2bY_i}[Y_i - (y - b)]^2\}$. Since $K(y; Y_i, b)$ is a smooth function the CDF estimate is defined as $G(y; Y_i, b) = \int_0^y K(t; Y_i, b)dt$. Then, integrating $\hat{h}(y; b)$ with respect to y yields the smooth CDF estimator

$$\hat{H}(y; b) = \frac{1}{n}\sum_{i=1}^{n} G\left(y; Y_i, b\right) = \frac{1}{n}\sum_{i=1}^{n}\left[\Phi\left(\frac{Y_i + b}{\sqrt{Y_i b}}\right) - \Phi\left(\frac{Y_i - (y - b)}{\sqrt{Y_i b}}\right)\right], \tag{9}$$

where $\Phi(\cdot)$ is the standard Gaussian CDF. An added benefit of using the RIG kernel over the gamma kernel is that the estimated CDF is in closed form and can be readily evaluated using standard software. We now investigate some theoretical properties of the asymmetric RIG kernel CDF estimator. Before proceeding however, we revisit the definition of the Inverse Gaussian and RIG densities for the sake of completeness and to fix ideas.

Definition 1 *A nonnegative random variable $U_{\mu,\lambda}$ is said to arise from the Inverse Gaussian distribution with parameters (μ,λ) if it has the density*

$$r(u;\mu,\lambda)=\begin{cases}\frac{\sqrt{\lambda}}{u\sqrt{2\pi u}}\exp\left\{-\frac{\lambda}{2\mu}\left(\frac{u}{\mu}-2+\frac{\mu}{u}\right)\right\}, & u>0\\ 0 & \textit{otherwise.}\end{cases}\tag{10}$$

Notationally, we write $U_{\mu,\lambda}\sim IG(\mu,\lambda)$. Also, we have $\mathbb{E}(U_{\mu,\lambda})=\mu$ and $\mathbb{V}ar(U_{\mu,\lambda})=\mu^3/\lambda$.

Definition 2 *A nonnegative random variable $V_{\mu,\lambda}$ is said to be from the Reciprocal Inverse Gaussian distribution with parameters (μ,λ) if it has the density*

$$s(v;\mu,\lambda)=\begin{cases}\frac{\sqrt{\lambda}}{\sqrt{2\pi v}}\exp\left\{-\frac{\lambda}{2\mu}\left(v\mu-2+\frac{1}{\mu v}\right)\right\}, & v>0\\ 0 & \textit{otherwise.}\end{cases}\tag{11}$$

Notationally, $V_{\mu.\lambda}\sim RIG(\mu,\lambda)$. Further, $V_{\mu.\lambda}$ is equivalent in law to $1/U_{\mu,\lambda}$ where $U_{\mu,\lambda}\sim IG(\mu,\lambda)$ and $\mathbb{E}(V_{\mu,\lambda})=1/\mu+1/\lambda$ while $\mathbb{V}ar(V_{\mu,\lambda})=(\lambda+2\mu)/(\lambda^2\mu)$.

We now develop some properties of the asymmetric kernel RIG to estimate the CDF.

Lemma 3 *Let $Y_1,Y_2,\ldots,Y_n$ be independent identically distributed nonnegative-valued random variables with CDF $H(y)$, and PDF $h(y)$ that is infinitely differentiable. Also, consider the RIG kernel density defined by $K(t;Y_i,b)=\phi[(Y_i-(t-b))/\sqrt{bY_i}]/\sqrt{bY_i}$ where $\phi(z)$ is the standard normal density evaluated at z. Consider estimating $H(y)$ using $\hat{H}(y;b)$ as defined in (9). Then, as $b\to 0$, $\mathbb{E}[\hat{H}(y;b)]=H(y)+b[yh'(y)-h(y)]/2+o(b)\equiv H(y)+\mathcal{O}(b)$ and $\mathbb{V}ar[\hat{H}(y;b)]\approx H(y)(1-H(y))/n-H(y)/(2n)-H(y)\sqrt{b}/(2n\sqrt{2\pi y})+o(\sqrt{b})\equiv H(y)(1-H(y))/n-H(y)/(2n)+\mathcal{O}(b)$.*

Proof See Appendix A.1. ■

Lemma 3 shows that $\hat{H}(y;b)$ has lower variance than the ECDF and has point-wise Mean Squared Error (MSE) at y that is given by $\text{MSE}[\hat{H}(y;b)]=\mathbb{V}\text{ar}[\hat{H}(y;b)]+[\text{Bias}\{\hat{H}(y;b)\}]^2$.

2.2.1. Bandwidth selection

Scaillet (2004) minimized the Mean Integrated Squared Error (MISE) to provide a rule-of-thumb bandwidth selector for the RIG kernel density estimator of the form

$$\hat{b}=\left[\frac{2\int_0^\infty y^{-1/2}h(y)\mathrm{d}y}{\sqrt{\pi}\int_0^\infty y^2\{h''(y)\}^2\mathrm{d}y}\right]^{2/5}n^{-2/5}.\tag{12}$$

However, (12) involves knowledge of the true density $h(\cdot)$ and is directly unusable. Scaillet (2004) proposed obtaining $\hat{b}$ by assuming an initial parametric density, say $h(\cdot;\boldsymbol{\theta})$, for $h(\cdot)$ and estimating the parameters $\boldsymbol{\theta}$ of the density from the sample. Exact derivations using a lognormal density for $h(\cdot;\boldsymbol{\theta})$ were provided (Scaillet, 2004) but this approach has been found to produce estimates that are biased downwards. We therefore adopt Scaillet (2004)'s approach but use an initial gamma density $h(y;\vartheta,\tau) = \exp(-y/\tau)y^{\vartheta-1}\tau^{\vartheta}/\Gamma(\vartheta)$ for $y > 0$ and zero otherwise. Under this setup, $\int_0^\infty y^{-1/2}h(y;\vartheta,\tau)\mathrm{d}y = \Gamma(\vartheta-1/2)\sqrt{\tau}/\Gamma(\vartheta)$ and $\int_0^\infty y^2\{h''(y;\vartheta,\tau)\}^2\mathrm{d}y = (6\vartheta-4)(\vartheta-1)\Gamma(2\vartheta)/\{4^{\vartheta}\tau^3\Gamma^2(\vartheta)(2\vartheta-1)\}$. Therefore, we have

$$\hat{b} = n^{-\frac{2}{5}}\left[\frac{2^{2\hat{\vartheta}+1}\hat{\tau}^{7/2}(2\hat{\vartheta}-1)\Gamma(\hat{\vartheta}-\frac{1}{2})\Gamma(\hat{\vartheta})}{\sqrt{\pi}(6\hat{\vartheta}-4)(\hat{\vartheta}-1)\Gamma(2\hat{\vartheta})}\right]^{\frac{2}{5}} \tag{13}$$

with $\hat{\vartheta}$ and $\hat{\tau}$ estimated from the sample $Y_1, Y_2, \ldots, Y_n$ using, for example, the method of moments. This $\hat{b}$ is used in (9) to obtain our smoothed RIG-kernel CDF estimator.

The development of this section, when applied to the normed residuals $\hat{\Psi}_1, \hat{\Psi}_2, \ldots, \hat{\Psi}_n$ (in place of $Y_1, Y_2, \ldots, Y_n$), yields a smooth nonparametric kernel-based estimator of their CDF. We use this kernel-estimated CDF in our development of the nonparametric estimation of the overlap measure between groups.

2.3. A nonparametric estimator of overlap between groups

Overlap between two groups is an indicator of the extent to which they are indistinguishable from each other. Maitra and Melnykov (2010) defined the pairwise overlap of two mixture components as the sum of the misclassification probabilities $\omega_{lk} \equiv \omega_{kl} = \omega_{l|k} + \omega_{k|l}$ with

$$\omega_{l|k} = \mathbb{P}[\boldsymbol{X} \text{ is assigned to } \boldsymbol{\mathcal{C}}_l \mid \boldsymbol{X} \text{ is truly in } \boldsymbol{\mathcal{C}}_k]. \tag{14}$$

For any two mixture components with densities $f(\boldsymbol{x};\boldsymbol{\theta}_k)$ and $f(\boldsymbol{x};\boldsymbol{\theta}_l)$ and mixing proportions π_k and π_l, we have

$$\omega_{k|l} = \mathbb{P}\left[\pi_k f(\boldsymbol{x}_i;\boldsymbol{\theta}_k) < \pi_l f(\boldsymbol{x}_i;\boldsymbol{\theta}_l) | \boldsymbol{x}_i \in f(\boldsymbol{x}_i;\boldsymbol{\theta}_l)\right],$$

where $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_l$ are the parameter sets associated with the kth and lth mixture components.

Maitra and Melnykov (2010) calculated (14) for Gaussian mixture densities, but the definition itself is general enough to include other clustering situations including those as general as when we have cluster distributions given by densities of the type in (1). For an equal-proportioned mixture of homogeneous spherical Gaussian densities, Maitra and Melnykov (2010) showed that $\omega_{k|l} = \Phi(\|\boldsymbol{\mu}_l - \boldsymbol{\mu}_k\|/2\sigma)$ between the kth and the lth cluster where $\Phi(\cdot)$ is the standard Gaussian CDF, $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_l$ are the kth and the lth cluster means and σ is the common (homogeneous) standard deviation for each group, estimated unbiasedly as $WSS_K/\{(n-K)p\}$ with WSS_K being the optimized value of the within-sums-of-squares (WSS) of the K-groups solution. The sum of $\omega_{k|l}$ and $\omega_{l|k}$ reduces to $\omega_{kl} = 2\Phi(\|\boldsymbol{\mu}_l - \boldsymbol{\mu}_k\|/2\sigma)$. The k-means formulation of (3) can be viewed more generally (Maitra et al., 2012) and extends beyond the case of Gaussian-distributed groups, so we develop nonparametric methods for estimating the overlap measure.

2.3.1. Pairwise overlap between two k-means groups

The pairwise overlap (14) between two groups can generally be calculated from $H_\Psi(\cdot)$ as

$$\omega_{l|k} = \mathbb{P}\left(\|\boldsymbol{X} - \boldsymbol{\mu}_l\| < \|\boldsymbol{X} - \boldsymbol{\mu}_k\| \mid \boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k\right) = 1 - \mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right) \tag{15}$$

where Ψ_k represents the normed residual obtained from the kth group, and $\Psi_{l(k)}$ represents the normed *pseudo-residual* which we define as the norm of the remainder that is obtained by subtracting the lth cluster mean $\boldsymbol{\mu}_l$ from an observation $\boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k$. Let $H_\Psi(y)$ be the RIG kernel-estimated smooth CDF obtained using the bandwidth selected as per (13). Then, $\mathbb{P}(\Psi_k < y)$ can be estimated using $\hat{H}_\Psi(y; \hat{b})$ (where Ψ in the subscript of $\hat{H}(\cdot;\cdot)$ denotes that the estimated CDF uses the normed residuals). However, the calculation of $\mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right)$ is not as straightforward. So we estimate $\mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right)$ using a naïve average estimator

$$\hat{\mathbb{P}}\left(\Psi_k < \Psi_{l(k)}\right) = \frac{1}{n_k^\circ} \sum_{i=1}^{n} \hat{\zeta}_{ik}^\circ \hat{H}_\Psi(\|\boldsymbol{X}_i - \hat{\boldsymbol{\mu}}_l^\circ\|; \hat{b}), \tag{16}$$

where $n_k^\circ = \sum_{i=1}^{n} \hat{\zeta}_{ik}^\circ$. The naïve estimator (16) can be considered as an empirical estimator of $\boldsymbol{E}[\hat{H}_\Psi(\|\boldsymbol{X}_i - \hat{\boldsymbol{\mu}}_l\|; \hat{b}) \mid \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\mu}}_l, \boldsymbol{X}_i \in$ kth spherically-dispersed group]. Similar estimates of $\omega_{k|l}$, and therefore ω_{kl}, can be obtained. We call this estimated overlap $\hat{\omega}_{kl} \equiv \hat{\omega}_{lk}$.

2.3.2. Pairwise overlap between two composite groups

As described in (2), a composite group is one that can be further decomposed into subpopulations. We now extend the definition of the pairwise overlap for such groups.

Let $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k}$ be defined as in (14) but for composite groups. That is, we use $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k}$ rather than $\omega_{l|k}$ in order to specify that the overlap measure is between composite clusters $\boldsymbol{\mathcal{C}}_l$ and $\boldsymbol{\mathcal{C}}_k$. Now $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k} = 1 - \mathbb{P}[\min_{r\in\boldsymbol{\mathcal{C}}_k} \|\boldsymbol{X} - \boldsymbol{\mu}_r\| < \min_{j\in\boldsymbol{\mathcal{C}}_l} \|\boldsymbol{X} - \boldsymbol{\mu}_j\| \mid \boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k]$. Suppose now that $\boldsymbol{\mathcal{C}}_{s\subset k}^\circ$ is the sth spherical sub-cluster of $\boldsymbol{\mathcal{C}}_k$ with mean $\boldsymbol{\mu}_s^\circ$, $s = 1, 2, \ldots, |\boldsymbol{\mathcal{C}}_k|$, with $|\boldsymbol{\mathcal{C}}_k|$ being the number of spherical sub-clusters in $\boldsymbol{\mathcal{C}}_k$. We assume that if $\boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k$, then $\operatorname{argmin}_{r\in\{1,2,\ldots,|\boldsymbol{\mathcal{C}}_k|\}} \|\boldsymbol{X} - \boldsymbol{\mu}_r^\circ\| = s \subset k$ implies that $\boldsymbol{X}$ is in the subgroup given by $\boldsymbol{\mathcal{C}}_{s\subset k}$. Under this assumption, the density of $\boldsymbol{X}$ is defined through its (sth) sub-cluster and so

$$\mathbb{P}\left(\min_{r\in\boldsymbol{\mathcal{C}}_k} \|\boldsymbol{X} - \boldsymbol{\mu}_r\| \leq y \mid \boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k\right) = 1 - \mathbb{P}\left(\min_{r\in\boldsymbol{\mathcal{C}}_k} \Psi_r > y\right) = 1 - [1 - \mathbb{P}(\Psi_r \leq y)]^{|\boldsymbol{\mathcal{C}}_k|} \tag{17}$$

where Ψ_r is a normed residual (obtained, for instance, from the k-means solution) for the rth spherically-dispersed subgroup in the kth cluster. We use the RIG kernel distribution estimator to obtain $\mathbb{P}(\Psi_r < y)$. From (14), and using the same ideas as in (16) we get the naïve estimator

$$\hat{\omega}_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k} = \left[1 - \frac{1}{n_c} \sum_{i=1}^{n_c} \hat{\boldsymbol{\zeta}}_{ic} \hat{H}_\Psi(\min_{r\in\boldsymbol{\mathcal{C}}_l} \|\boldsymbol{X}_i - \boldsymbol{\mu}_r\|; \hat{b})\right]^{|\boldsymbol{\mathcal{C}}_k|} \tag{18}$$

and similarly for $\hat{\omega}_{\boldsymbol{\mathcal{C}}_k|\boldsymbol{\mathcal{C}}_l}$, from where we calculate $\hat{\omega}_{\boldsymbol{\mathcal{C}}_l\boldsymbol{\mathcal{C}}_k} \equiv \hat{\omega}_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l} = \hat{\omega}_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k} + \hat{\omega}_{\boldsymbol{\mathcal{C}}_k|\boldsymbol{\mathcal{C}}_l}$. Our definitions of $\boldsymbol{\mathcal{C}}_k$s and $\hat{\omega}_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l}$ are consistent in the sense that if $\boldsymbol{\mathcal{C}}_k = \{k\}$ and $\boldsymbol{\mathcal{C}}_l = \{l\}$ are both k-means groups, then $\hat{\omega}_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l} = \hat{\omega}_{kl}$. We use this equivalence in the description of our KNOB-SynC algorithm in Section 2.4 below.

2.3.3. Summarizing overlap in a partitioning

Our development so far has provided us with pairwise overlap measures for k-means-type (Section 2.3.1) and composite (Section 2.3.2) groups. For a K-groups (whether of the composite or k-means type) partitioning, we get $\binom{K}{2}$ pairwise overlap measures. Summarizing the pairwise overlap measures is important to provide a sense of clustering complexity so Maitra and Melnykov (2010) originally proposed regulating $\breve{\omega}$ (maximum of all pairwise overlaps) and $\bar{\omega}$ (average of all $\binom{K}{2}$ pairwise overlaps) and demonstrated (see Figures 2 and 3 of Maitra and Melnykov, 2010) the ability to summarize a wide range of cluster geometries. However, because specifying two measures simultaneously is cumbersome, later versions of the CARP (Melnykov and Maitra, 2011) and MixSim (Melnykov et al., 2012) software packages borrowed ideas from Maitra (2010) to obtain the *generalized overlap* $\ddot{\omega} = (\check{\lambda}_{\mathbf{\Omega}} - 1)/(K-1)$ where $\check{\lambda}_{\mathbf{\Omega}}$ is the largest eigenvalue of the (symmetric) matrix $\mathbf{\Omega}$ of pairwise overlaps $\omega_{l,k}$ ($\omega_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l}$ for composite groups) and with diagonal entries that are all 1. $\ddot{\omega}$ lies in [0,1] with zero indicating perfect separation between all group densities and 1 indicating indistinguishability between any of them. In this paper, we obtain the estimated generalized overlap $\hat{\ddot{\omega}}$ using the estimated matrix $\hat{\mathbf{\Omega}}$ with off-diagonal entries given by the kernel-estimated pairwise overlaps $\hat{\omega}_{l,k}$ or $\hat{\omega}_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l}$, depending on whether we have simple k-means-type or composite groups.

2.4. The KNOB-SynC Algorithm

Having provided theoretical development for the machinery that we will use, we now describe our multi-phased KNOB-SynC algorithm:

1. *The k-means phase:* This phase finds the optimal partition of the dataset in terms of homogeneous spherically-dispersed groups and has the following steps:

   (a) For each $K \in \{1, 2, \ldots, K_{\max}\}$, obtain K-means partitions initialized each of nKp times with K distinct seeds randomly chosen from the dataset and run to termination. The best – in terms of the value of the objective function (WSS) at termination – of each set of nKp runs is our putative optimal K-means partition for that $K \in \{1, 2, \ldots, K_{\max}\}$. We use $K_{\max} = \max\{\sqrt{n}, 50\}$.

   (b) When n is small relative to p (operationally, $n < p^2$), use Krzanowski and Lai (1988)'s KL criterion to decide on the optimal K. Otherwise, for larger n, we use the jump statistic (Sugar and James, 2003) on the optimal K-means partitions ($K \in \{1, 2, \ldots, K_{\max}\}$) obtained in Step 1a to determine the optimal K (denoted by $\hat{K}$). In calculating the jump statistic, we have used $y = p/2$, which has become the default in most applications. We refer to Sugar and James (2003) for more detailed discussion on this choice of y. The corresponding $\hat{K}$-means solution is the optimal homogeneous spherically-dispersed partition of the dataset. This concludes the k-means phase of the algorithm.

2. *The initial overlap calculation phase:* This phase starts with the output of Step 1. That is, we start with a structural definition of the dataset in terms of $\hat{K}$ optimal homogeneous spherically-dispersed groups. Our objective here is to calculate the overlap between each of these groups using nonparametric kernel estimation methods. We proceed as follows:

   (a) For each observation $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, compute its normed residual $\hat{\Psi}_i = \sqrt{\hat{\boldsymbol{\epsilon}}_i'\hat{\boldsymbol{\epsilon}}_i}$ where $\hat{\boldsymbol{\epsilon}}_i$ is defined as in (4). Also, obtain the normed pseudo-residual $\hat{\Psi}_{i;l(k)} = \|\boldsymbol{X}_i - \hat{\boldsymbol{\mu}}_l\|$ for $\boldsymbol{X}_i \in \boldsymbol{\mathcal{C}}_k$, and $l \neq k \in \{1, 2, \ldots, \hat{K}\}$.

(b) Using the set of normed residuals $\{\hat{\Psi}_i; i = 1, 2, \dots, n\}$, obtain its RIG-kernel-estimated CDF using (7) with bandwidth determined as per (13).

(c) For any two groups $k \neq l \in \{1, 2, \dots, \hat{K}\}$, estimate the pairwise overlap $\hat{\omega}_{lk} = \hat{\omega}_{l|k} + \hat{\omega}_{k|l}$, where $\hat{\omega}_{l|k}$ and $\hat{\omega}_{k|l}$ are calculated using (15) and (16). We obtain the estimated overlap matrix $\hat{\boldsymbol{\Omega}}$ (with diagonal elements all equal to unity). For clarity, denote this overlap matrix as $\hat{\boldsymbol{\Omega}}^{(1)}$ and pairwise overlaps as $\hat{\omega}^{(1)}_{kl} \equiv \hat{\omega}^{(1)}_{\boldsymbol{\mathcal{C}}_k \boldsymbol{\mathcal{C}}_l}$.

(d) From the overlap matrix $\hat{\boldsymbol{\Omega}}^{(1)}$, calculate the generalized overlap $\ddot{\omega}$. Call it $\ddot{\omega}^{(1)}$.

3. *The merging phase:* The merging phase is triggered only if some of the overlap measures between overlapping clusters are more than the others (operationally, if $4\ddot{\omega}^{(1)} \ngeq \check{\omega}$ where $\check{\omega}$ is the maximum of the estimated pairwise overlaps) or if $\ddot{\omega}$ is not negligible, that is, if $\ddot{\omega}^{(1)} \not\approx 0$ (operationally $\ddot{\omega}^{(1)} \geq 10^{-5}$). In that case, this phase merges groups, provides pairwise overlap measures between newly-formed composite groups, the updated overlap matrix and the generalized overlap, continuing for as long as the generalized overlap keeps decreasing (by at least 10^{-5}) or is not negligible. Specifically, this phase iteratively proceeds for $\ell = 1, 2, \dots$ with the following steps:

   (a) Merge the groups with the maximum overlap and every pair of groups that have individual pairwise overlaps substantially larger than the generalized overlap $\ddot{\omega}^{(\ell)}$. That is, merge every pair of groups $\mathcal{C}_k$, $\mathcal{C}_l$, $k \neq l$ such that $\hat{\omega}^{(\ell)}_{lk} \equiv \check{\omega}^{(\ell)}$ or $\hat{\omega}^{(\ell)}_{lk} > \kappa\ddot{\omega}^{(\ell)}$, for some κ as described in the comments section below. Call the new merged group $\boldsymbol{\mathcal{C}}_{\min(k,l)}$ and decrease the index labels of the groups with indices greater than $\max(k, l)$. Decrement $\hat{K}$ by 1 for every merged pair.

   (b) Using (18), update the pairwise overlap measures that have changed as a result of the merges in Step 3a. Call the updated measures $\hat{\omega}^{(\ell+1)}_{\boldsymbol{\mathcal{C}}_k \boldsymbol{\mathcal{C}}_l}$. Obtain the updated overlap matrix (call it $\hat{\boldsymbol{\Omega}}^{(\ell+1)}$) and the updated generalized overlap $\ddot{\omega}^{(\ell+1)}$. Set $\ell \leftarrow \ell + 1$.

   (c) The merging phase terminates if $\ddot{\omega}^{(\ell)} > \ddot{\omega}^{(\ell-1)}$, $\ddot{\omega}^{(\ell)} \approx 0$, or $\ddot{\omega}^{(\ell)} \approx \check{\omega}^{(\ell)}$. The terminating $\hat{K}$ is the $\hat{C}$ of (1).

4. *Final clustering solution:* The grouping $\{\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2, \dots, \boldsymbol{\mathcal{C}}_{\hat{C}}\}$ at the end of the merging phase is the final partition of the dataset. This gives us a total of $\hat{C}$ general-shaped groups in the dataset.

Comments: We provide some additional remarks on KNOB-SynC and relate it to other algorithms for finding general-shaped clusters and settings:

1. The k-means phase finds regular-structured (more specifically, homogeneous spherical) groups and, in this regard, is similar to the initial stages of K-mH (Peterson et al., 2018) and EAC (Fred and Jain, 2005). However, EAC repeats k-means with fixed K several times and is built upon the premise that each k-means run does not end up with the same clustering, especially when we do not have underlying homogeneous spherically-dispersed groups. On the other hand, K-mH uses a separability index built

on Gaussian assumptions for each cluster and has a large number of user-specified parameters. KNOB-SynC uses nonparametric CDF estimation with a plugin bandwidth selector and a naïve average estimator to calculate the overlap between spherically-dispersed groups and a naïve estimator for the overlap between composite groups. Our methodology has one parameter (κ) that is chosen in a completely data-driven framework. No parameter requires fine-tuning by the practitioner. Also, the number of general-structured groups is decided upon termination that is objectively declared whenever the generalized overlap vanishes or does not go down further.

2. As with MMC, DEMP or DEMP+, the use of cluster distributions in the overlap calculations simplifies and keeps practical computations even for large datasets. In contrast, EAC, DBSCAN, DBSCAN*, DP and K-mH require memory-intensive cross-tabulation of the entire dataset across multiple clusterings because $n \times n$ frequency tables need to be calculated and/or stored.

3. KNOB-SynC uses a naïve estimator to update the overlap between composite groups, unlike DEMP+ which uses Monte Carlo simulations and is slower. Further, DEMP+ uses the maximum overlap that is very sensitive to individual pairwise overlap measures while KNOB-SynC uses the generalized overlap measure (Maitra, 2010) that provides a nonlinear summary of all the individual pairwise overlaps.

4. Unlike DEMP or DEMP+, the stopping criterion of KNOB-SynC is data-driven, thus allowing for the possibility of obtaining well-separated and less well-separated partitionings as supported by the data. Our algorithm also has the potential, unlike MMC, DEMP or DEMP+, to merge multiple pairs of groups in a step.

5. KNOB-SynC uses nonparametric CDF estimation but does so in univariate space by exploiting the inherent spherically-dispersed structure (ellipsoidal in the case of clustering with the Mahalanobis distance) of the sub-clusters. Therefore, it has greater immunity against the curse of dimensionality that bedevils multivariate density estimation that is used in algorithms such as DBSCAN* and DP.

6. The parameter κ determines the types of composite groups that are formed. For larger values of κ, we have groups formed by merging a few pairs at each iteration while smaller values κ prefer many simultaneous mergers. (For $\kappa \to \infty$, no merging is possible.) In the first case, we expect to have stringy groups while in the second case, we find clusters that are irregular-shaped but less stringy. A data-driven approach to choosing κ, that we adopt, runs the algorithm with different values of $\kappa = 1, 2, 3, 4, 5, \infty$ and uses the final partitioning with the smallest terminating $\ddot{\omega}$ as the optimal clustering.

7. Unlike other syncytial clustering algorithms like DEMP, DEMP+ or K-mH, KNOB-SynC allows for the possibility of multiple pairs of groups to be merged at an iteration.

8. Our initial stage uses k-means for speed and efficiency that also allows us to explore larger candidate values of K. However, the approach could very well have been used with clustering algorithms obtained using, say, the *generalized Mahalanobis distance*. The overlap calculations are then easily modified. To see this, suppose that the

generalized Mahalanobis distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by $d_{\boldsymbol{\Gamma}}(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})'\boldsymbol{\Gamma}^-(\boldsymbol{x}-\boldsymbol{y})$, where $\boldsymbol{\Gamma}$ is any appropriate nonnegative-definite matrix with (say, Moore-Penrose) inverse given by $\boldsymbol{\Gamma}^-$. (A positive definite $\boldsymbol{\Gamma}$ leads to the usual Mahalanobis distance.) Under the generalized Mahalanobis distance framework, (14) reduces to

$$\begin{aligned} \omega_{l|k} &= \mathbb{P}\left(\|(\boldsymbol{\Gamma}^-)^{1/2}(\boldsymbol{X}-\boldsymbol{\mu}_l)\| < \|(\boldsymbol{\Gamma}^-)^{1/2}(\boldsymbol{X}-\boldsymbol{\mu}_k)\| \mid \boldsymbol{X} \in \text{ } k\text{th group }\right) \\ &= 1 - \mathbb{P}\left(\|(\boldsymbol{\Gamma}^-)^{1/2}(\boldsymbol{X}-\boldsymbol{\mu}_k)\| < \|(\boldsymbol{\Gamma}^-)^{1/2}(\boldsymbol{X}-\boldsymbol{\mu}_l)\| \mid \boldsymbol{X} \in \text{ } k\text{th group }\right), \end{aligned} \tag{19}$$

which means that the problem reduces to the Euclidean case if we use what we here refer to as the normed Mahalanobis-free residuals (and pseudo-residuals). Operationally, this is equivalent to obtaining $\hat{\boldsymbol{\varepsilon}}_i = (\boldsymbol{\Gamma}^-)^{1/2}(\boldsymbol{X} - \sum_{i=1}^K \hat{\zeta}^\circ_{ik}\hat{\boldsymbol{\mu}}^\circ_k)$, and replacing the $\hat{\epsilon}_i$ with $\hat{\boldsymbol{\varepsilon}}_i$ in the calculation of (5) and proceeding as before. This framework also includes the case when we scale each variable before clustering, as happens when the features are on vastly different scales, or when we use principal components (PCs) as our clustering variables – we illustrate these scenarios in Sections 3.3.6 and 4.1.

9. Our algorithm accommodates clustering scenarios in the presence of scatter as provided, for instance, by the output of the k-clips algorithm of Maitra and Ramler (2009). Scatter observations are those that are unlike any other and may be considered as individual groups in their own right. KNOB-SynC incorporates these scatter observations as individual clusters in addition to the groups found from the output of the k-clips algorithm and proceeds with the overlap calculation and merging phases as described earlier in this section. We illustrate this scenario in Section 3.4.1.

10. Datasets often have incomplete records with missing observations in some features. The k_m-means algorithm (Lithio and Maitra, 2018) provides a k-means type algorithm for Euclidean distance clustering in this setting. Then, instead of k-means, KNOB-SynC can incorporate results from k_m-means in the first stage. For the incompletely-observed records, we calculate the rescaled normed residual in the presence of missing information by removing the missing value from their calculation and re-weighting it appropriately. Specifically, we calculate the ith rescaled normed residual as

$$\hat{\varPsi}_i = \frac{p}{p_i}\sum_{l=1}^{p_i}(X_{ij_l} - \hat{\mu}_{kj_l})^2 \tag{20}$$

where $X_{ij_1}, X_{ij_2}, \ldots, X_{ij_{p_i}}$ represent the p_i available features for the ith record that has been assigned to the kth spherically-dispersed sub-group with estimated mean $\hat{\boldsymbol{\mu}}_k$. Similar arguments allow for the calculation of the rescaled normed pseudo-residual $\hat{\varPsi}_{il(k)}$. The use of the nonparametric CDF estimator in KNOB-SynC provides us with the flexibility to calculate the initial overlap estimates from these scaled-up normed residuals. The merging phase and termination criteria of our algorithm remain unchanged. Section 3.4.2 illustrates KNOB-SynC on a dataset with incomplete records.

11. The use of nonparametric methods in the overlap calculations means that a large number of methods may be possible to use in the initial partitional phase. The method

> can also potentially be modified to apply to other kinds of datasets. For instance, the initial clustering can be done for categorical datasets using k-modes (Huang, 1997, 1998; Chaturvedi et al., 2001; Dorman and Maitra, 2020) and then the Generalized or Gaussianized Distributional Transform (Rüschendorf, 2013; Zhu et al., 2019) and copula model (Nelsen, 2006) can potentially be applied to each cluster to obtain numerical-valued residuals for use with our overlap estimation and calculations.

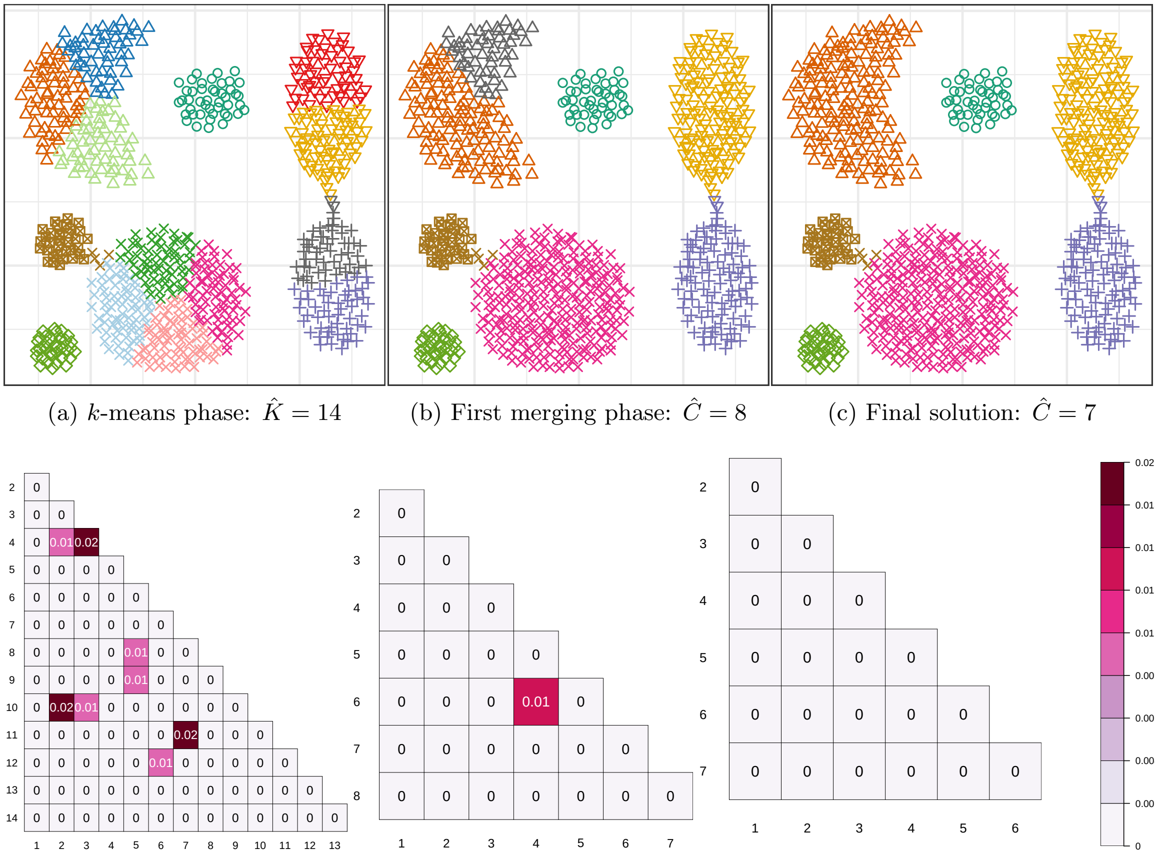
Having proposed our KNOB-SynC algorithm, we now illustrate and evaluate its performance in relation to a host of competing methods.

3. Performance Evaluations

We first illustrate the performance of KNOB-SynC on the 2D Aggregation dataset of Gionis et al. (2007) and then follow with more detailed performance evaluations on a large number of datasets usually used to evaluate competing algorithms in the literature. These datasets range from two to many dimensions. We compare our methods with a wide range of suitors. These rival methods are the syncytial clustering techniques of K-mH (Peterson et al., 2018) using author-supplied R code, MMC (Baudry et al., 2010) as implemented in the R package MixModCombi, DEMP (Hennig, 2010) using the R package fpc and DEMP+ (Melnykov, 2016). We also evaluate performance with EAC (Fred and Jain, 2005) and GSL-NN (Stuetzle and Nugent, 2010) using publicly available author-supplied code. We also apply two common connectivity-based techniques of spectral and kernel k-means clustering. Both these methods need the number of groups to proceed: for spectral clustering we decide this number to be the one with the highest gap in successive eigenvalues of the similarity matrix (von Luxburg, 2007). For kernel k-means, we set K to be the true value: we recognize that our evaluation of kernel k-means potentially provides this method with an unfair advantage, however, we proceed in this fashion in order to understand the best case scenario of this competing method. Finally, we also compare our method's performance relative to DBSCAN* as implemented in the R package dbscan (Hahsler and Piekenbrock, 2018), DP clustering (Rodriguez and Laio, 2014) as implemented in the R package densityClust (Pedersen et al., 2017), PGMM (McNicholas and Murphy, 2008) using the R package pgmm (McNicholas et al., 2018), MSAL using the R package MixSAL (Franczak et al., 2018) and MGHD using the R package MixGHD (Tortora et al., 2019). Many of these algorithms have multiple parameters that need to be set – in our experiments, we use the default settings where guidance for choosing these parameters is not explicitly available. Also, the DBSCAN* and DP clustering algorithms identify scatter/outliers that by definition are those observations that are unlike any other in the dataset, so we follow Maitra and Ramler (2009) in considering them as individual singleton clusters in our performance assessments. Performance for each method is evaluated by Hubert and Arabie (1985)'s adjusted Rand index $\mathcal{R}$ measured between the true partition and the estimated final partitioning. In general, $\mathcal{R} \leq 1$: values closer to 1 indicating greater similarity between partitionings and good clustering performance. The index takes values farther from 1 as performance becomes poorer and is expected to take a value of zero for a random assignmment. The index $\mathcal{R}$ can take arbitrarily negative values, but as very helpfully pointed out to us by a reviewer, the probability of observing $\mathcal{R} < -1$ is relatively small (see Steinley, 2004, for further discussion on the characteristics of this index).

3.1. Illustrative Example: the `Aggregation` dataset

The 2D `Aggregation` dataset (Gionis et al., 2007) has $n = 788$ observations from $C = 7$ groups of different characteristics. Figure 1 displays the results of the different phases and iterations of KNOB-SynC. We display the stages of KNOB-SynC for $\kappa = 1$ which is when we have the lowest terminating $\ddot{\hat{\omega}}$ (from among $\kappa = 1, 2, 3, \infty$) for this example. The k-means phase of our algorithm identifies 14 clusters with partitioning as in Figure 1a and estimates the initial overlap matrix $\hat{\Omega}$ to be as in Figure 1d. The first merging phase yields the partitioning in Figure 1b with the updated $\hat{\Omega}$ of Figure 1e. The next merging phase only combines one pair of groups and is terminal, resulting in the final partitioning of the dataset as in Figure 1c. The overlap matrix (Figure 1f) indicates well-separated clusters,



(a) k-means phase: $\hat{K} = 14$

(b) First merging phase: $\hat{C} = 8$

(c) Final solution: $\hat{C} = 7$

	1	2	3	4	5	6	7	8	9	10	11	12	13
2	0												
3	0	0											
4	0	0.01	0.02										
5	0	0	0	0									
6	0	0	0	0	0								
7	0	0	0	0	0	0							
8	0	0	0	0	0.01	0	0						
9	0	0	0	0	0.01	0	0	0					
10	0	0.02	0.01	0	0	0	0	0	0				
11	0	0	0	0	0	0	0.02	0	0	0			
12	0	0	0	0	0	0.01	0	0	0	0	0		
13	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0

(d) $\ddot{\hat{\omega}} = 0.0013$

	1	2	3	4	5	6	7
2	0						
3	0	0					
4	0	0	0				
5	0	0	0	0			
6	0	0	0	0.01	0		
7	0	0	0	0	0	0	
8	0	0	0	0	0	0	0

(e) $\ddot{\hat{\omega}} = 0.0008$

	1	2	3	4	5	6
2	0					
3	0	0				
4	0	0	0			
5	0	0	0	0		
6	0	0	0	0	0	
7	0	0	0	0	0	0

(f) $\ddot{\hat{\omega}} = 10^{-6}$

Figure 1: Illustrating all three stages of the KNOB-SynC algorithm on the `Aggregation` dataset: Results of (a) the k-means phase, (b) the first merging phase and (c) the second (and final) merging phase of the algorithm. In these and all such subsequent figures, character denotes true class membership while color indicates estimated class membership. (d)–(f) Estimated pairwise nonparametric overlap values corresponding to the partitions in (a), (b) and (c).

with only six mislabeled observations relative to the true, and a $\mathcal{R}$ of 0.98 between the true and estimated classifications.

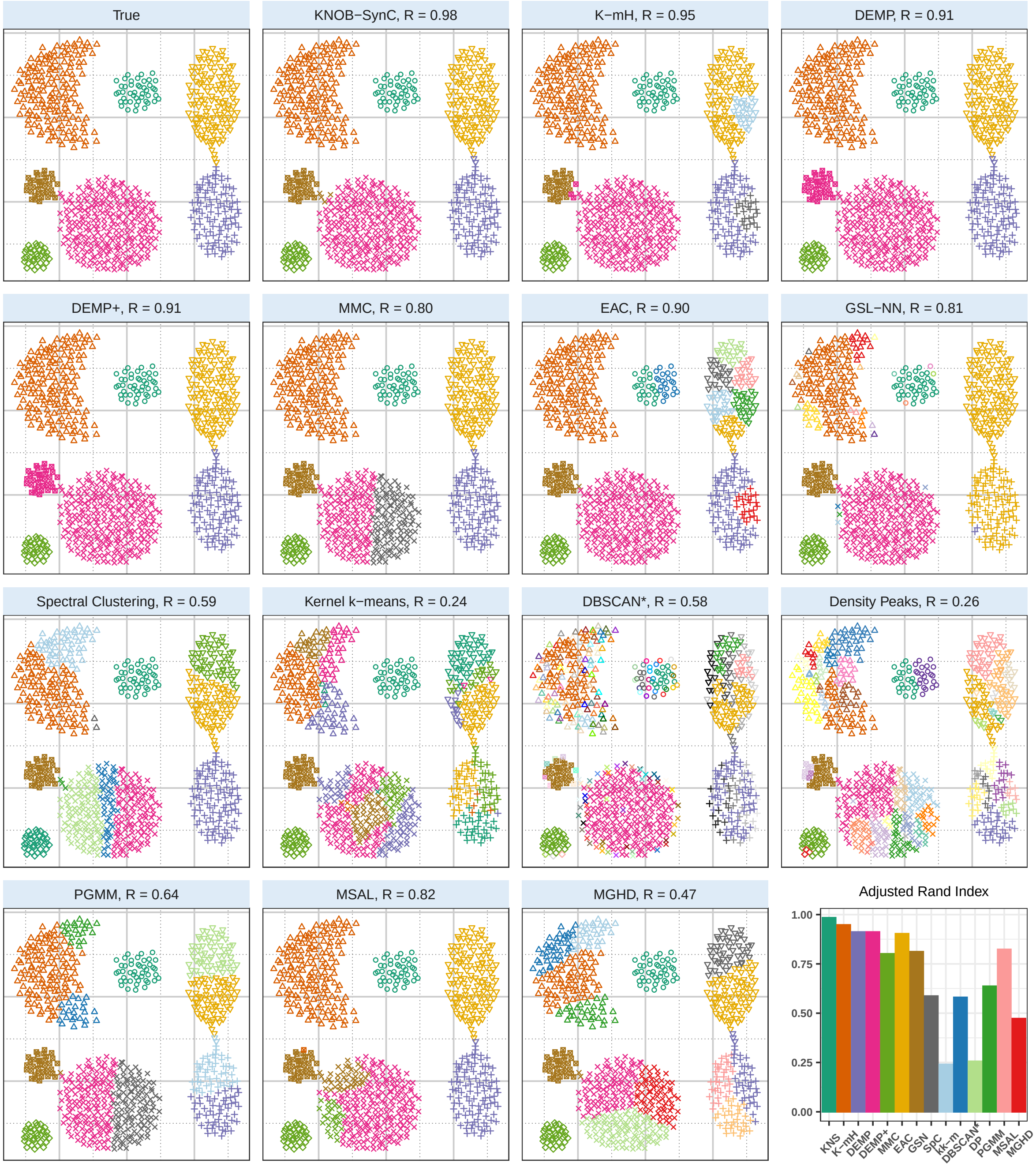


Figure 2: Clustering performance of each algorithm on the `Aggregation` dataset.

The competing methods (Figure 2) all perform marginally to substantially worse. K-mH is the second best performer ($\mathcal{R} = 0.95$) finding $\hat{C} = 9$ groups but breaking the top right cluster into two and also grouping a few other stray observations. Both DEMP and DEMP+ yield the same result ($\mathcal{R} = 0.91, \hat{C} = 6$), but MMC ($\mathcal{R} = 0.8$, $\hat{C} = 8$) has trouble with the largest group, splitting it into two sub-groups. EAC breaks the top central and large groups on the right into many clusters, resulting in $\hat{C} = 14$ but $\mathcal{R} = 0.9$. Thus, in spite of identifying a large number of groups, EAC is able to capture a fair bit of the complex group structure of this dataset. GSL-NN can not distinguish between the groups on the right but also finds many other small groups elsewhere, ending with $\hat{C} = 12$ groups and $\mathcal{R} = 0.81$. The performance of spectral clustering is worse: it finds $\hat{C} = 12$ groups and has a $\mathcal{R} = 0.59$ with the true classification. Despite being provided with the true $C = 7$, kernel k-means with $\mathcal{R} = 0.24$ is the worst performer in this example, with DP ($\mathcal{R} = 0.26$, $\hat{C} = 26$) only marginally better. DBSCAN* at $\mathcal{R} = 0.58$ and $\hat{C} = 277$, correctly finds the large circular group and one of the smallest groups, but the observations in the top left group are almost all classified as outliers/scatter. Among the MBC methods for general-shaped clusters, MSAL at $\mathcal{R} = 0.82$ is the best performer, finding $\hat{C} = 7$ groups but having trouble with the larger group at the bottom. PGMM finds $\hat{C} = 12$ groups ($\mathcal{R} = 0.64$) with the larger groups split further, while the worst-performing MBC method is MGHD ($\mathcal{R} = 0.47$, $\hat{C} = 15$).

3.2. Additional 2D Experiments

3.2.1. Experimental framework

Figure 3 displays the 12 additional 2D datasets used to evaluate performance of KNOB-SynC and its competitors. Barring the first example, all these datasets have been used by other authors to demonstrate and evaluate performance of their methods. The groups in these datasets have structure ranging from the regular (*e.g.*, the 7-spherically-dispersed Gaussian clusters dataset that is modeled on a similar example in Maitra, 2009a, and where sophisticated methods like KNOB-SynC are superfluous and unnecessary) to widely-varying complexity. The `Banana Arcs` dataset has $n = 4515$ observations clumped in four banana-shaped structures arced around each other. The `Banana-clump` and `Bullseye` datasets are from Stuetzle and Nugent (2010) – the former has 200 observations with one spherical group and another arced around it on the left like a banana, while the latter has 400 observations grouped, as its name implies, as a bullseye. The more complex-structured `Bullseye-Cigarette` dataset (Peterson et al., 2018) has three concentric-ringed groups, two elongated groups above two spherical groups on the left, and another group that is actually a superset of two overlapping spherical groups ($n = 3025$ and $C = 8$). The `Compound` dataset (Zahn, 1971) is very complex-structured with $n = 399$ observations in $C = 6$ groups that are not just varied in shape, but a group that sits atop another on the right. The `Half-ringed clusters` dataset (Jain and Law, 2005) has 373 observations in two arc-shaped clusters, one of which is dense and the other being very sparsely-populated. The `Path-based` dataset (Chang and Yeung, 2008) has 300 observations in three groups, two of which are regular-shaped and surrounded by a widely arcing third group. The `Spiral` dataset (Chang and Yeung, 2008) has 312 observations in three spiral groups that are very difficult for standard clustering algorithms to recover accurately. The `SSS` dataset has 5015 observations in three S-shaped groups of varying density and orientations while the `XXXX` dataset has $n = 415$ observations distributed in four cross-shaped structures.

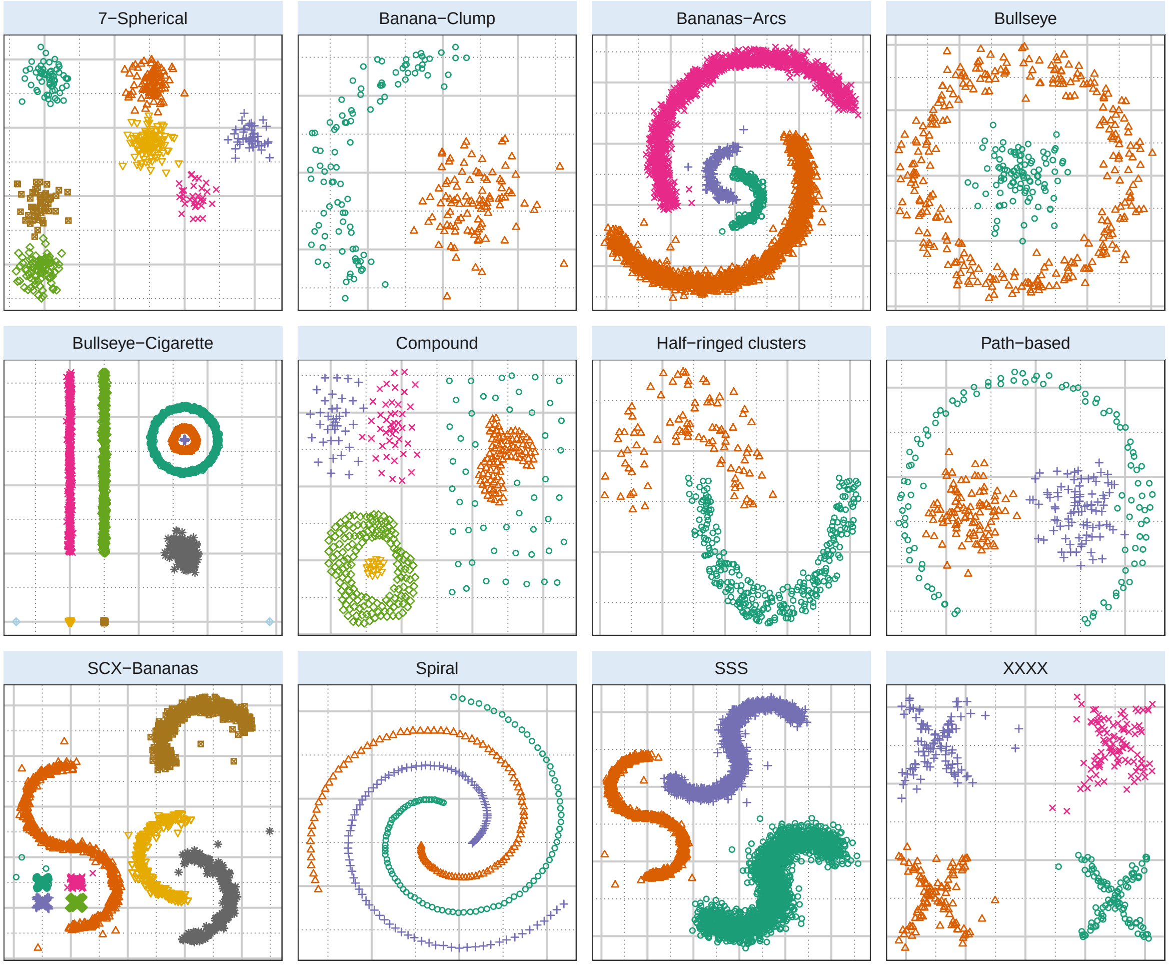
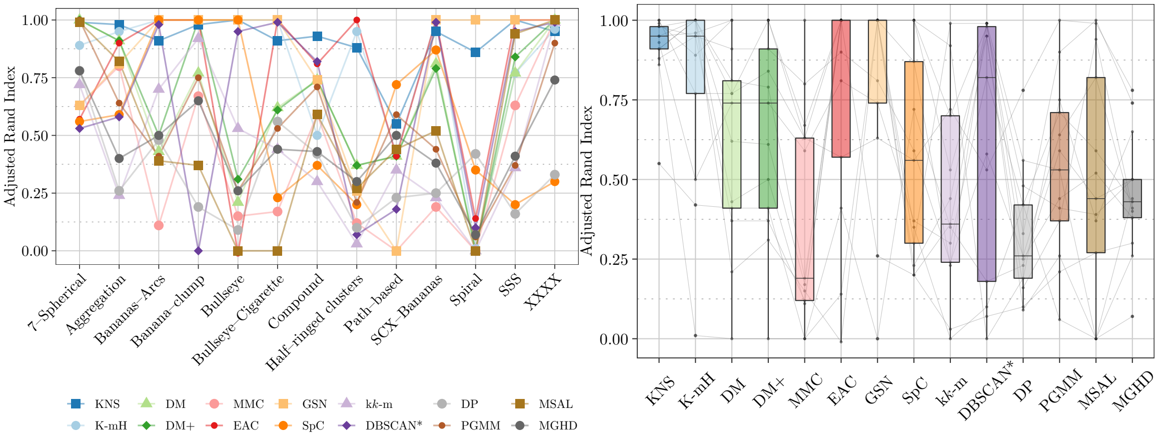


Figure 3: Shape datasets used in the two-dimensional performance evaluations.

3.2.2. Results

Figure 4 and Table 8 summarize the performance of all methods on the 2D experimental datasets. Detailed displays of different methods on individual datasets are in Appendix B. The summaries indicate across-the-board good performance of KNOB-SynC with it always being a top performer. In its worst case, KNOB-SynC gets a $\mathcal{R} = 0.55$ (on the `Path-based` dataset) where it terminates early (Figure 18) but here also it is the fourth-best performer, behind spectral clustering ($\mathcal{R} = 0.72$), MGHD ($\mathcal{R} = 0.60$) and PGMM ($\mathcal{R} = 0.59$). The competing syncytial clustering methods do well in some cases, but not in others where other methods perform better. Among the syncytial clustering methods, K-mH performs better than DEMP, DEMP+ and MMC whose performance can sometimes be poor (*e.g.*, on the `Bullseye`, `Half-ringed clusters` and `Spiral` datasets – vide Figures 14, 17 and 20). It is on these datasets that the other methods (EAC, GSL-NN, and spectral clustering) do better. The performance of kernel-k-means even with known true number of groups is varied, being very good sometimes (*e.g.*, in the `Bananas-clump` dataset of Figure 13) but



(a) Performance by dataset

(b) Performance by method

Figure 4: Performance of KNOB-SynC (abbreviation: KNS), K-mH, DEMP (DM), DEMP+ (DM+), MMC, EAC, GSL-NN (GSN), spectral clustering (SpC), kernel k-means (kk-m), DBSCAN*, DP, PGMM, MSAL and MGHD on 2D datasets.

very poor in other cases (*e.g.*, as seen before in the `Aggregation` dataset) where almost every other method does well. The three general MBC methods perform similarly but PGMM does a bit better than MSAL and MGHD. DBSCAN* and, especially DP, generally perform poorly – however, DBSCAN* performs well in some cases. We quantify performance of each method against its competitors in terms of its average deviation from the best performer. Specifically, for each dataset, we compute the deviation (or difference) in $\mathcal{R}$ of a method from that of the best performer for that dataset. The average deviation ($\bar{\mathcal{D}}$) of a method over all datasets is an overall indicator of its performance. Table 8 provides $\bar{\mathcal{D}}$ and the standard deviation or SD ($\mathcal{D}_\sigma$) of the differences. On the average, KNOB-SynC is the best performer (and with the lowest $\mathcal{D}_\sigma$) followed by GSL-NN, K-mH and EAC. This conclusion of KNOB-SynC's superior overall performance is also supported by Figure 4(b). We surmise that KNOB-SynC does well across the different datasets because of its ability by construction to merge many or few components at a time, with the exact choice of merges and termination objectively selected and determined by the distinctiveness of the resulting partitioning as per $\ddot{\omega}$.

3.3. Higher-Dimensional Datasets

We also study the performance of KNOB-SynC and its competitors on higher-dimensional datasets. These datasets are modest- to higher-dimensional, with between 173 to 10993 records. For the higher-dimensional datasets (*i.e.*, with non-redundant dimension greater than 10), we find that all methods other than GSL-NN generally perform better when used on the first few (m) kernel principal components (KPCs) rather than on the raw data. For these methods and these datasets, we use the first m KPCs of each dataset with m chosen as the first time after which increases in the eigenvalues corresponding to the successive KPCs are below 0.5%. GSL-NN is implemented on the original datasets. Further, KNOB-SynC, K-mH and EAC are built on k-means whose results depend on the scale of the features. So, for these methods, we scaled each feature by the SD prior to analysis unless the features were all collected on a similar scale, as with the *E. coli* example of Section 3.3.2 or the log-transformed GRB dataset of Section 4.1. (Following the usual rule-of-thumb in multivariate statistics, we assume that features are on similar scales if the most variable feature has SD no more than four times that of the feature with lowest variability.) Each dataset and the performance of each method is first described individually. A comprehensive summary of the performance of each method on each dataset follows in Section 3.3.11.

3.3.1. Simplex-7 Gaussian Clusters

This dataset, from Stuetzle and Nugent (2010), is of Gaussian realizations of size 50, 60, 70, 80, 90, 100 and 110 each from seven clusters with means set at the vertices of the seven-dimensional unit simplex and homogeneous spherical dispersions with common SD of 0.25 in each dimension. Like the `7-Spherical` dataset, this dataset exemplifies a case where standard methods such as k-means or Gaussian MBC should be adequate. Therefore it is a test of whether our algorithm and its competitors are able to refrain from identifying spurious complexity. All methods, except for EAC, DBSCAN* and DP, identify seven groups and have good clustering performance. In particular, the syncytial methods have very good performance ($\mathcal{R} = 0.97$); other methods have $\mathcal{R} \in [0.92, 0.98]$. EAC still performs well ($\mathcal{R} = 0.94$) but finds $\hat{C} = 6$ groups. DBSCAN* is the worst performer ($\mathcal{R} = 0.03$) on this dataset, finding many outliers ($\hat{C} = 508$). DP's performance is middling at $\mathcal{R} = 0.58$ and with many outliers ($\hat{C} = 79$).

3.3.2. E. coli Protein Localization

The *E. coli* dataset, publicly available from the University of California Irvine's Machine Learning Repository (UCIMLR) (Newman et al., 1998), concerns identification of protein localization sites for the *E. coli* bacteria (Nakai and Kinehasa, 1991). There are eight protein localization sites: cytoplasm, inner membrane without signal sequence, periplasm, inner membrane with an uncleavable signal sequence, outer membrane, outer membrane lipoprotein, inner membrane lipoprotein, and inner membrane with a cleavable signal sequence. Identifying these sites is an important early step for finding remedies (Nakai and Kinehasa, 1991). Each protein sequence has a number of numerical attributes – see Horton and Nakai (1985) for a listing and their detailed description. Two attributes are binary, but 326 of the 336 sequences have common values for these attributes. We restrict our investigation to these sequences and drop the two binary attributes from our list of variables. These 326 sequences have no representation from the inner membrane or outer membrane lipoproteins. Additionally, we also drop two sequences because they are the lone representatives from the inner membrane with cleavable sequence site (Maitra, 2002). Therefore we have $n = 324$ observations from $C = 5$ true classes. KNOB-SynC identifies $\hat{C} = 9$ groups. Table 1 presents the confusion matrix containing the number of times a protein from a localization site is assigned to each KNOB-SynC group. Ignoring stray assignments, the sites are fairly well-defined in the first four groups, with $\mathcal{R} = 0.72$. Uncleavable signal sequences from the inner membrane site are difficult to distinguish from those that are also from there but have no signal sequence. Sequences from the other sites are better-clarified. Among the alterna-

Table 1: Confusion matrix of the KNOB-SynC groups against the true *E. coli* localization sites.

	1	2	3	4	5	6	7	8	9
cytoplasm	138	0	0	4	0	0	0	0	1
inner membrane, no signal sequence	7	62	0	0	0	3	3	1	0
inner membrane, uncleavable signal sequence	1	32	0	0	0	0	0	1	0
outer membrane	0	0	17	2	0	0	0	0	0
periplasm	3	1	2	38	8	0	0	0	0

tive methods, EAC does slightly better ($\mathcal{R} = 0.77$) but identifies 10 groups. The remaining methods all do slightly to substantially worse. DEMP, DEMP+ and K-mH each identify four groups but with $\mathcal{R} \in [0.63, 0.70]$. K-mH finds only two groups ($\mathcal{R} = 0.41$) while the rest find more groups but disagree more strongly with the true localizations. DBSCAN* finds a large number of groups ($\hat{C} = 174$) with relatively poor performance ($\mathcal{R} = 0.31$). DP is marginally better ($\mathcal{R} = 0.39, \hat{C} = 12$) while PGMM ($\mathcal{R} = 0.48$) and MGHD ($\mathcal{R} = 0.6$) improves on DP, each finding 8 groups. (MSAL did not converge to a solution.) Overall, EAC and KNOB-SynC are the top two performers, with DEMP and DEMP+ close behind.

3.3.3. Standard Wine Recognition

The standard wine recognition dataset (Forina et al., 1988; S. Aeberhard and de Vel, 1992), also available from the UCIMLR contains $p = 13$ measurements on $n = 178$ wine samples that are obtained from its chemical analysis. There are 59, 71 and 48 wines of the Barolo, Grignolino and Barbera cultivars, so $C = 3$. Because $p > 10$ here, we use $m = 17$ KPCs. KNOB-SynC is the best performer, finding $\hat{C} = 3$ groups with a clustering performance of $\mathcal{R} = 0.92$. The first group contains all the 59 wines from the Barola cultivar and 2 Grignolino wines. The second group contains 66 wines, all exclusively from the Grignolino cultivar. The third group has 2 Grignolino and 48 Barbera wines. Thus, there is very good definition among the KNOB-SynC groups. On the other hand, only MMC ($\mathcal{R} = 0.67; \hat{C} = 5$), K-mH ($\mathcal{R} = 0.62, \hat{C} = 6$), PGMM ($\mathcal{R} = 0.66, \hat{C} = 5$) and EAC ($\mathcal{R} = 0.60; \hat{C} = 9$) perform modestly while the others are substantially worse with DBSCAN*, in particular, classifying all observations as outliers, resulting in $\hat{C} = 178$ and $\mathcal{R} = 0$.

3.3.4. Extended Wine Recognition

A reviewer very helpfully pointed out that the dataset used in Section 3.3.3 is actually a reduced variant, and a fuller version of the dataset with 27 variables is available in the R package pgmm. We used $m = 26$ KPCs in our experimental evaluations on this larger dataset. DEMP and DEMP+ ($\mathcal{R} = 1, \hat{C} = 3$) show perfect classification while MGHD ($\mathcal{R} = 0.95, \hat{C} = 3$) and MMC ($\mathcal{R} = 0.91, \hat{C} = 4$) also perform well. KNOB-SynC is a top performer and the best among the distribution-free methods, finding $\hat{C} = 3$ groups and with a clustering performance of $\mathcal{R} = 0.93$. The first group here has 58 Barola and 2 Grignolino wines. The second group contains 68 wines from the Grignolino cultivar and the one Barolo wine that was not placed in the first group. The third group has the 48

Barbera wines and the one remaining Grignolino wine. Similar to the 13-dimensional case, we get good definition among the groups. Other methods not discussed here do moderately to substantially worse.

3.3.5. Olive Oils

The olive oils dataset (Forina and Tiscornia, 1982; Forina et al., 1983) has measurements on 8 chemical components for 572 samples of olive oil taken from 9 different areas in Italy that are from three regions: Sardinia and Northern and Southern Italy. This is an interesting dataset with sub-classes (areas) inside classes (regions). Indeed, Peterson et al. (2018) were able to identify, with one misclassification, sub-groups within the regions but not the areas ($\mathcal{R} = 0.67, \hat{C} = 11$; we however get $\mathcal{R} = 0.56, \hat{C} = 8$ using the authors' supplied code) − they surmised that it may be more possible to identify characteristics of olive oils based on regions defined by physical geography rather than areas demarcated by political geography. We therefore analyze performance on this dataset both in terms of how regions and areas are recovered. KNOB-SynC identifies $\hat{C} = 4$ regions (Table 2) with oils from the Sardinian and Northern regions correctly classified into the first two groups. The Southern region oils are split into our two remaining groups, one containing all but 2 of the 25 North Apulian samples and 6 of the 36 Sicilian samples, and the other group containing all the Southern oils. In terms of clustering performance, KNOB-SynC gets $\mathcal{R} = 0.55$ when compared to the true areal grouping but $\mathcal{R} = 0.87$ when compared to the true regional grouping. For this dataset DEMP ($\mathcal{R} = 0.85; \hat{C} = 7$), DEMP+ ($\mathcal{R} = 0.82; \hat{C} = 12$) and MSAL ($\mathcal{R} = 0.7; \hat{C} = 9$) are the top performers with respect to the true areal grouping. The remaining methods all have middling performance. When compared with the true regional grouping, KNOB-SynC is by far the best performer. Overall, the clustering performance of KNOB-SynC for the regional grouping marginally trumps the performance of DEMP for the areal grouping and so may be considered to be more accurate in uncovering the group structure in the dataset.

Table 2: Confusion matrix of the KNOB-SynC grouping of the Olive Oils dataset.

Region	Area	1	2	3	4
Sardinia	Coast-Sardinia	33	0	0	0
	Inland-Sardinia	65	0	0	0
North	East-Liguria	0	0	50	0
	West-Liguria	0	0	50	0
	Umbria	0	0	51	0
South	Calabria	0	56	0	0
	North-Apulia	0	2	0	23
	South-Apulia	0	206	0	0
	Sicily	0	30	0	6

3.3.6. Image Segmentation

The image segmentation dataset, also available from the UCIMLR, is on 19 attributes of the scene in each 3 × 3 image manually classified to be from BRICKFACE, CEMENT, FOLIAGE, GRASS, PATH, SKY and WINDOW. (Thus, $C = 7$.) We combine the training and test datasets to obtain 330 instances of each scene, so $n = 2310$. There is a lot of redundancy in the attributes so we reduce the dataset to 8 PCs that together explain at least 99.9% of the total variance in the dataset. The PCs are obtained from the correlation matrix because the 19 attributes have vastly different scales. The KNOB-SynC solution finds $\hat{C} = 12$ clusters, with $\mathcal{R} = 0.55$. The confusion matrix (Table 3) indicates that the SKY images are perfectly identified while GRASS and, to a lesser extent, PATH and CEMENT, are fairly well-identified. On the other hand, the partitioning struggles to distinguish between BRICKFACE, FOLIAGE and WINDOW. Among other methods, only EAC ($\mathcal{R} = 0.59, \hat{C} = 40$) and MGHD ($\mathcal{R} = 0.56, \hat{C} = 7$) are modestly to marginally better than KNOB-SynC. Inspection of the EAC grouping indicates many small groups but also difficulty in separating FOLIAGE and WINDOW, placing them together in one group. Further, BRICKFACE is split into five groups, four of which are predominantly of this kind, but the fifth group is unable to distinguish 146 observations of BRICKFACE from 32, 52 and 62 observations of CEMENT, FOLIAGE and WINDOW, respectively. The other methods all perform moderately to substantially worse (Table 9) with PGMM, MSAL and DBSCAN* unable to find clustering solutions.

Table 3: Confusion matrix of the KNOB-SynC grouping against the true for the Image segmentation dataset.

	1	2	3	4	5	6	7	8	9	10	11	12
BRICKFACE	330	0	0	0	0	0	0	0	0	0	0	0
CEMENT	42	257	0	4	0	27	0	0	0	0	0	0
FOLIAGE	300	5	0	0	0	5	2	7	3	1	3	4
GRASS	1	0	327	0	0	2	0	0	0	0	0	0
PATH	0	0	0	269	0	61	0	0	0	0	0	0
SKY	0	0	0	0	330	0	0	0	0	0	0	0
WINDOW	309	13	0	0	0	8	0	0	0	0	0	0

3.3.7. Yeast Protein Localization

The yeast protein localization dataset (Nakai, 1996), also obtained from the UCIMLR, was used by Melnykov (2016) to illustrate the application of DEMP+. This dataset is on the localization of the proteins in yeast into one of $C = 10$ sites and has two attributes (presence of "HDEL" substring and peroxisomal targeting signal in the C-term) that are essentially binary and trinary. Following Melnykov (2016), we drop these variables and use the other $p = 6$ variables, namely signal sequence recognition scores based on (a) McGeoch's and (b) von Heijne's methods, (c) ALOM membrane spanning region prediction score, and discriminant analysis scores of the amino acid content of (d) N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins and (e) vacuolar and extracellular proteins and (f) discriminant scores of nuclear localization signals of nuclear and non-nuclear proteins. For this dataset, all methods perform poorly. KNOB-SynC ($\mathcal{R} = 0.226; \hat{C} = 7$) is the best performer – the other clustering methods essentially randomly allocate observations. Surprisingly, DEMP+ ($\hat{C} = 6$, $\mathcal{R} = -0.01$) performs very poorly.

(Melnykov, 2016, only used the first five of our variables to illustrate the DEMP+ method: we find no appreciable improvement even then, with $\hat{C} = 7$ and $\mathcal{R} = -0.009$. Personal queries to the author did not successfully resolve this discrepancy.) It appears therefore that the yeast protein localization dataset may be difficult to accurately partition in a completely unsupervised framework.

3.3.8. Acute Lymphoblastic Leukemia

The Acute Lymphoblastic Leukemia (ALL) training dataset of Yeoh et al. (2002) was used by Stuetzle and Nugent (2010) to illustrate GSL-NN in a high-dimensional small sample size framework. We use the standardized dataset in Stuetzle and Nugent (2010) that measured the oligonucleotide expression levels of the 1000 highest-varying genes in 215 patients suffering from one of seven leukemia subtypes, namely, T-ALL, E2A-PBX1, BCR-ABL, TEL-AML1, MLL rearrangement, Hyperploid > 50 chromosomes, or an unknown category labeled OTHER. Some subtypes have very few cases: for instance, only 9, 14 and 18 patients are of type BCR-ABL, MLL and E2A-PBX1, respectively. For this dataset, we use $m = 42$ KPCs for all methods but GSL-NN. The k-means stage of KNOB-SynC identifies six groups, none of which are merged in the merging phase, resulting in the best partitioning among all competing methods. Table 4 presents the confusion matrix containing the number of cases a patient of a leukemia subtype was assigned to a KNOB-SynC group. We see that most leukemia subtypes are distinctively identified in the KNOB-SynC solution. The alternative methods perform mildly to substantially worse with PGMM, spectral clustering, GSL-NN and DP having clustering solutions ($\mathcal{R} = 0.61, 0.55, 0.54, 0.53$) that are the next best after KNOB-SynC. Other methods generally do poorly, with MSAL and MGHD unable to find solutions while DBSCAN* classifies all observations as outliers.

Table 4: Confusion matrix of the KNOB-SynC grouping against the true leukemia subtypes for the ALL dataset.

	1	2	3	4	5	6
Hyperdiploid>50	0	0	2	35	5	0
E2A-PBX1	0	17	0	0	1	0
BCR-ABL	0	0	2	1	6	0
TEL-AML1	4	2	8	2	36	0
MLL	0	3	10	0	1	0
T-ALL	0	0	1	0	0	27
OTHER	51	0	0	0	1	0

3.3.9. Zipcode images

The zipcode images (Stuetzle and Nugent, 2010) dataset consists of $n = 2000$ 16×16 images of handwritten Hindu-Arabic numerals and is our second higher-dimensional example. As in the ALL dataset of Section 3.3.8, we normalize the observations to have zero mean and unit variance so that the Euclidean distance between any two normalized images is negatively and linearly related to the correlation between their pixels. We extract and use the first

$m = 33$ KPCs for all algorithms but GSL-NN. KNOB-SynC identifies 9 groups and has the best clustering performance ($\mathcal{R} = 0.76$). DP is the second best ($\mathcal{R} = 0.58$) performer but finds $\hat{C} = 53$ groups (including singletons) followed by MGHD ($\mathcal{R} = 0.56, \hat{C} = 6$), K-mH ($\mathcal{R} = 0.55, \hat{C} = 22$), GSL-NN and spectral clustering (both with $\mathcal{R} = 0.54$ but $\hat{C} = 7$ and 23). The other methods all perform moderately to substantially worse. Figure 5 displays the 9 KNOB-SynC groups. While misclassifications abound in almost all groups, there is good agreement with 0, 1, 2, the leaner 8s and, (to a lesser extent) 3 and 6, largely correctly identified. The digit 2 is placed in two groups, of the leaner and the rounded versions. The group where 3 predominates also has some 5s and 8s but the categorization makes visual sense. Another group is composed largely of 4s, 7s and 9s but that placement also appears visually explainable. Clearer and straighter 7s and 9s are placed in a separate group. Our partitioning finds it harder to distinguish between 5 and 6 but here also the commonality of the strokes in the digits assigned to this group explains this categorization. Thus we see that KNOB-SynC is not only the best performer for this dataset but also provides interpretable results. We comment that our application of all methods to this dataset has been entirely

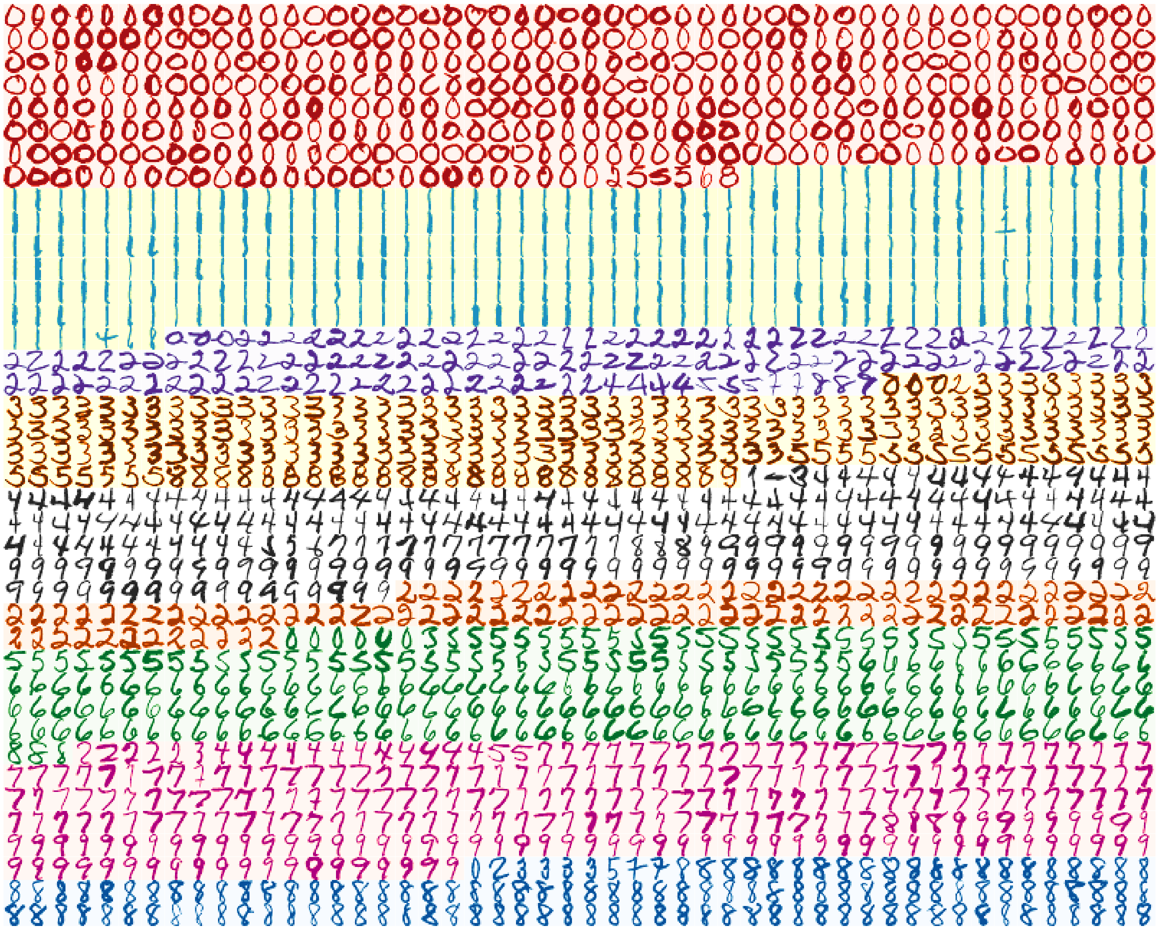


Figure 5: KNOB-SynC groups, with colormap indicating group, of the Zipcode dataset.

unsupervised: methodologies that also account for spatial context and pixel neighborhood may further improve the grouping but are outside the purview of this paper.

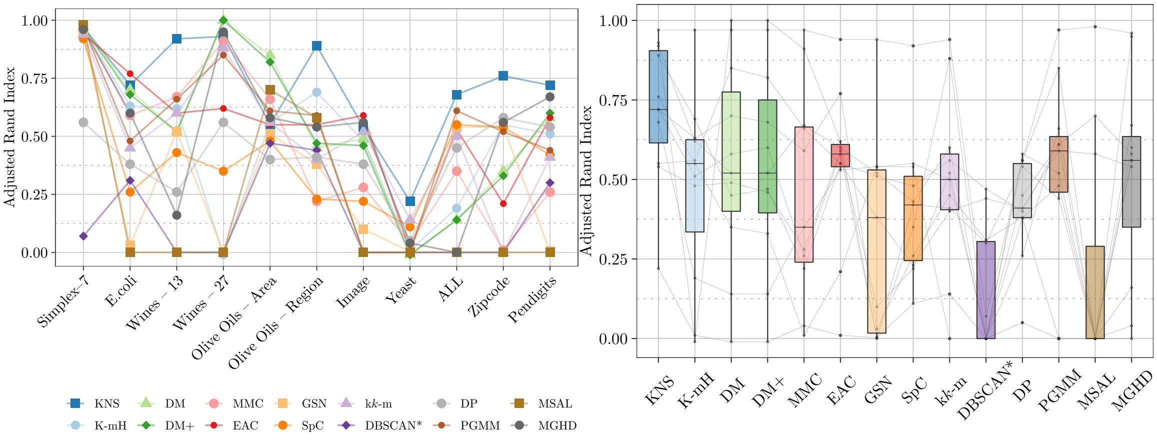
3.3.10. Handwritten Pen-digits

The Handwritten Pen-digits dataset (Alimoglu, 1996; Alimoglu and Alpaydin, 1996) available at the UCIMLR is a larger dataset that has 16 attributes from 250 handwritten samples of 30 writers. (There are $n = 10992$ records because eight samples are unavailable.) We use $m = 18$ KPCs in our analysis (Peterson et al., 2018, used the first 7 PCs and got $\mathcal{R} = 0.64$ and $\hat{C} = 24$). KNOB-SynC finds $\hat{C} = 15$ groups and is the best performer ($\mathcal{R} = 0.723$). It separates the digits 0, 2, 3, 4, 6 and, to a lesser extent, 7 fairly well but identifying 1, 5 and 9 is a bit more challenging (Table 5). It also identifies multiple types of 8. MGHD finds the correct number and is the next-best performer ($\mathcal{R} = 0.67$). The other methods perform moderately to substantially worse with MSAL unable to find a clustering solution.

Table 5: Confusion matrix for the Handwritten Pen-digits dataset.

	0	1	2	3	4	5	6	7	8	9	11	12	13	14	15
0	1099	1	0	0	19	0	21	0	0	2	0	1	0	0	0
1	0	657	358	34	1	0	2	2	0	89	0	0	0	0	0
2	0	2	1141	0	0	0	0	0	0	1	0	0	0	0	0
3	0	4	2	1046	1	0	0	0	0	2	0	0	0	0	0
4	0	5	1	2	1118	0	1	0	0	17	0	0	0	0	0
5	0	1	0	252	0	625	0	0	2	175	0	0	0	0	0
6	0	0	1	0	0	1	1054	0	0	0	0	0	0	0	0
7	0	144	5	2	0	0	0	914	0	0	0	0	77	0	0
8	4	0	0	3	0	1	0	1	461	0	139	321	48	24	53
9	24	9	0	72	3	0	0	0	1	714	0	0	0	232	0

3.3.11. Summary of Performance



(a) Performance by dataset

(b) Performance by method

Figure 6: Overall performance of all competing methods on all higher-dimensional datasets. Abbreviations are as in Figure 4.

Figure 6 and Table 9 summarize performance of all methods on the higher-dimensional experiments. As in the 2D case, KNOB-SynC is almost always among the top performers for high-dimensional datasets. Indeed, KNOB-SynC has the lowest average difference in $\mathcal{R}$ from that of the best-performing method over all datasets (Table 9). The other methods generally perform worse, with EAC, PGMM and kernel-k-means (with true number of groups) among the better ones. Thus, the results of our experiments on real and synthetic datasets indicate good performance of KNOB-SynC relative to its competitors.

3.4. Extensions of KNOB-SynC

As indicated in Section 2, the development of our syncytial clustering methodology is based on the nonparametric estimation of the CDF of the residuals and so can be applied to other scenarios. We explore performance of our methodology in two such settings.

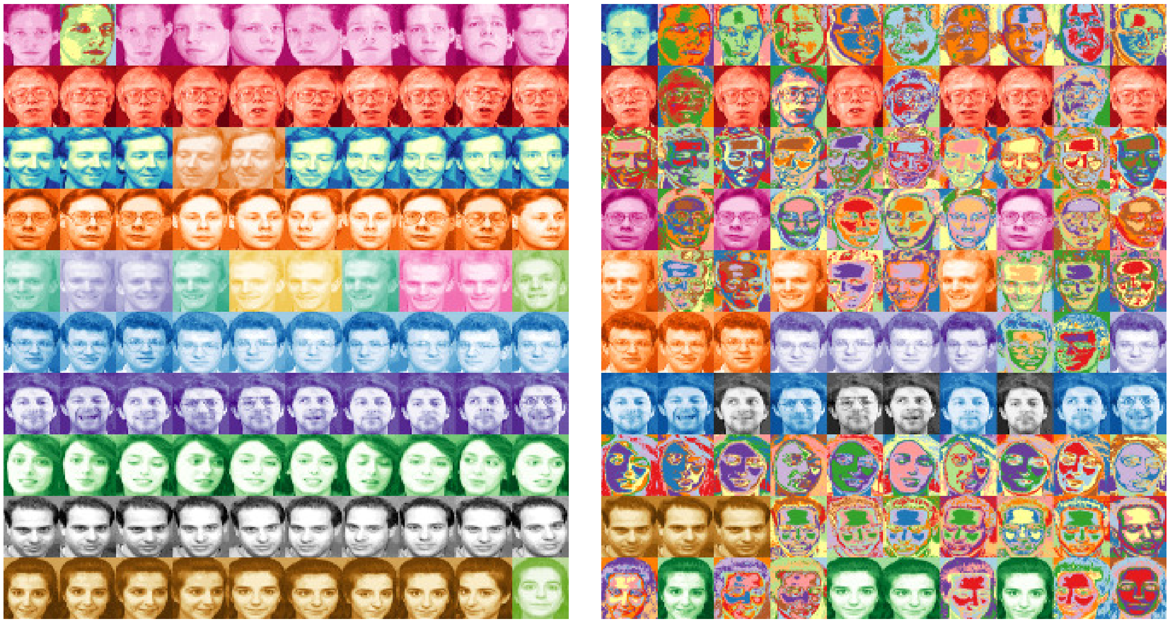
3.4.1. KNOB-SynC in the presence of scatter

Maitra and Ramler (2009) provided the k-clips algorithm for k-means clustering in the presence of scatter, or observations that are unlike any other in the dataset. Our KNOB-SynC methodology and software readily incorporates k-clips results by replacing the k-means phase with that algorithm, and proceeding by including the scatter points as individual singleton clusters. We illustrate our methodology on the first 100 images of the Olivetti faces database (Samaria and Harter, 1994) that were used by Rodriguez and Laio (2014) to illustrate their DP algorithm. The 100 images under our consideration are of 10 faces each of 10 individuals taken at different angles and under different light conditions. Therefore, each individual can be considered to be a group with members that are that person's 10 images. Each 112×92 image has a total of 10,304 pixels so we use the first 37 KPCs. While this application does not have any true scatter points, we use this application to illustrate KNOB-SynC with k-clips because it was used by Rodriguez and Laio (2014) to showcase DP that finds scatter (outliers, in their parlance) in addition to clusters.

The k-clips algorithm with the default Bayesian Information Criterion (BIC) (Schwarz, 1978) finds only two well-defined homogeneous spherical clusters and 68 scatter points. We use the trace of the within-sums-of-squares-and-products matrix, rather than its determinant (Maitra and Ramler, 2009), in our objective function in order to satisfy the condition of homogenous spherical clusters around which our base KNOB-SynC algorithm is built. Thus, we have a total of 70 initial groups. KNOB-SynC's merging phase ends with 9 large groups, 5 small groups and 1 scatter observation (so $\hat{C} = 16$) and $\mathcal{R} = 0.902$. The results are displayed in Table 6 and Figure 7 – for comparison, the latter also displays the results reported in Rodriguez and Laio (2014) which found 9 clusters and 62 scatter points, resulting in $\hat{C} = 71$ and $\mathcal{R} = 0.22$. (The figure displays images assigned to a group by means of a distinctive sequential palette. Because there are not enough colors to also identify each scatter point with its individual sequential palette, we use an individual randomized nominal palette for each scatter assignment.) KNOB-SynC identifies images from six individuals (Persons 2, 4, 6, 7, 8 and 9) perfectly and the first, third and the tenth individuals nearly so. The fifth individual is characterized into 5 smaller groups that includes the case where one image is grouped together with the one misclassified image of the tenth person. The performance of our algorithm overwhelms that reported in Rodriguez and Laio (2014).

Table 6: Confusion matrix of the KNOB-SynC results for the Olivetti faces dataset.

	Assigned Groups															
Individual	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	8	0	0	0	0	0	0	0	2	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	3	0	0	0	0	0	0	2	0	2	1	2
6	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	9	0	0	0	0	1	0



(a) KNOB-SynC: $\mathcal{R} = 0.90$

(b) DP: $\mathcal{R} = 0.22$

Figure 7: Clusters of the first 100 images in the Olivetti database obtained by (a) KNOB-SynC and (b) DP as reported in Rodriguez and Laio (2014). Each group is represented by its own distinctive sequential palette. Scatter observations (*i.e.* singleton groups) are represented by individual randomized nominal palettes.

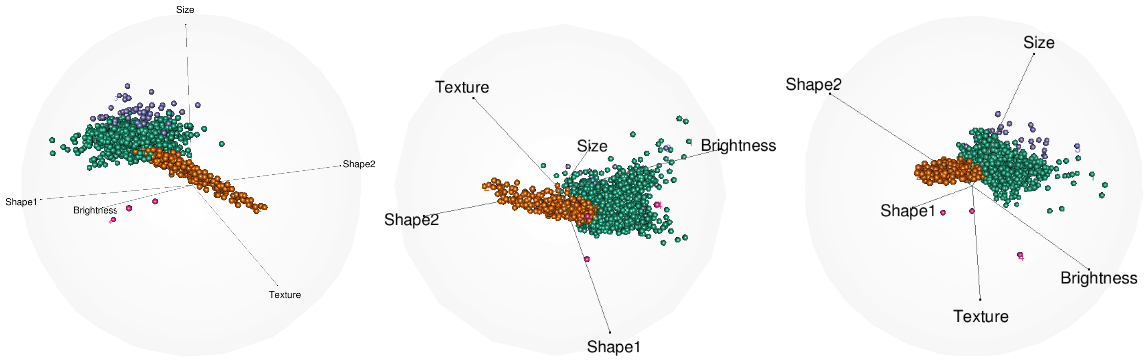
We note that we used the first 37 KPCs with our KNOB-SynC algorithm while Rodriguez and Laio (2014) used the original images with similarity metric as in Sampat et al. (2009). Using DP ($\hat{C} = 83, \mathcal{R} = 0.06$) or DBSCAN* ($\hat{C} = 100, \mathcal{R} = 0$) with Euclidean similarity on the 37 KPCs gave us worse results.

3.4.2. KNOB-SynC with incomplete records

We now illustrate a scenario where KNOB-SynC is applied to a dataset with incomplete records. In this example, we replace the k-means phase with Lithio and Maitra (2018)'s k_m-means algorithm that modifies k-means to account for incomplete records. The authors also develop a modified jump statistic to select the number of groups. The k_m-means results are input into the merging phase of KNOB-SynC and the algorithm proceeds as usual.

We illustrate our methodology on a subset (Wagstaff, 2004) of the Sloan Digital Sky Survey (SDSS) dataset that measures five features (brightness, in psfCounts, size in petroRads, texture, and two measures of shape (M_e1 and M_e2 that we refer to as Shape1 and Shape2 in our analysis) on 1220 galaxies and 287 stars. Thus the true $C = 2$ and $n = 1507$. The dataset has some missing values for the shape measures of 42 galaxies.

The k_m-means algorithm with the modified jump statistic of Lithio and Maitra (2018) finds $K_0 = 46$ homogeneous spherically-dispersed groups. The initial overlap calculations of Step 2 of our algorithm yield $\ddot{\hat{\omega}} = 0.0297$ and $\hat{\check{\omega}} = 0.381$. The merging phase is triggered, and terminates with $\hat{C} = 4$ groups. Figure 8 provides a 3D radial visualization (Zhu et al., 2019) of the clustering results and a confusion matrix of the obtained grouping vis-a-vis the true classification. We see that KNOB-SynC groups all the 287 stars together, but also includes 2 galaxies. The remaining galaxies are all partitioned into groups of 1159, 56 and 3 observations. The large galaxy group and the group with stars are all well-separated from the ones in the smaller galaxy groups. The second-largest KNOB-SynC galaxy group has larger-sized galaxies while the three galaxies in the last group have larger Shape1 and brightness. This illustration demonstrates KNOB-SynC's ability to identify general-shaped



	KNOB-SynC Groups			
	1	2	3	4
Galaxies	1159	2	56	3
Stars	0	287	0	0

Figure 8: (Top) Three views of 3D radial visualization displays of the KNOB-SynC groups found in the SDSS dataset. Only the completely observed records are displayed in the figures. (Bottom) Confusion matrix between the true classifications of Galaxies and Stars with the KNOB-SynC grouping that yielded $\mathcal{R} = 0.86$.

clusters even in the presence of incomplete records. We note that some of the competing methods such as K-mH or EAC may be modified to incorporate k_m-means results but such modifications to both the methodology and software is outside the scope of this paper.

Our experimental evaluations comprehensively demonstrate that our KNOB-SynC algorithm works very well in finding general-shaped clusters. Indeed, our methodology can also incorporate scenarios that allow for scatter or incomplete records in the dataset.

4. Real-world applications

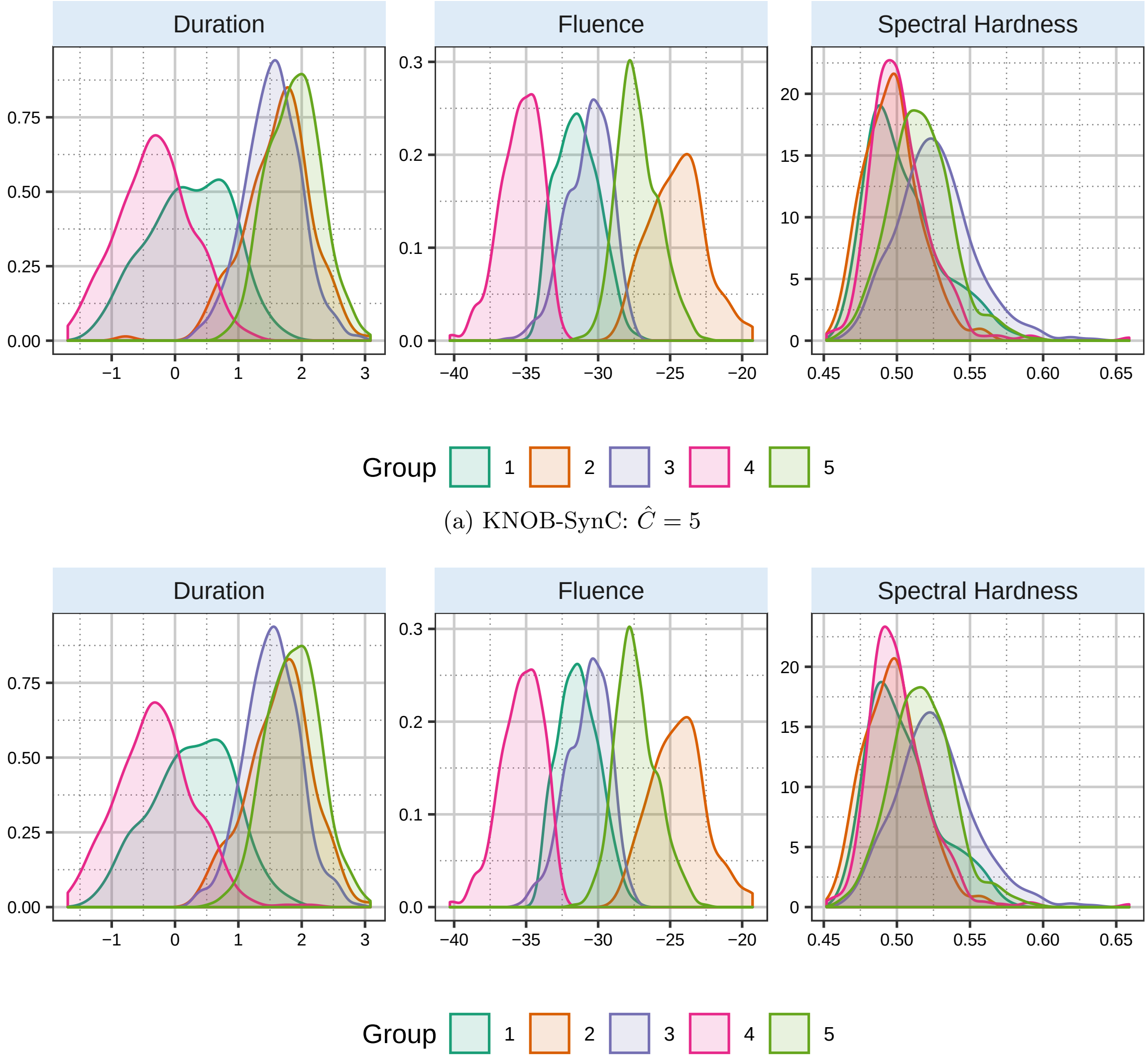
In this section we apply KNOB-SynC to first find the different kinds of Gamma Ray Bursts (GRBs) in an astronomy catalog and second, to identify activation detected in fMRI experiments. The ground truth is unknown in both these applications, so we compare our results with other available evidence in the literature.

4.1. Determining the distinct kinds of Gamma Ray Bursts

There is tremendous interest in understanding the source and nature of Gamma Ray Bursts (GRBs) that are the brightest electromagnetic events known to occur in space (Chattopadhyay et al., 2007; Piran, 2005). Many researchers (Mazets et al., 1981; Norris et al., 1984; Dezalay et al., 1992) have hypothesized that GRBs are of several kinds, but the exact number and descriptive properties of these groups is an area of active research and investigation. Most analyses have traditionally focused on univariate and bivariate statistical and descriptive methods for classification and found two groups but other authors (Mukherjee et al., 1998; Chattopadhyay et al., 2007) have found three different kinds of GRBs when using more variables in the clustering. Recent careful analyses (Chattopadhyay and Maitra, 2017, 2018) has conclusively established five ellipsoidally-shaped groups in the GRB dataset obtained from the BATSE 4Br catalog. Indeed, Chattopadhyay and Maitra (2018) established that all nine fields of the BATSE 4Br catalog have important clustering information using methods developed in Raftery and Dean (2006). These nine fields are the two duration variables (time by which 50% and 90% of the flux arrive), the four time-integrated fluences in the 20-50, 50-100, 100-300, and > 300 keV spectral channels, and the (three) measurements on peak fluxes in time bins of 64, 256 and 1024 milliseconds. The authors used multivariate t-mixtures MBC on the logarithm of the measurements, and BIC for model selection, to arrive at their result of five ellipsoidally-shaped groups.

GRB datasets have typically been analyzed after using a $\log_{10}$ transformation to remove skewness in the dataset. This summary transformation is somewhat arbitrary so Berry and Maitra (2019) used their Transformation-infused K-means (TiK-means) algorithm to alternately transform features and cluster skewed datasets. A modification (Berry and Maitra, 2019) of the jump statistic that accounts for the use of transformations in the algorithm found five groups that were characterized as long-intermediate-intermediate, short-faint-intermediate, short-faint-soft, long-bright-hard and long-intermediate-hard in terms of their duration (T_{90}), total fluence (F_{total}) and spectral hardness (H_{321}) which are the summaries used to characterize GRB groups (Mukherjee et al., 1998).

The fields of the BATSE 4Br catalog are heavily correlated in the log-scale. This has led many researchers to argue for and summarily ignore all but a few variables in their analysis. Here we explore performance using KNOB-SynC on the first three PCs (accounting for



(a) KNOB-SynC: $\hat{C} = 5$

(b) Tik-means: $\hat{C} = 5$

Figure 9: Summary of duration (T_{90}), total fluence (F_{total}) and spectral hardness (H_{321}) for the groups obtained using (a) KNOB-SynC with the generalized Mahalanobis distance and (b) TiK-means.

96.27% of the total variance) of the nine scaled log-transformed variables which is equivalent to using KNOB-SynC with the generalized Mahalanobis distance. The k-means phase applied on the 3 PCs finds 5 groups. KNOB-SynC does not enter the merging stage at all since the maximum and generalized overlaps are the same. Comparison of our results (Figure 9 and Table 7) with those of Berry and Maitra (2019) shows fairly good agreement in the confusion matrix ($\mathcal{R} = 0.868$). The first and the fourth groups have short burst durations (T_{90}) and soft spectral hardness (H_{321}) although the first group has fainter total fluence (F_{total}). Groups 2 and 3 have long durations and bright fluences but different spectral hardness. Group 5 has long duration GRBs but with intermediate fluence and spectral hardness. The true number and kinds of GRB groups is not known but our results show that the KNOB-SynC solution yields groups that are distinct, interpretable and in line with the newer results obtained by TiK-means (Berry and Maitra, 2019) or MBC (Chattopadhyay and Maitra, 2017, 2018).

Table 7: (a) Summary of duration (T_{90}), fluence (F_{total}) and spectral hardness (H_{321}) for each of the five groups obtained using KNOB-SynC with the generalized Mahalanobis distance and TiK-means. (b) Confusion matrix of the TiK-means and KNOB-SynC solutions ($\mathcal{R} = 0.868$).

(a)

	k	n_k	T_{90}	F_{total}	H_{321}	$T_{90} - F_{total} - H_{321}$
KNOB-SynC	1	207	0.234 ± 0.003	-31.423 ± 0.007	0.505 ± 10^{-4}	short-intermediate-soft
	2	187	1.619 ± 0.003	-24.492 ± 0.01	0.497 ± 10^{-4}	long-bright-soft
	3	459	1.543 ± 0.001	-30.682 ± 0.003	0.526 ± 10^{-4}	long-intermediate-hard
	4	318	-0.32 ± 0.002	-35.381 ± 0.004	0.503 ± 10^{-4}	short-faint-soft
	5	428	1.882 ± 0.001	-27.235 ± 0.003	0.516 ± 10^{-4}	long-bright-hard
TiK-means	1	197	0.272 ± 0.003	-31.349 ± 0.007	0.505 ± 10^{-4}	short-intermediate-soft
	2	188	1.65 ± 0.003	-24.439 ± 0.01	0.498 ± 10^{-4}	long-bright-soft
	3	429	1.531 ± 0.001	-30.727 ± 0.003	0.526 ± 10^{-4}	long-intermediate-hard
	4	333	-0.304 ± 0.002	-35.302 ± 0.004	0.502 ± 10^{-4}	short-faint-soft
	5	452	1.867 ± 0.001	-27.362 ± 0.003	0.517 ± 10^{-4}	long-bright-hard

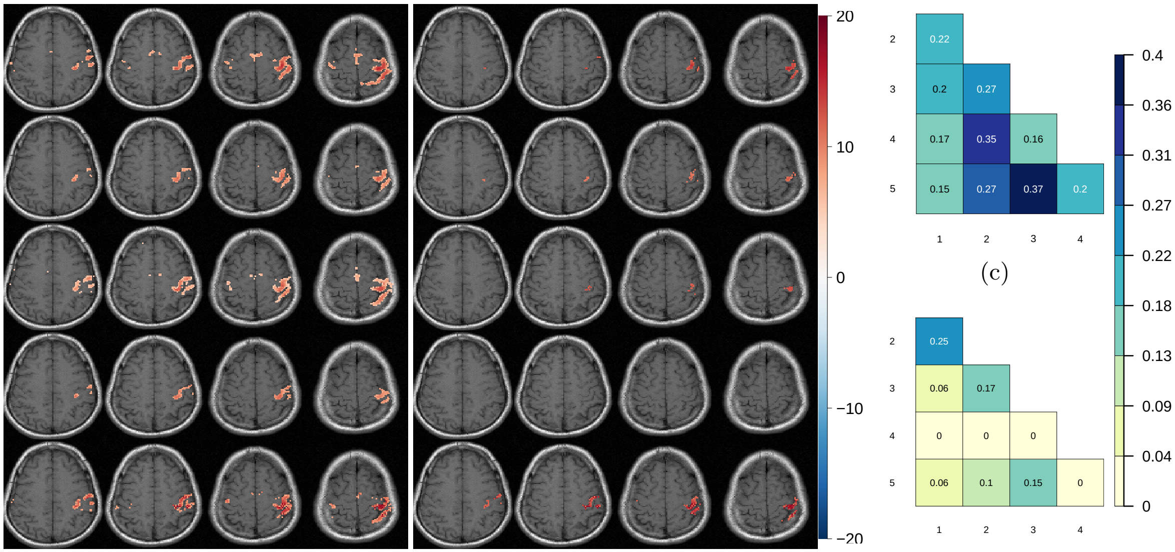
(b)

		KNOB-SynC				
	k	1	2	3	4	5
Tik-means	1	188	2	2	3	2
	2	0	181	0	0	7
	3	1	0	420	2	6
	4	13	0	7	313	0
	5	5	4	30	0	413

4.2. Activation detection in a fMRI finger-tapping task experiment

Our second application uses KNOB-SynC to identify activation in fMRI experiments. One objective of fMRI is to determine cerebral regions that respond to a task or particular stimulus (Bandettini et al., 1993; Belliveau et al., 1991; Kwong et al., 1992; Ogawa et al., 1990). A typical approach relates, after correction and pre-processing, the observed Blood Oxygen Level Dependent (BOLD) time course sequence at each image voxel to the expected BOLD response (Friston et al., 1994; Glover, 1999; Lazar, 2008) by fitting a general linear model (Friston et al., 1995) and obtaining a test statistic (often a t-statistic) that tests for significance at that voxel. Thresholding methods (Forman et al., 1995; Genovese et al., 2002) are often used on these t-statistics to determine activation. Attempts to use clustering algorithms have been made, but Thirion et al. (2014) found that despite the advantages of speed and simplicity, k-means is not, in general, a good performer because it fits "data idiosyncracies" and pathologies. We therefore explore if KNOB-SynC can improve the k-means clustering solution on these datasets.

Our dataset for this experiment is from a right-hand finger-tapping experiment of a right-hand-dominant male and was acquired over twelve regularly-spaced sessions in a two-month span. We choose only 5 of these sessions that were identified in Maitra (2010) as the ones with the highest reliability. Because there is no known gold standard, our comparison here will be of the five partitionings detected in each replication with each other. Each dataset was preprocessed and voxel-wise Z-scores were obtained that quantified the test statistic under the hypothesis of no activation at each voxel. We refer to Maitra et al. (2002) and Maitra (2009b) for imaging details. At each of the $n = 179364$ voxels, we compute the Z-scores to test the hypothesis that the expected BOLD levels are significantly related to the right-hand tapping at a voxel. These Z-scores for each replication are our (one-dimensional) dataset. Because of the large size of the dataset, most competing methods are impractical to apply, so we only use KNOB-SynC here. (For computational reasons also, we do not estimate K_0 in the k-means phase but set it at $K_0 = 50$.)



(a) KNOB-SynC (b) AR-FAST (c) (d)

Figure 10: Observed Z-scores at the voxels in the smaller (activated) group for the right-hand finger-thumb opposition task experiments obtained using (a) KNOB-SynC and (b) AR-FAST. For each set of experiments, we display activation maps for the 18th, 19th, 20th and 21st slices in each column. The replications are represented by rows. Jaccard indices of activation between each pair of replicates using (c) KNOB-SynC and (d) AR-FAST algorithms.

The 50 homogeneous k-means groups in each of the five replicates when supplied to the merging phase each terminated with $\hat{C} = 2$ syncytial groups. For the first replicate, the largest group has 178307 (99.4%) voxels – this is essentially the region of no activation. The other replicates have 178898 (99.7%), 178129 (99.3%), 179087 (99.8%), and 178658 (99.6%) voxels in this group. Figure 10a displays the Z-scores at the activated voxels over four slices of the brain. The displayed slices comprise the ipsi- and contra-lateral pre-motor cortices (pre-M1), the primary motor cortex (M1), the pre-supplementary motor cortex (pre-SMA), and the supplementary motor cortex (SMA). We see broad agreement between the replications in each of the four slices. We compare our KNOB-SynC results with the robust adaptive smoothed thresholding (AR-FAST) algorithm of Almodóvar-Rivera and Maitra (2019) implemented by the `R` package `RFASTfMRI` (Almodóvar-Rivera and Maitra, 2019) which shows far less agreement among the 5 replicates in terms of detected activation in the four slices. Specifically, KNOB-SynC (Figure 10a) identifies activation in the left M1 and in the ipsi-lateral pre-M1 areas. There is some identified activation in the contra-lateral pre-M1, pre-SMA and SMA voxels. On the other hand, AR-FAST (Figure 10b) finds less activation in the left M1 and in the ipsi-lateral pre-M1 areas. Figure 10c displays the Jaccard (1901) index of the activation detected (using KNOB-SynC) between each pair of replications. Figure 10d displays similar Jaccard index calculations with regard to activation detected using AR-FAST. The Jaccard indices are higher for KNOB-SynC-found activation for each pair of replications and show greater reproducibility. The summarized Jaccard index of Maitra (2010) which provides an overall measure of reproducibility of activation detected across replicates is 0.238 for KNOB-SynC and 0.102 for AR-FAST which was shown (Almodóvar-Rivera and Maitra, 2019) to be a top performer on this dataset. We comment that while the overall Jaccard indices are low for both methods, the low value of 0.238 also reflects the challenge of activation detection in single-subject fMRI. Seen in this context, KNOB-SynC does quite well. This example illustrates the potential of KNOB-SynC to improve and refine clustering solutions making it possible, for instance, to use k-means and to alleviate some of the concerns raised in Thirion et al. (2014).

5. Discussion

This paper has proposed a syncytial clustering algorithm called KNOB-SynC that merges groups found by standard clustering algorithms such as k-means, and does so in a data-driven and fully objective way. A R package called SynClustR implements our method in the function `KNOBSynC` and the competing K-mH syncytial algorithm in the function `kmH` and is publicly available at `https://github.com/ialmodovar/SynClustR`. Our method is distribution-free and can apply to the results of many standard clustering algorithms. We use the overlap measure of Maitra and Melnykov (2010) for merging and for decisions but use kernel-based nonparametric methods to calculate this overlap. Our algorithm has no parameters that require fine-tuning by the user and, as pointed out by a reviewer, shows robust performance across many datasets of many dimensions and with little to tremendous complexity, when compared against a host of other methods. Further, our methodology is general enough to extend to situations with incomplete records or where clustering is done in the presence of scatter. Application of KNOB-SynC to data from the BATSE 4Br catalog provides further evidence of five kinds of GRBs. Our approach is also demonstrated to potentially make it possible to adapt k-means clustering for activation detection in fMRI.

This paper also developed estimation methods of the CDF using the asymmetric RIG kernel. We used the plugin-bandwidth selector that minimizes the MISE as our bandwidth choice but it would be good to develop and investigate more sophisticated approaches. Further, our development in this paper provides an opportunity to develop nonparametric methods for diagnostics in clustering. For instance, our developed kernel CDF estimator could be used to determine uncertainties in k-means classifications. A reviewer has also very kindly drawn our attention to the fact that the construction of composite clusters that underlies the idea behind syncytial clustering has also been used in the context of semi-supervised clustering (Śmieja and Wiercioch, 2017) where, instead of estimated overlap as used in this paper, available class labels are used in deciding to merge pairs of groups. We believe that such an approach may also benefit from our methodology, especially when not all classes have representation in the supervised portion of the dataset. Thus, we see that although we have made an important contribution, a number of issues remain that would benefit from further attention.

Acknowledgments

The authors sincerely thank the Action Editor and three anonymous reviewers whose helpful and insightful comments on an earlier version of this article greatly improved its content. The authors also acknowledge partial support for this research from the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institutes of Health (NIH) under its Award No. R21EB016212, I. Almodóvar-Rivera also acknowledges receipt of a fellowship from Iowa State University's Alliance for Graduate Education and the Professoriate (AGEP) program for underrepresented graduate students in STEM fields. R. Maitra also acknowledges support from the United States Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA) Hatch project IOW03617. The content of this paper however is solely the responsibility of the authors and does not represent the official views of either the NIBIB, the NIH, the NIFA or the USDA.

Appendix A.

A.1. Proof of Lemma 3

Proof We have

$$
\begin{aligned}
\mathbb{E}[\hat{H}(y;b)] = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} G(y;Y_i,b)\right) = \mathbb{E}\left(G(y;Y_1,b)\right) &= \int_0^\infty G(t;y,b)h(t)\mathrm{d}t \\
&= \int_0^\infty \left(\int_0^y K(t;w,b)\mathrm{d}w\right) h(t)\mathrm{d}t \\
&= \int_0^y \int_0^\infty K(t;w,b)h(t)\mathrm{d}t\mathrm{d}w \\
&= \int_0^y \mathbb{E}[h(V_{1/(w-b),1/b})]\mathrm{d}w,
\end{aligned}
$$

where the random variable $V_{1/(w-b),1/b} \sim \text{RIG}[1/(w-b), 1/b]$. The last equality holds because the inner integral $\int_0^\infty K(t;w,b)h(t)\mathrm{d}t = \mathbb{E}[h\{V_{1/(w-b),1/b}\}]$. Then, expanding $V_{1/(w-b),1/b}$ around its mean w and also using $\mathbb{V}\text{ar}\{V_{1/(w-b),1/b}\} = b(w+b)$ yields

$$\begin{aligned}\int_0^y \mathbb{E}[h\{V_{1/(w-b),1/b}\}]\mathrm{d}t &= \int_0^y h(w)\mathrm{d}w + \frac{1}{2}\int_0^y (bw+b^2)h''(w)\mathrm{d}w + o(b^2)\\ &= H(y) + \frac{b}{2}\int_0^y wh''(w)\mathrm{d}w + o(b^2)\\ &= H(y) + \frac{by}{2}h'(y) - \frac{b}{2}\left[h(y)-h(0)\right] + o(b^2)\\ &= H(y) + \frac{b}{2}\big[yh'(y) - h(y)\big] + o(b) \equiv H(y) + \mathcal{O}(b).\end{aligned} \tag{21}$$

For the variance, we have from the definition,

$$\begin{aligned}\mathbb{V}ar[\hat{H}(y;b)] = \mathbb{V}ar\left[n^{-1}\sum_{i=1}^n G(y;Y_i,b)\right] &= \frac{1}{n}\mathbb{V}\text{ar}\left[G(y;Y_1,b)\right]\\ &= \frac{1}{n}\mathbb{E}\left[G^2(y;Y_1,b)\right] - \frac{1}{n}\left[\mathbb{E}(G(y;Y_1,b))\right]^2.\end{aligned}$$

The second term is easily obtained from (21). It remains to derive the second moment of the estimator, $\mathbb{E}\left[G^2(y;Y_1,b)\right] = \int_0^\infty G^2(y;t,b)h(t)\mathrm{d}t$ which can be recast as

$$\begin{aligned}\mathbb{E}[G^2(y;Y_1,b)] &= \int_0^\infty G^2(y;t,b)h(t)\mathrm{d}t\\ &= \int_0^\infty G(y;t,b)\Phi\left(\sqrt{\frac{t}{b}}+\sqrt{\frac{b}{t}}\right)\mathrm{d}t - \int_0^\infty G(t;y,b)F(t;b)\mathrm{d}t\\ &= \int_0^\infty G(y;t,b)\Phi\left(\sqrt{\frac{t}{b}}+\sqrt{\frac{b}{t}}\right)\mathrm{d}t - \int_0^y \mathbb{E}[F(V_{1/(w-b),1/b};y,b)]\mathrm{d}w,\end{aligned} \tag{22}$$

where $F(t;y,b) = \Phi\left(\sqrt{t/b}+\sqrt{b/t}-y/\sqrt{tb}\right)h(t)$ using a similar random variable $V_{1/(w-b),1/b}$ and tactics as used in the reductions leading to (21). Since $t,b>0$, we have that $2 \leq \sqrt{t/b}+\sqrt{b/t} < \infty$ and so $\Phi(2) \leq \Phi\left(\sqrt{t/b}+\sqrt{b/t}\right) \leq 1$. Therefore, we have

$$\Phi(2)\int_0^\infty G(y;t,b)h(t)\mathrm{d}t \leq \int_0^\infty \Phi\left(\sqrt{\frac{t}{b}}+\sqrt{\frac{b}{t}}\right)G(y;t,b)h(t)\mathrm{d}t \leq \int_0^\infty G(y;t,b)h(t)\mathrm{d}t.$$

But $\int_0^\infty G(y;t,b)h(t)\mathrm{d}t \equiv \mathbb{E}[G(y;Y_1,b) = H(y)+b[yh(y)-h(y)]/2+o(b)$ and $\Phi(2) = 0.97725$ so that

$$\int_0^\infty \Phi\left(\sqrt{\frac{t}{b}}+\sqrt{\frac{b}{t}}\right)G(y;t,b)h(t)\mathrm{d}t \approx H(y) + \frac{b}{2}[yh(y)-h(y)] + o(b) \tag{23}$$

For the second term in (22), expanding $V_{1/(w-b),1/b}$ around its mean w yields

$$\begin{aligned}
&\int_0^y \mathbb{E}[F(V_{1/(w-b),1/b}; y, b)]\mathrm{d}w \\
&\qquad = \int_0^y F(w; y, b)\mathrm{d}w + \frac{1}{2}\int_0^y \mathbb{V}\mathrm{ar}(V_{1/(w-b),1/b}; y, b)F''(w; y, b)\mathrm{d}w + o(b) \\
&\qquad = \int_0^y \Phi\left(\sqrt{w/b} + \sqrt{b/w} - y/\sqrt{wb}\right) h(w)\mathrm{d}w + \frac{b}{2}\int_0^y (w + b)F''(w; y, b)\mathrm{d}w + o(b) \\
&\qquad = \Phi(\sqrt{b/y})H(y) - \int_0^y \left\{\frac{\mathrm{d}}{\mathrm{d}w}\Phi\left(\sqrt{w/b} + \sqrt{b/w} - y/\sqrt{wb}\right)\int h(w)\mathrm{d}w\right\}\mathrm{d}w \\
&\qquad\qquad\qquad + \frac{b}{2}\int_0^y (w + b)F''(w; y, b)\mathrm{d}w + o(b)
\end{aligned} \tag{24}$$

The derivative in the integrand is

$$\frac{\mathrm{d}}{\mathrm{d}w}\Phi\left(\sqrt{w/b} + \sqrt{b/w} - y/\sqrt{wb}\right) = \left(\frac{1}{\sqrt{bw}} + \frac{y - b}{w\sqrt{bw}}\right)\phi\left(\sqrt{\frac{w}{b}} - \frac{(y - b)}{\sqrt{bw}}\right)$$

so that (24) equals $\Phi(\sqrt{b/y})H(y) + o(\sqrt{b})$. We now expand $\Phi(\sqrt{b/y})$ using a Taylor series expansion around 0 to get

$$\Phi(\sqrt{b/y}) = 1/2 + \sqrt{b}/(2\sqrt{2\pi y}) + \mathcal{O}(b).$$

Inserting this result into (24) and combining with (23) means that (22) is

$$\mathbb{E}[G^2(y; Y_1, b)] = H(y)/2 - H(y)\sqrt{b}/(2\sqrt{2\pi y}) + o(\sqrt{b})$$

and the approximate expressions for the variance in Lemma 3 follow. ■

Appendix B. Detailed Experimental Evaluations

Figures 11-22 illustrate performance on 2D datasets obtained by KNOB-SynC, K-mH, DEMP, DEMP+, MMC, EAC, GSL-NN, Spectral clustering, kernel k-means, DBSCAN*, Density peaks, PGMM, MSAL and MGHD. In all cases, plotting character and color represent the true and estimated group indicator. Tables 8 and 9 display performance, in terms of $\mathcal{R}$ and estimated $\hat{C}$, on 2D and higher-dimensional datasets, of KNOB-SynC (denoted as KNS in the table), K-mH, DEMP (DM), DEMP+ (DM+), MMC, EAC, GSL-NN (GSN), spectral clustering (SpC), kernel-k-means (k-km), DBSCAN* (D*), density peaks (DP), PGMM (PGM), MSAL (MSL) and MGHD (MHD). For k-km, $\hat{C}$ was set at the true C and not estimated. For each method, the absolute deviation of $\mathcal{R}$ for that method from the highest $\mathcal{R}$ for each dataset was obtained: the average and SD of these absolute deviations are also reported for each method in the last row.

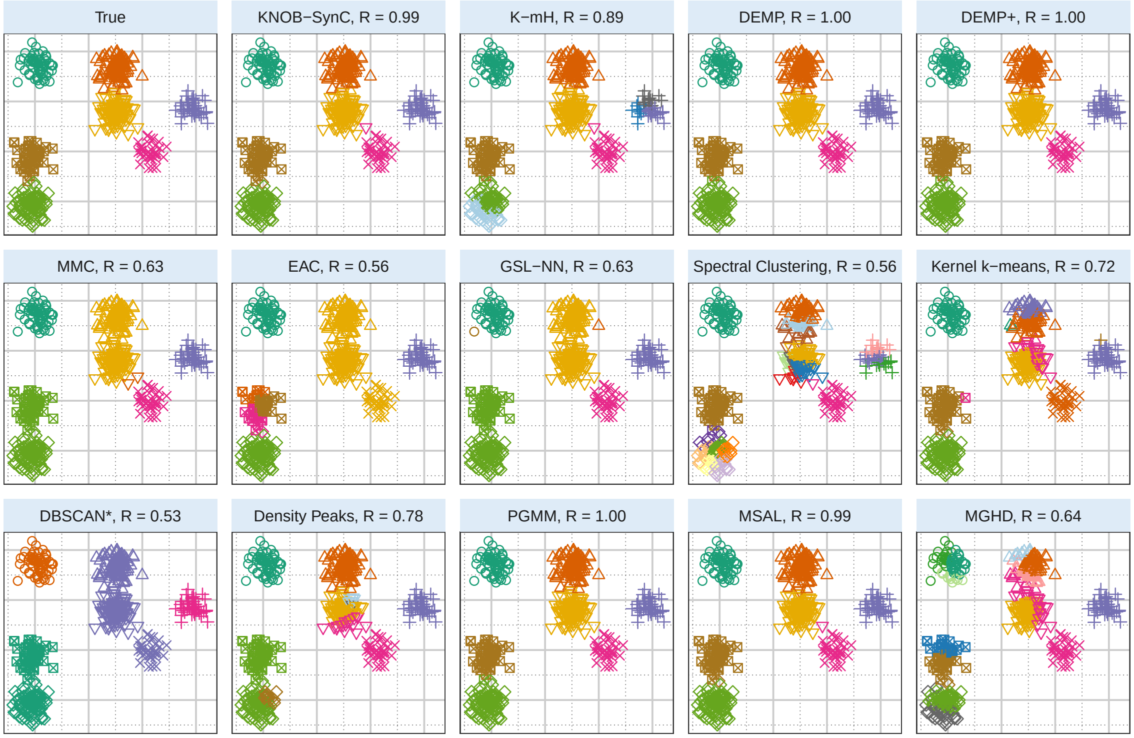


Figure 11: The `Spherical-7` example, and clusterings obtained using the 14 methods.

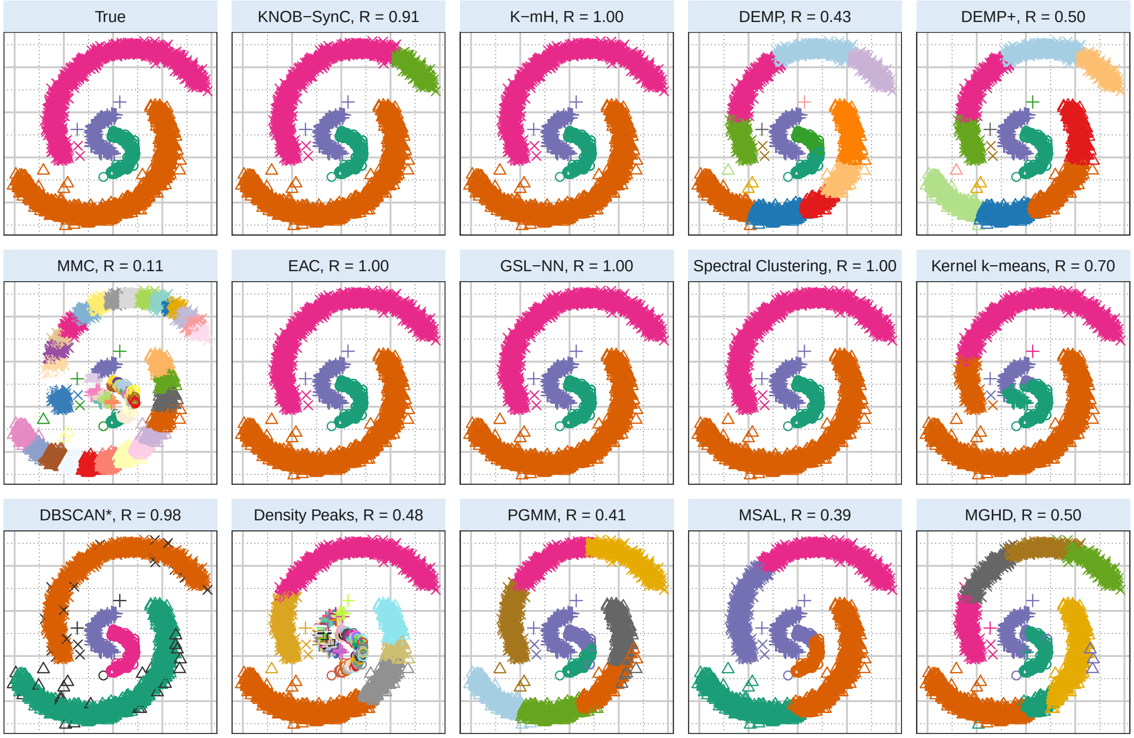


Figure 12: The `Bananas-Arcs` example and groupings obtained using the 14 methods.

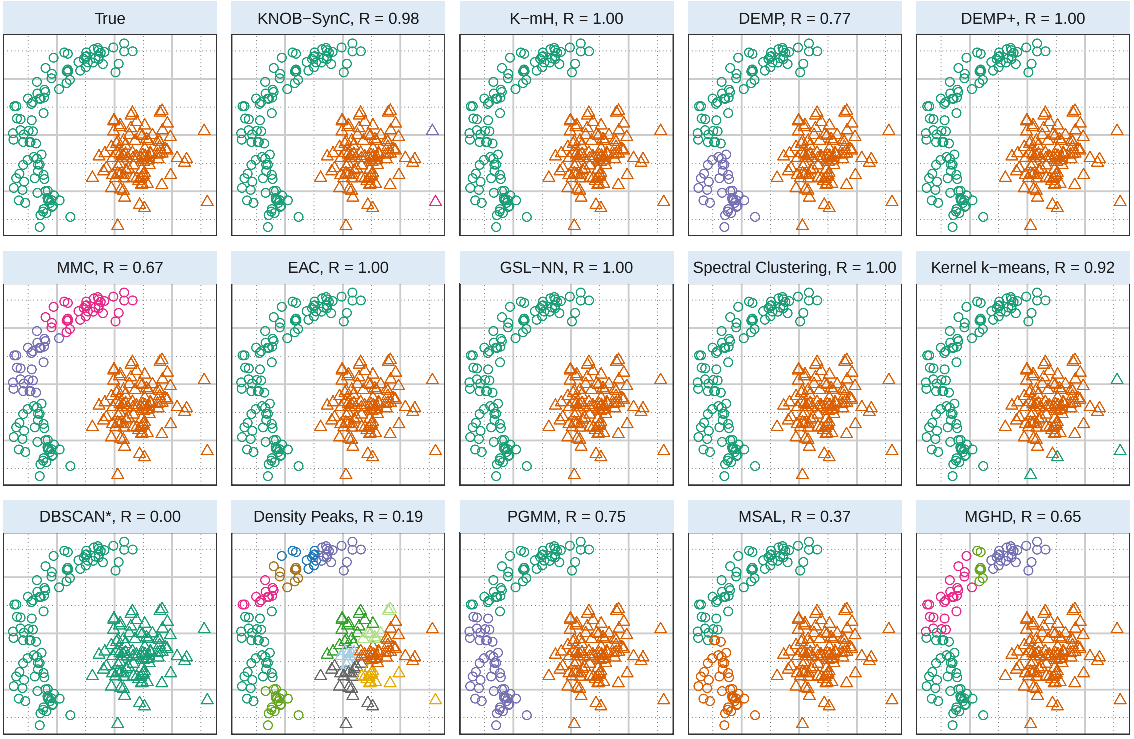


Figure 13: The `Banana-clump` example and groups obtained using the 14 methods.

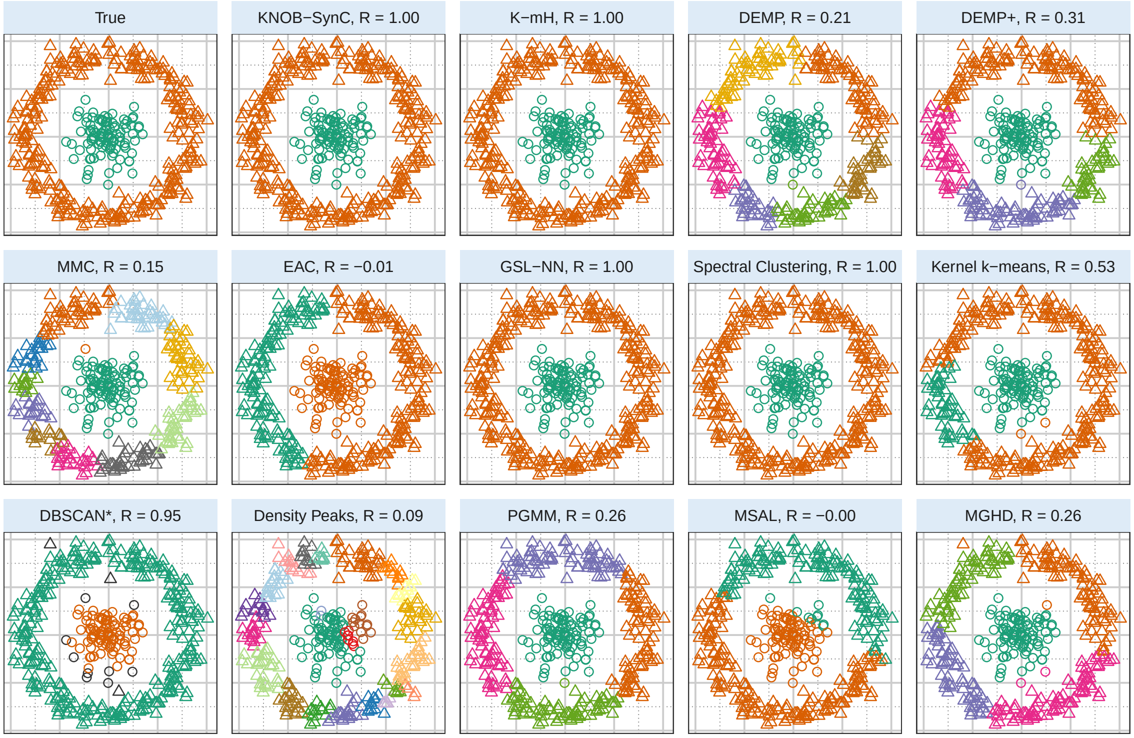


Figure 14: The `Bullseye` example and clusters obtained using the 14 methods.

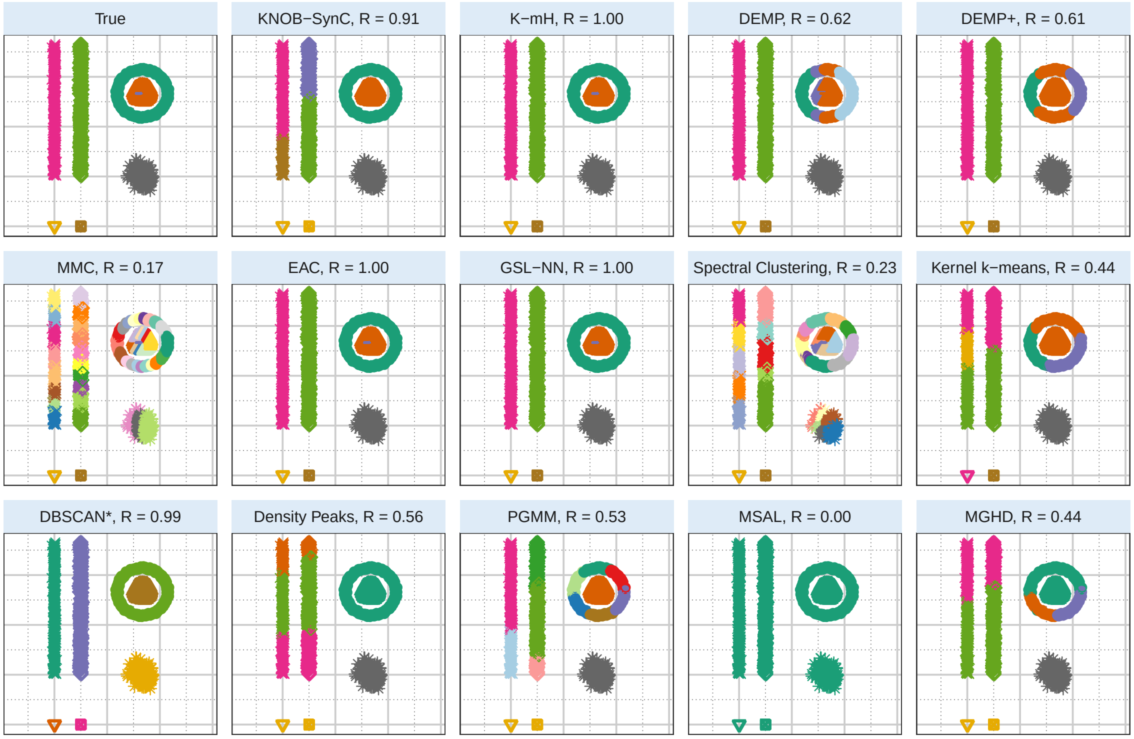


Figure 15: The `Bullseye-Cigarette` example and groups obtained with the 14 methods.

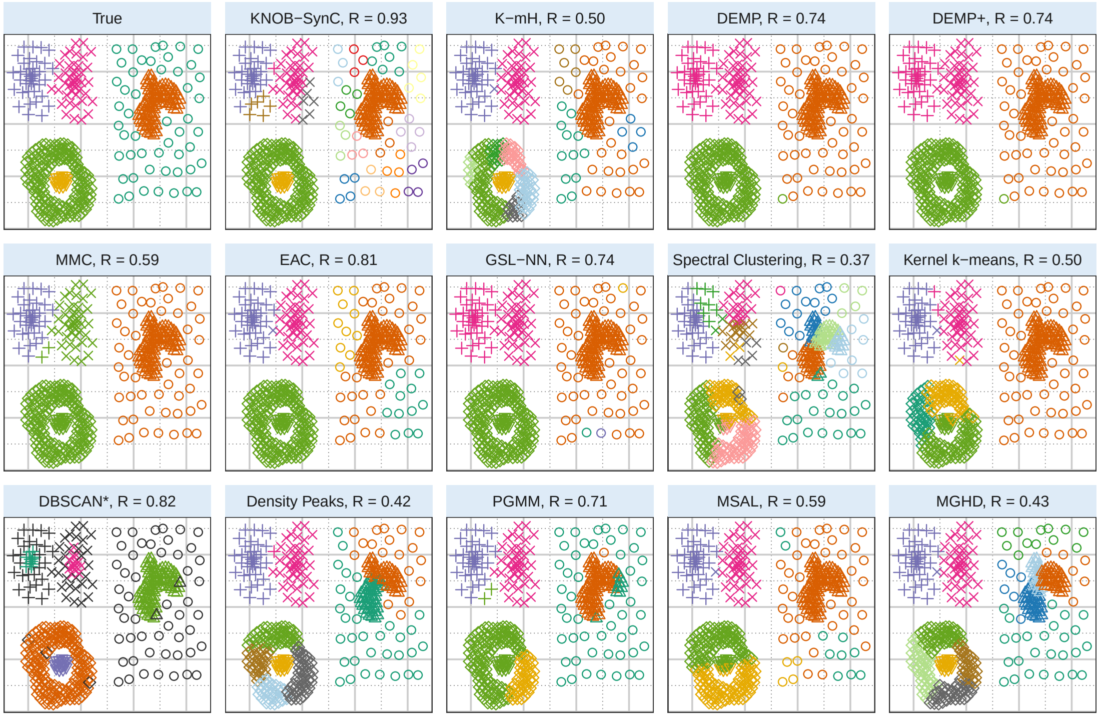


Figure 16: The `Compound` example and partitionings obtained using the 14 methods.

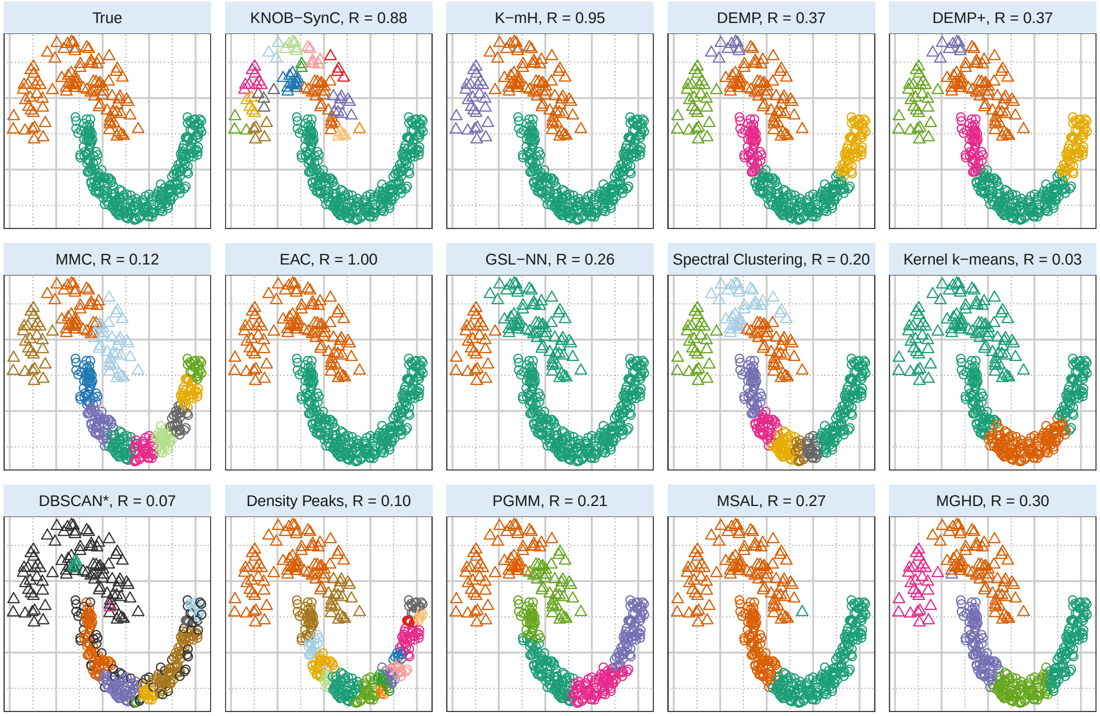


Figure 17: The `Half-Ringed clusters` example and groups obtained with the 14 methods.

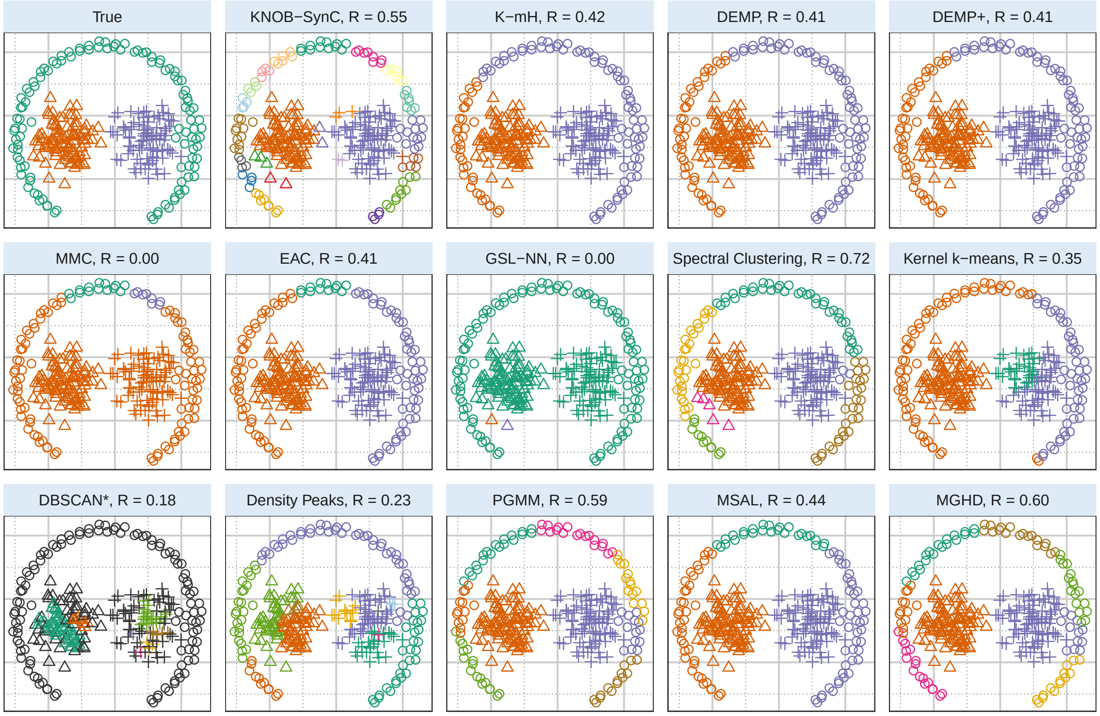


Figure 18: The `Path-based` example and groups obtained with the 14 methods.

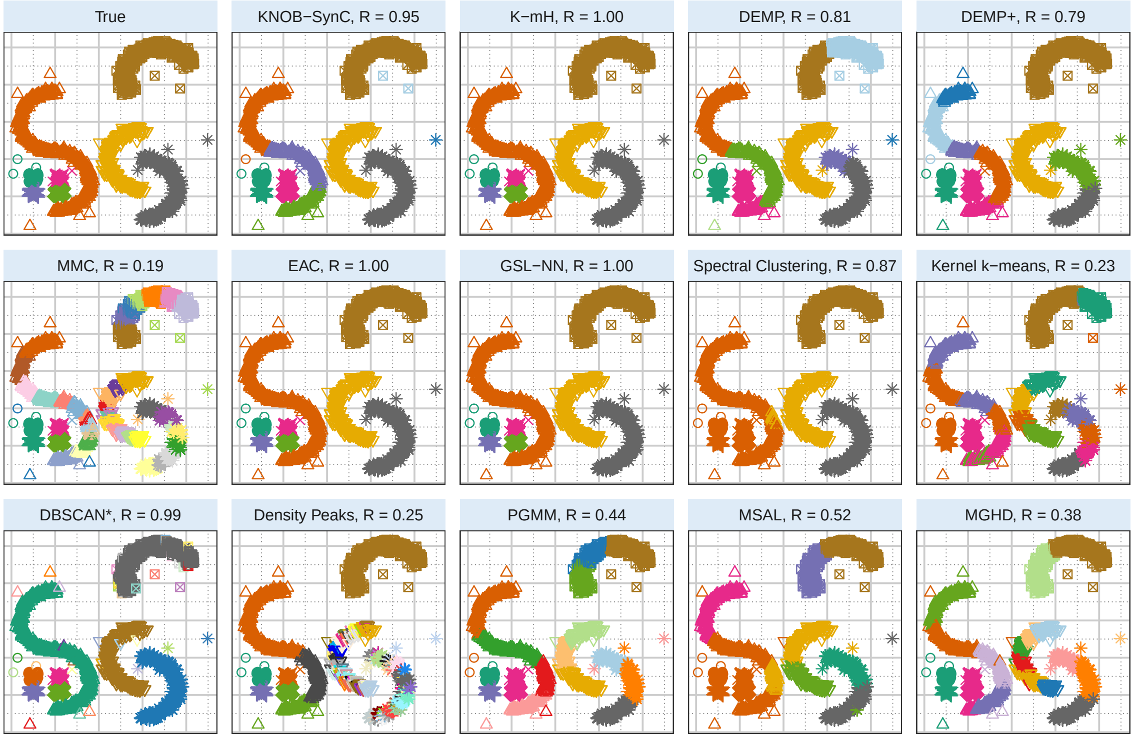


Figure 19: The `SCX-Bananas` example and clusters obtained using the 14 methods.

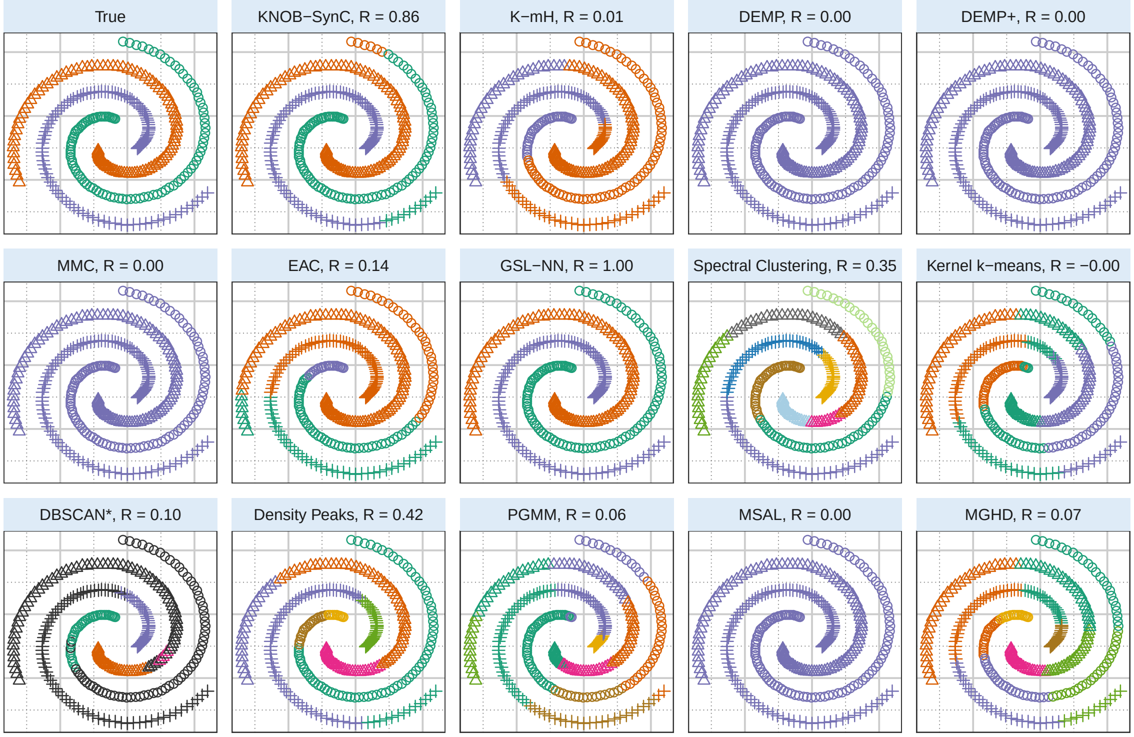


Figure 20: The `Spiral` example and groupings obtained using the 14 competing methods.

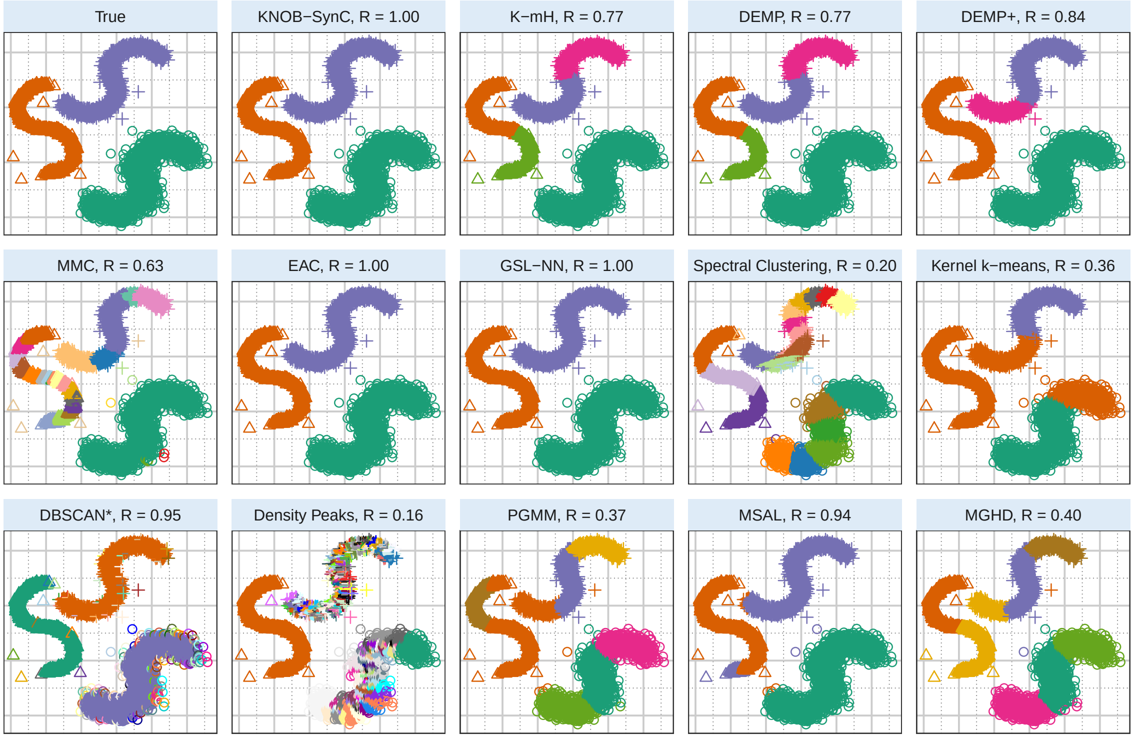


Figure 21: The SSS example and clusters obtained with the 14 competing methods.

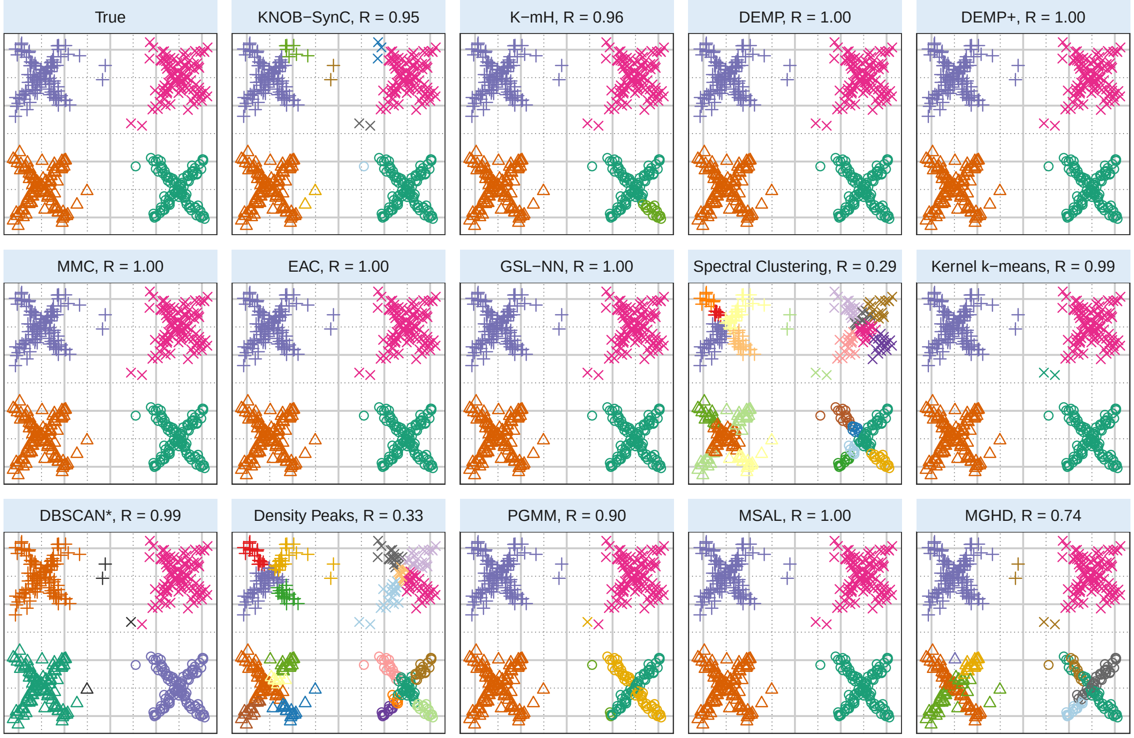


Figure 22: The XXXX example and groupings obtained using the 14 competing methods.

Table 8: Performance, in terms of $\mathcal{R}$ and estimated $\hat{C}$, on 2D datasets, of competing methods.

Dataset (n, p, K)	KNS	K-mH	DM	DM+	MMC	EAC	GSN	SpC	kk-m	D*	DP	PGM	MSL	MHD
7-Spherical	0.99	0.89	1	1	0.63	0.56	0.63	0.56	0.72	0.53	0.78	1	0.99	0.64
$(500, 2, 7)$	7	10	7	7	5	7	5	20	7	4	9	7	7	13
Aggregation	0.98	0.95	0.91	0.91	0.80	0.90	0.81	0.59	0.24	0.58	0.26	0.64	0.82	0.43
$(788, 2, 7)$	7	9	6	6	8	12	11	7	7	277	54	12	7	15
Banana-arcs	0.91	1	0.43	0.50	0.11	1	1	1	0.7	0.98	0.48	0.41	0.39	0.50
$(4515, 2, 4)$	5	4	11	10	52	4	4	4	4	47	460	9	4	8
Banana-clump	0.98	1	0.77	1	0.67	1	1	1	0.92	0	0.19	0.75	0.37	0.65
$(200, 2, 2)$	4	2	3	2	4	2	2	4	2	1	12	3	2	5
Bullseye	1	1	0.21	0.31	0.15	-0.01	1	1	0.53	0.95	0.09	0.26	-0.00	0.26
$(400, 2, 2)$	2	2	7	5	11	2	2	2	2	14	23	5	2	5
Bullseye-Cig	0.91	1	0.62	0.61	0.17	1	1	0.23	0.44	0.99	0.56	0.53	0	0.44
$(3025, 2, 8)$	7	6	7	7	5	9	14	16	8	17	115	4	1	4
Compound	0.93	0.5	0.74	0.74	0.59	0.81	0.74	0.37	0.5	0.82	0.42	0.71	0.59	0.43
$(399, 2, 6)$	19	13	5	5	5	6	6	13	6	117	9	6	6	12
Half-ringed	0.88	0.95	0.37	0.37	0.12	1	0.26	0.20	0.03	0.07	0.1	0.21	0.27	0.3
$(373, 2, 2)$	16	3	6	6	11	2	2	8	2	155	17	5	2	5
Path-based	0.55	0.42	0.41	0.41	0	0.41	0	0.72	0.35	0.18	0.23	0.59	0.44	0.60
$(300, 2, 3)$	21	2	2	2	3	15	3	3	3	217	9	7	3	7
SCX-Bananas	0.95	1	0.81	0.79	0.19	1	1	0.87	0.23	0.99	0.25	0.44	0.52	0.38
$(3420, 2, 8)$	8	8	7	9	39	8	8	4	8	25	389	16	8	17
Spiral	0.86	0.01	0	0	0	0.14	1	0.35	-0	0.1	0.42	0.06	0	0.09
$(312, 2, 3)$	3	2	1	1	1	9	3	11	3	208	7	7	1	7
SSS	1	0.77	0.77	0.84	0.63	1	1	0.20	0.36	0.95	0.16	0.37	0.94	0.40
$(5015, 2, 3)$	3	5	5	4	27	4	3	19	3	29	385	7	3	7
XXXX	0.95	0.96	1	1	1	1	1	0.29	0.99	0.99	0.33	0.90	1	0.74
$(415, 2, 4)$	10	5	4	4	4	4	4	20	4	8	20	6	4	9
$\bar{\mathcal{D}}$	0.06	0.17	0.35	0.32	0.58	0.22	0.17	0.40	0.53	0.35	0.64	0.44	0.48	0.52
$\mathcal{D}_\sigma$	0.06	0.28	0.31	0.31	0.33	0.35	0.27	0.34	0.31	0.39	0.20	0.29	0.38	0.21

Table 9: Performance, in terms of $\mathcal{R}$ and estimated $\hat{C}$, on high-dimensional datasets. (In the table, m displays the effective dimension of the dataset and is the number of coordinates, PCs or KPCs used as per the descriptions in Section 3.3.)

Dataset (n, p, K, m)	KNS	K-mH	DM	DM+	MMC	EAC	GSN	SpC	kk-m	DBSCAN*	DP	PGMM	MSAL	MGHD
Simplex-7	0.97	0.97	0.97	0.97	0.97	0.94	0.94	0.92	0.94	0.03	0.58	0.97	0.98	0.96
$(560, 7, 7, 7)$	7	7	7	7	7	6	7	7	7	508	79	7	7	7
E.coli	0.72	0.63	0.70	0.68	0.59	0.77	0.03	0.26	0.45	0.31	0.38	0.48	0	0.60
$(336, 7, 7, 5)$	9	4	4	4	8	10	16	15	7	174	12	8	1	8
Wines-13	0.92	0.62	0.52	0.5	0.67	0.6	0.38	0.43	0.6	0	0.26	0.66	0	0.16
$(178, 13, 3, 17)$	3	11	7	7	7	8	7	23	7	178	15	5	1	6
Wines-27	0.93	-0.01	1	1	0.91	0.62	0	0.35	0.88	0	0.56	0.85	0	0.95
$(178, 27, 3, 26)$	3	2	3	3	4	7	1	8	3	178	12	3	1	3
Olive Oils-Area	0.55	0.56	0.85	0.82	0.66	0.55	0.51	0.48	0.56	0.47	0.40	0.61	0.70	0.58
$(572, 8, 9, 8)$	4	8	7	12	11	14	5	18	9	201	27	5	9	7
Olive Oils-Region	0.89	0.69	0.45	0.47	0.22	0.46	0.67	0.23	0.4	0.44	0.41	0.59	0.58	0.54
$(572, 8, 3, 8)$	4	8	7	12	11	14	5	18	9	201	27	5	9	7
Image	0.54	0.48	0.49	0.46	0.28	0.59	0.10	0.22	0.52	0	0.38	0	0	0.56
$(2310, 19, 7, 8)$	12	17	18	17	46	40	48	45	7	1	56	1	1	7
Yeast	0.22	0.01	-0.01	-0.01	0.04	0.004	0.003	0.11	0.14	0	0.03	0	0	0.04
$(1484, 8, 10, 6)$	5	4	6	6	37	13	33	17	10	1	44	1	1	10
ALL	0.68	0.19	0.14	0.14	0.35	0.53	0.54	0.55	0.5	0	0.45	0.61	0	0
$(215, 1000, 7, 42)$	6	18	6	6	9	9	5	12	7	215	41	7	1	1
Zipcode	0.76	0.55	0.35	0.33	0.01	0.21	0.54	0.54	0	0	0.58	0.52	0	0.56
$(2000, 256, 10, 33)$	9	22	36	33	1	45	7	23	10	2000	53	6	1	6
Pendigits	0.72	0.51	0.58	0.6	0.26	0.58	0.004	0.42	0.3	0.3	0.54	0.44	0	0.67
$(10992, 16, 10, 18)$	15	9	40	32	59	58	59	48	10	7	9	6	1	10
$\bar{\mathcal{D}}$	0.04	0.29	0.21	0.22	0.31	0.22	0.44	0.35	0.27	0.62	0.35	0.24	0.56	0.25
$\mathcal{D}_\sigma$	0.09	0.27	0.20	0.20	0.23	0.17	0.29	0.21	0.21	0.26	0.16	0.15	0.33	0.26

References

F. Alimoglu. Combining multiple classifiers for pen-based handwritten digit recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, 1996.

F. Alimoglu and E. Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96)*, Istanbul, Turkey, 1996.

I. Almodóvar-Rivera and R. Maitra. RFASTfMRI: Fast adaptive smoothing and thresholding for improved activation detection in low-signal fMRI, 2019. R Package, URL http://github.com/ialmodovar/RFASTfMRI.

I. A. Almodóvar-Rivera and R. Maitra. FAST adaptive smoothed thresholding for improved activation detection in low-signal fMRI. *IEEE Transactions on Medical Imaging*, 38(12): 2821–2828, 2019. doi: 10.1109/TMI.2019.2915052.

A. Azzalini. A note on the estimation of a distribution function and quantiles by a kernel method. *Biometrika*, 68(1):326–328, 1981.

P. A. Bandettini, A. Jesmanowicz, E. C. Wong, and J. S. Hyde. Processing strategies for time-course data sets in functional mri of the human brain. *Magnetic Resonance in Medicine*, 30:161–173, 1993.

J.-P. Baudry and G. Celeux. *RmixmodCombi: Combining Mixture Components for Clustering*, 2014. URL `https://CRAN.R-project.org/package=RmixmodCombi`. R package version 1.0.

J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332 – 353, 2010.

J. W. Belliveau, D. N. Kennedy, R. C. McKinstry, B. R. Buchbinder, R. M. Weisskoff, M. S. Cohen, J. M. Vevea, T. J. Brady, and B. R. Rosen. Functional mapping of the human visual cortex by magnetic resonance imaging. *Science*, 254:716–719, 1991.

N. S. Berry and R. Maitra. Tik-means: Transformation-infused k-means clustering for skewed groups. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12(3):223–233, 2019.

T. Bouezmarni and O. Scaillet. Consistency of asymmetric kernel density estimators and smoothed histograms with application to income data. *Econometric Theory*, 21(02):390–412, 2005.

C. Bouveyron, G. Celeux, B. T. Murphy, and A. E. Raftery. *Model-Based Clustering and Classification for Data Science: With Applications in R.* Cambridge Series in Statistical and Probabilistic Mathematics, 2019.

R. P. Browne and P. D. McNicholas. A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, 43(2):176–198, 2015.

R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

H. Chang and D.-Y. Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41 (1):191 – 203, 2008. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2007.04.010. URL http://www.sciencedirect.com/science/article/pii/S0031320307002038.

S. Chattopadhyay and R. Maitra. Gaussian-mixture-model-based cluster analysis finds five kinds of gamma-ray bursts in the BATSE catalogue. *Monthly Notices of the Royal Astronomical Society*, 469(3):3374–3389, 2017. doi: 10.1093/mnras/stx1024.

S. Chattopadhyay and R. Maitra. Multivariate t-mixture-model-based cluster analysis of BATSE catalogue establishes importance of all observed parameters, confirms five distinct ellipsoidal sub-populations of gamma-ray bursts. *Monthly Notices of the Royal Astronomical Society*, 481(3):3196–3209, 07 2018. ISSN 0035-8711. doi: 10.1093/mnras/sty1940.

T. Chattopadhyay, R. Misra, A. K. Chattopadhyay, and M. Naskar. Statistical evidence for three classes of gamma-ray bursts. *Astrophysical Journal*, 667(2):1017, 2007. doi: https://doi.org/10.1086/520317.

A. Chaturvedi, P. E. Green, and J. D. Caroll. K-modes clustering. *Journal of Classification*, 18:35–55, 2001.

S. X. Chen. Probability density function estimation using gamma kernels. *Annals of the Institute of Statistical Mathematics*, 52(3):471–480, 2000.

J.-P. Dezalay, C. Barat, R. Talon, R. Syunyaev, O. Terekhov, and A. Kuznetsov. Short cosmic events - A subset of classical GRBs? In W. S. Paciesas and G. J. Fishman, editors, *American Institute of Physics Conference Series*, volume 265 of *American Institute of Physics Conference Series*, pages 304–309, 1992.

I. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas at Austin, 2004.

K. S. Dorman and R. Maitra. An efficient k-modes algorithm for clustering categorical datasets. *ArXiv e-prints:2006.03936*, 2020.

V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability and its Applications*, 14:153158, 1969. doi: 10.1137/1114019.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96*, volume 96, pages 226–231, 1996.

B. S. Everitt, S. Landau, and M. Leesem. *Cluster Analysis (4th ed.)*. Hodder Arnold, New York, 2001.

E. Forgy. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. *Biometrics*, 21:768–780, 1965.

M. Forina and E. Tiscornia. Pattern recognition methods in the prediction of italian olive oil origin by their fatty acid content. *Annali di Chimica*, 72:143–155, 1982.

M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia. Classification of olive oils from their fatty acid composition. In *Food Research and Data Analysis*, pages 189–214. Applied Science Publishers, London, 1983.

M. Forina, R. Leardi, and S. Lanteri. PARVUS - an extendible package for data exploration, classification and correlation, 1988.

S. D. Forman, J. D. Cohen, M. Fitzgerald, W. F. Eddy, M. A. Mintun, and D. C. Noll. Improved assessment of significant activation in functional magnetic resonance imaging (fmri): Use of a cluster-size threshold. *Magnetic Resonance in Medicine*, 33:636–647, 1995.

C. Fraley and A. E. Raftery. How many clusters? which cluster method? answers via model-based cluster analysis. *Computer Journal*, 41:578–588, 1998.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

B. C. Franczak, R. P. Browne, and P. D. McNicholas. Mixtures of shifted asymmetric Laplace distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1149–1157, 2014.

B. C. Franczak, R. P. Browne, P. D. McNicholas, and K. L. Burak. *MixSAL: Mixtures of Multivariate Shifted Asymmetric Laplace (SAL) Distributions*, 2018. URL `https://CRAN.R-project.org/package=MixSAL`. R package version 1.0.

A. L. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):835–850, 2005.

K. J. Friston, P. Jezzard, and R. Turner. Analysis of functional MRI time-series. *Human Brain Mapping*, 1:153–171, 1994.

K. J. Friston, A. P. Holmes, K. J. Worsley, J.-B. Poline, C. D. Frith, and R. S. J. Frackowiak. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, 2:189–210, 1995.

C. R. Genovese, N. A. Lazar, and T. E. Nichols. Thresholding of statistical maps in functional neuroimaging using the false discovery rate:. *Neuroimage*, 15:870–878, 2002.

Z. Ghahramani and G. E. Hinton. The EM algorithm for factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Toronto, Canada, 1997.

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

G. H. Glover. Deconvolution of impulse response in event-related BOLD fMRI. *Neuroimage*, 9:416–429, 1999.

M. Hahsler and M. Piekenbrock. *dbscan: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*, 2018. URL `https://CRAN.R-project.org/package=dbscan`. R package version 1.1-3.

J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28: 100–108, 1979.

C. Hennig. Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification*, 2010. doi: 10.1007/s11634-010-0058-3.

A. Hinneburg and D. Keim. Cluster discovery methods for large databases: from the past to the future. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, 1999.

P. Horton and K. Nakai. A probablistic classification system for predicting the cellular localization sites of proteins. *Intelligent Systems in Molecular Biology*, pages 109–115, 1985.

Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference*, page 2134, Singapore, 1997. World Scientific.

Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283304, 1998.

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

P. Jaccard. Ètude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Sociètè Vaudoise des Sciences Naturelles*, 37:547579, 1901.

A. Jain and R. Dubes. *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

A. K. Jain and M. H. C. Law. Data clustering: A users dilemma. In S. K. Pal, B. S., and B. S., editors, *Pattern Recognition and Machine Intelligence. PReMI 2005*, volume 3776 of *Lecture Notes in Computer Science*, pages 1–10, Berlin, Heidelberg, 2005. Springer.

Y. Jeon and J. H. T. Kim. A gamma kernel density estimation for insurance loss data. *Insurance: Mathematics and Economics*, 53:569–579, 2013. doi: http://dx.doi.org/10.1016/j.insmatheco.2013.08.009.

S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32:3:241–254, 1967.

L. Kaufman and P. J. Rousseuw. *Finding Groups in Data.* John Wiley & Sons, New York, 1990.

W. J. Krzanowski and Y. Lai. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, pages 23–34, 1988.

K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weisskoff, B. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, R. Turner, H.-M. Cheng, T. J. Brady, and B. R. Rosen. Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proceedings of the National Academy of Sciences of the United States of America*, 89:5675–5679, 1992.

N. A. Lazar. *The Statistical Analysis of Functional MRI Data*. Springer, 2008.

A. Lithio and R. Maitra. An efficient k-means-type algorithm for clustering datasets with incomplete records. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(6):296–311, 2018.

S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium*, 1:281–297, 1967.

P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):4955, 1936.

R. Maitra. Clustering massive datasets with applications to software metrics and tomography. *Technometrics*, 43(3):336–346, 2001.

R. Maitra. A statistical perspective to data mining. *Journal of the Indian Society of Probability and Statistics*, 6:28–77, 2002.

R. Maitra. Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6:144–157, 2009a. doi: 10.1109/TCBB.2007.70244.

R. Maitra. Assessing certainty of activation or inactivation in test-retest fMRI studies. *Neuroimage*, 47(1):88–97, 2009b.

R. Maitra. A re-defined and generalized percent-overlap-of-activation measure for studies of fMRI reproducibility and its use in identifying outlier activation maps. *Neuroimage*, 50(1):124–135, 2010.

R. Maitra and V. Melnykov. Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19 (2):354–376, 2010. doi: 10.1198/jcgs.2009.08054.

R. Maitra and I. P. Ramler. Clustering in the presence of scatter. *Biometrics*, 65:341 – 352, 2009.

R. Maitra, S. R. Roys, and R. P. Gullapalli. Test-retest reliability estimation of functional MRI data. *Magnetic Resonance in Medicine*, 48:62–70, 2002.

R. Maitra, V. Melnykov, and S. Lahiri. Bootstrapping for significance of compact clusters in multi-dimensional datasets. *Journal of the American Statistical Association*, 107(497): 378–392, 2012. doi: http://dx.doi.org/10.1080/01621459.2011.646935.

E. P. Mazets, S. V. Golenetskii, V. N. Ilinskii, V. N. Panov, R. L. Aptekar, I. A. Gurian, M. P. Proskura, I. A. Sokolov, Z. I. Sokolova, and T. V. Kharitonova. Catalog of cosmic gamma-ray bursts from the KONUS experiment data. I. *Astrophysics and Space Science*, 80:3–83, Nov. 1981. doi: 10.1007/BF00649140.

G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley and Sons, Inc., New York, 2000.

G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

P. D. McNicholas. *Mixture model-based classification*. Chapman and Hall/CRC, 2016.

P. D. McNicholas and T. B. Murphy. Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3):285–296, 2008.

P. D. McNicholas, A. ElSherbiny, A. F. McDaid, and T. B. Murphy. *pgmm: Parsimonious Gaussian Mixture Models*, 2018. URL `https://CRAN.R-project.org/package=pgmm`. R package version 1.2.3.

V. Melnykov. Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, 25(1):66–90, 2016.

V. Melnykov and R. Maitra. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010. ISSN 1935-7516. doi: 10.1214/09-SS053.

V. Melnykov and R. Maitra. CARP: Software for fishing out good clustering algorithms. *Journal of Machine Learning Research*, 12:69 – 73, 2011.

V. Melnykov, W.-C. Chen, and R. Maitra. MixSim: An R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25, 2012. URL `http://www.jstatsoft.org/v51/i12/`.

C. D. Michener and R. R. Sokal. A quantitative approach to a problem in classification. *Evolution*, 11:130–162, 1957.

S. Mukherjee, E. D. Feigelson, G. Jogesh Babu, F. Murtagh, C. Fraley, and A. Raftery. Three types of gamma-ray bursts. *Astrophyical Journal*, 508:314–327, Nov. 1998. doi: 10.1086/306386.

K. Nakai. UCI machine learning repository, 1996. URL `http://archive.ics.uci.edu/ml`.

K. Nakai and M. Kinehasa. Expert sytem for predicting protein localization sites in gram-negative bacteria. *PROTEINS: Structure, Function, and Genetics*, 11:95–110, 1991.

R. B. Nelsen. *An Introduction to Copulas*. Springer, New York, 2 edition, 2006.

D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

J. P. Norris, T. L. Cline, U. D. Desai, and B. J. Teegarden. Frequency of fast, narrow gamma-ray bursts. *Nature*, 308:434, Mar. 1984. doi: 10.1038/308434a0.

S. Ogawa, T. M. Lee, A. S. Nayak, and P. Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic Resonance in Medicine*, 14:68–78, 1990.

E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065, 1962. doi: 10.1214/aoms/1177704472.

T. L. Pedersen, S. Hughes, and X. Qiu. *densityClust: Clustering by Fast Search and Find of Density Peaks*, 2017. URL `https://CRAN.R-project.org/package=densityClust`. R package version 0.3.

A. D. Peterson, A. P. Ghosh, and R. Maitra. Merging k-means with hierarchical clustering for identifying general-shaped groups. *Stat*, 7(1):e172, 2018. doi: 10.1002/sta4.172.

T. Piran. The physics of gamma-ray bursts. *Rev. Mod. Phys.*, 76:1143–1210, Jan 2005. doi: 10.1103/RevModPhys.76.1143.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

A. E. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101:168–178, 2006.

D. B. Ramey. Nonparametric clustering techniques. In *Encyclopedia of Statistical Science*, volume 6, pages 318–319. Wiley, New York, 1985.

R.-D. Reiss. Nonparametric estimation of smooth distribution functions. *Scandinavian Journal of Statistics*, pages 116–119, 1981.

A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344 (6191):1492–1496, 2014.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832, 1956. doi: 10.1214/aoms/1177728190.

L. Rüschendorf. *Mathematical Risk Analysis*. Springer-Verlag, Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-33590-7.

D. C. S. Aeberhard and O. de Vel. Comparison of classifiers in high dimensional settings. Technical Report 92-02, Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland, 1992.

F. S. Samaria and A. C. Harter. Parameterization of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142, Sarasota, Florida, 1994.

M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385–2401, 2009. doi: 10.1109/TIP.2009.2025923.

O. Scaillet. Density estimation using inverse and reciprocal inverse Gaussian kernels. *Nonparametric Statistics*, 16(1-2):217–226, 2004.

G. Schwarz. Estimating the dimensions of a model. *Annals of Statistics*, 6:461–464, 1978.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, London, 1986. ISBN 0-412-24620-1.

M. Śmieja and M. Wiercioch. Constrained clustering with a complex cluster structure. *Advances in Data Analysis and Classification*, 11(3):493–518, 2017.

D. Steinley. Properties of the Hubert-Arabie adjusted Rand index. *Psychological methods*, 9(3):386, 2004.

W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 2010. doi: 10.1198/jcgs.2009.07049.

C. A. Sugar and G. M. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463), 2003.

B. Thirion, G. Varoquaux, E. Dohmatob, and J.-B. Poline. Which fMRI clustering gives good brain parcellations? *Frontiers in Neuroscience*, 8:167, 2014. ISSN 1662-453X. doi: 10.3389/fnins.2014.00167. URL `https://www.frontiersin.org/article/10.3389/fnins.2014.00167`.

D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, Chichester, U.K., 1985.

C. Tortora, A. ElSherbiny, R. P. Browne, B. C. Franczak, , P. D. McNicholas, and D. D. Amos. *MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions*, 2019. URL `https://CRAN.R-project.org/package=MixGHD`. R package version 2.3.2.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

K. Wagstaff. Clustering with missing values: No imputation required. In *Classification, clustering, and data mining applications*, pages 649–658. Springer, 2004.

M. P. Wand and M. C. Jones. *Kernel Smoothing.* Chapman & Hall/CRC, London, 1995. ISBN 0-412-55270-1.

R. Xu and D. C. Wunsch. *Clustering.* John Wiley & Sons, NJ, Hoboken, 2009.

E.-J. Yeoh, M. E. Ross, S. A. Shurtleff, W. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W. E. Evans, C. Naeve, L. Wong, and J. R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133 – 143, 2002. ISSN 1535-6108. doi: https://doi.org/10.1016/S1535-6108(02)00032-6.

C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971.

Y. Zhu, F. Dai, and R. Maitra. Three-dimensional radial visualization of high-dimensional continuous or discrete datasets. *ArXiv e-prints:1905.09505*, Mar. 2019.