Bioinformatics, 35(12), 2019, 2125–2132
doi: 10.1093/bioinformatics/bty932
Advance Access Publication Date: 8 November 2018
Original Paper

OXFORD

Data and text mining

Clusterdv: a simple density-based clustering method that is robust, general and automatic

João C. Marques[1,2] and Michael B. Orger [1,*]

[1]Champalimaud Research, Champalimaud Centre for the Unknown, Avenida Brasília, Doca de Pedrouços, Lisboa 1400-038, Portugal and [2]Rowland Institute at Harvard, 100 Edwin H. Land Boulevard, Cambridge, MA 02142, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on July 23, 2018; revised on October 15, 2018; editorial decision on November 2, 2018; accepted on November 7, 2018

Abstract

Motivation: How to partition a dataset into a set of distinct clusters is a ubiquitous and challenging problem. The fact that data vary widely in features such as cluster shape, cluster number, density distribution, background noise, outliers and degree of overlap, makes it difficult to find a single algorithm that can be broadly applied. One recent method, clusterdp, based on search of density peaks, can be applied successfully to cluster many kinds of data, but it is not fully automatic, and fails on some simple data distributions.

Results: We propose an alternative approach, clusterdv, which estimates density dips between points, and allows robust determination of cluster number and distribution across a wide range of data, without any manual parameter adjustment. We show that this method is able to solve a range of synthetic and experimental datasets, where the underlying structure is known, and identifies consistent and meaningful clusters in new behavioral data.

Availability and implementation: The clusterdv is implemented in Matlab. Its source code, together with example datasets are available on: https://github.com/jcbmarques/clusterdv.

Contact: michael.orger@neuro.fchampalimaud.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A notable feature in data is that the points, rather than being evenly distributed, are more densely clustered in some regions of space than others. Unsupervised computational methods that can automatically determine the number of clusters in data and define their natural boundaries are useful to identify unsuspected natural phenomena and are widely used across many disciplines of science. One of the aims of machine learning is to develop general purpose clustering heuristics that function automatically for the most diverse types of data possible and hence many clustering strategies have been proposed (Wiwie *et al.*, 2015; Xu and WunschII, 2005); however, there is no universal consensus on the definition of a cluster or on which clustering algorithm is the most effective (Jain *et al.*, 1999).

Some widely used clustering strategies, as e.g. k-means (Lloyd, 1982), k-medoids (Kaufmann and Rousseeuw, 1987), mixture models, and affinity propagation (Frey and Dueck, 2007), depend on assumptions about the cluster shape, and are therefore not suitable for detecting clusters with arbitrary shapes. Spectral clustering methods that use the eigenvectors of the similarity matrices (Donath and Hoffman, 1973; Shi and Malik, 1997), are able to detect clusters of arbitrary shape and a completely automatic version, self-tuning spectral clustering (ST-spectral), provides a means to select the optimal number of clusters (Perona and Zelnik-Manor, 2004). However, these methods can be sensitive to noise or clusters distributed over multiple scales (Zhang *et al.*, 2016). Density-based methods, such as density-based spatial clustering of applications with noise (DBSCAN), OPTICS and density peak clustering (clusterdp), also allow for clusters with arbitrary shape to be discovered (Ankerst *et al.*, 1999; Ester *et al.*, 1996; Rodriguez and Laio, 2014), but still require user input to set parameters, or partition the resulting dendrogram. Here we focus on one of these methods, clusterdp, which is fast, resilient to noise and captures clusters of arbitrary shapes. It also performed well in classifying faces from images and, in a meta-study of clustering methods, to identify clusters in

biomedical data (Wiwie *et al.*, 2015). However, in many cases, the output of clusterdp is critically dependent on the parameter that is used for estimating the local densities (d_c). Additionally, as we also show here, the clusterdp heuristic fails when applied to certain distributions with clearly distinguishable clusters.

We designed an alternative approach that is both more general and allows cluster selection to be fully automated. Our method, which we call clustering by density valleys (clusterdv), is based on similar principles to clusterdp, but differs in several key elements. We use an adaptive Gaussian density estimator to compute the local densities, in common with other variants of the method (Wang and Xu, 2017), and we define point separation based on how deep a density valley had to be traversed to connect pairs of points. We developed a robust rule to identify, and to hierarchically order, putative cluster centers. Last, we implemented methods, based on statistical comparison with reference distributions and the largest jump within the cluster hierarchy, to select the number of clusters automatically.

We validated clusterdv by applying it, without parameter tuning, to a wide variety of artificial and real-world test data with known ground truth cluster identity. We show that clusterdv can identify the correct number of clusters very reliably, including in distributions that cannot be clustered using clusterdp. The method also assigns points to the correct clusters with high accuracy. Finally, we show that it allows robust identification of behavioral categories in experimental data from larval zebrafish. Altogether clusterdv is an automatic unsupervised method for density cluster identification that achieves state of the art performance over a wide range of types of data, making it an ideal tool to discover structure in real world data where the ground truth is not known.

2 Materials and methods

2.1 Density valley clustering algorithm

The core algorithm for clusterdv is as follows:

i. Let $\mathbf{x}_i$ be a point in the dataset and $\rho(\mathbf{x}_i)$ the value of the kernel density estimation (KDE) at that point.
ii. The density valley depth on a line between two points, $D(\mathbf{x}_i, \mathbf{x}_j)$, is defined as the minimum value of the KDE sampled at a number (η) of discrete intervals on a line from $\mathbf{x}_i$ to $\mathbf{x}_j$. So, $D(\mathbf{x}_i, \mathbf{x}_j) = \min_{\mathrm{n}\in \mathrm{N}:\, \mathrm{N}=\left\{0,\, 1/\eta,\, 2/\eta \,\ldots\, 1\right\}} \rho((1-\mathrm{n})\cdot \mathbf{x}_i + \mathrm{n}\cdot \mathbf{x}_j)$.
iii. The density valley, $D(\mathbf{p})$, along a path $\mathbf{p}$ containing n unique points is defined as the smallest density valley between any pair of consecutive points. $D(\mathbf{p}) = \min_{\mathrm{m}\in \mathrm{N}:\, \mathrm{N}=\{1,\,2,\,3\ldots \mathrm{n}-1\}} D(\mathbf{x}_\mathbf{m}, \mathbf{x}_{\mathbf{m}+1})$.
iv. The path valley between two points, $P(\mathbf{x}_i, \mathbf{x}_j)$, $\mathrm{i} \neq \mathrm{j}$, is defined as the largest $D(\mathbf{p})$ for all paths that connect $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{P}_{ij}$). $\mathrm{P}(\mathbf{x}_i, \mathbf{x}_j) = \max_{\mathbf{p}\in \mathbf{P}_{ij}} D(\mathbf{p})$.
v. The maximum density valley for point $\bar{\mathbf{x}}$, $V(\bar{\mathbf{x}})$, is defined as the highest path valley connecting to any point of higher density. $V(\bar{\mathbf{x}}) = \max_{y:\, \rho(\mathbf{y})>\rho(\bar{\mathbf{x}})} P(\bar{\mathbf{x}}, \mathbf{y})$.
vi. The Separability Index, $\mathrm{SI}(\mathbf{x}_i)$, for all points $\mathbf{x}_i$ in the dataset, is calculated as $\mathrm{SI}(\mathbf{x}_i) = 1 - V(\mathbf{x}_i)/\rho(\mathbf{x}_i)$.
vii. The points are re-ordered in descending order of $\mathrm{SI}(\mathbf{x})$. The number of cluster centers is selected as $\operatorname{argmax}_{n:\, 1\geq n> nPts} \mathrm{SI}(\mathbf{x}_n) - \mathrm{SI}(\mathbf{x}_{n+1})$ ('max SI jump criterion'; nPts = number of data points with $\mathrm{SI} > 0$).

For a detailed description of clusterdv see Supplementary Material.

2.2 Clusterdv determination of number of clusters and point assignment

Data points with positive SI value were considered 'putative' cluster centers. Their SI value was used to construct dendrograms that reflect the ranking of cluster centers. For all datasets we computed three different criteria for determining a threshold SI value at which to cut the dendrogram. For the 'max SI jump' we take the largest jump in the SI between successive cluster centers. Alternatively, we take the 95th percentile of the second highest SI value from 100 reference distributions computed using the 'simplex' or the 'onion' methods (see Supplementary Material for details). All cluster centers with higher SI values than each of these criteria formed the associated solutions for each dataset. For all datasets the non-cluster center points were assigned, sequentially and in decreasing density order, to the same cluster of the nearest neighbor of higher density (Rodriguez and Laio, 2014).

See Supplementary Material for a detailed description of all the methods used.

3 Results

3.1 Identifying limitations of density peak clustering

The clusterdp algorithm relies in calculating two quantities: the local density (ρ) at each point and the minimum distance from each point to a point with higher ρ (δ) (Rodriguez and Laio, 2014). Both of these values depend on the choice of d_c, a free parameter in the clustering method which determines the spatial scale used to calculate local densities. Since clusterdp requires the user to select a number of cluster centers based on the distributions of ρ and δ, we developed an automated heuristic for this choice that we named automatic clusterdp (see Supplementary Material and Supplementary Fig. S1). In most cases, this method, when applied to datasets for which the ground truth was available, performed as well or better than human observers in selecting the correct number of clusters (Supplementary Fig. S2).

We took advantage of the good performance of automatic clusterdp to sample the d_c parameter for eight datasets used in the original study and found that the results of clusterdp are highly dependent on d_c, for all datasets we tested, and there was no single value or range of values that consistently gave correct results (Supplementary Fig. S3).

Another potential pitfall of clusterdp is that, if δ uses Euclidean distance as its metric, it will tend to favor density peaks that are far apart within large dense regions of points over nearby peaks that are clearly separated by an empty region of space. Arguably, though, the latter is a more salient feature of the data. To illustrate this point, we constructed a synthetic dataset, hereinafter called 'exclamation mark 1', that consists of groups of points drawn from two spatially uniform, rectangular distributions: an isolated group of low-density points, very close to an extended high-density region (Fig. 1A). This situation is commonly observed in real datasets, for example the zebrafish swimming data described later, where groups may be very unevenly represented and the more spatially restricted cluster has much lower density. The clusterdp algorithm is unable to find the low-density cluster without also splitting the single dense cluster into several parts (Fig. 1B). This is due to the fact that clusterdp ranks cluster centers according to the values of δ and local density, making it impossible, in case of the 'exclamation mark 1' example, for a user or an automatic algorithm to select the low-density cluster center (red dot) without including first the two cluster centers that exist in the high-density cluster (cyan and blue dots)

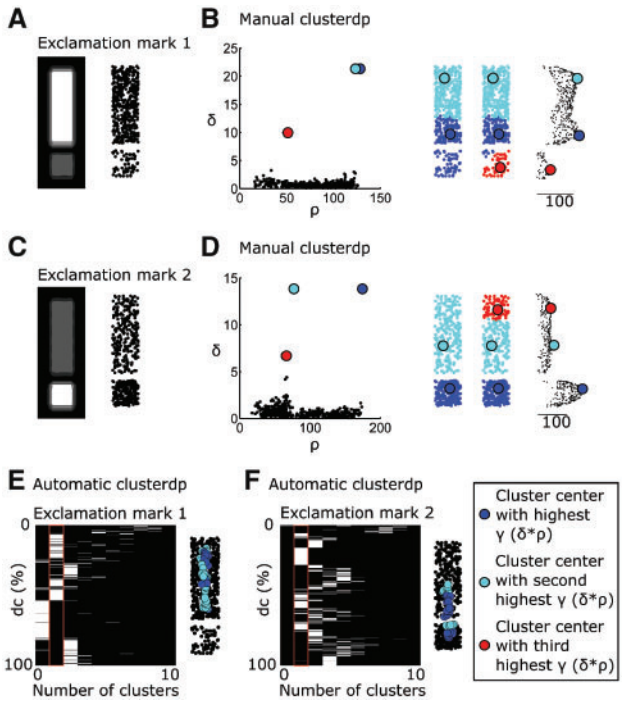

Fig. 1. Density peak clustering fails with uneven clusters. (**A, C**) The 'exclamation mark 1 and 2' datasets were drawn from two-part probability distributions (left). White represents 2.5-fold higher probability than gray and black is probability 0. (**B, D**) Manual clusterdp applied to the 'exclamation mark' datasets. Left to right: clusterdp decision plot (ρ versus δ) of the distribution in (A, C). Clusterdp solutions of data in (A, C) by picking the two or three cluster centers with highest γ ($\delta^*\rho$). Density profile of data in (A, C) ($d_c = 9\%$). (**E–F**) Left: Number of cluster centers picked by automatic clusterdp in function of d_c value for the 'exclamation mark 1' **(E)** and the 'exclamation mark 2' **(F)** datasets. Red outlines mark the ground truth (two clusters). Right: the cluster centers obtained by automatic clusterdp whenever the two-cluster solution was selected. Cluster centers and points are color coded blue-cyan-red in order of decreasing γ ($\delta^*\rho$) as in legend

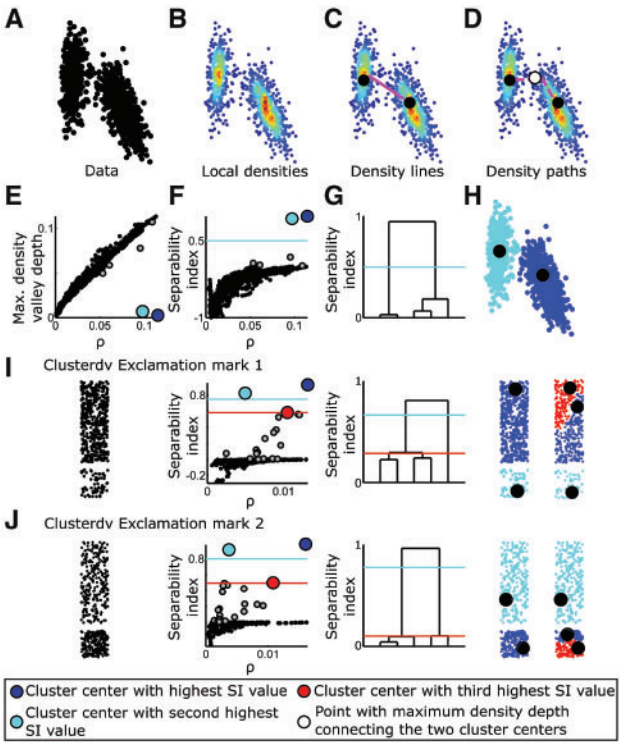

Fig. 2. The clusterdv method. **(A)** Point distribution drawn from a mixture of two gaussians. **(B)** Local densities are calculated using an adaptive gaussian density estimator. **(C)** Density profiles along straight lines between pairs of points are calculated, in a set of discrete steps. **(D)** The path valley is the highest minimum density along any path that connects one point to another, via this set of straight lines (path shown by white point). For each point, the maximum density valley depth is the highest such value between that point and a point of higher density. **(E)** Maximum density valley depth versus local density (ρ). **(F)** SI versus local density (ρ). Paths that don't have a dip in density can give negative values because the two points are in the same cluster and the end points are not considered. Blue line is a manually selected cutoff that selects the 'real' cluster centers (cyan and clue circles). Gray circles are 'sporadic' cluster centers that were not selected. **(G)** Dendrogram computed from SI. **(H)** Cluster assignment of the point distribution in (A) obtained by choosing the cluster centers with higher SI value than the blue line in (F, *G*). (**I–J**) Left to right: distribution, clusterdv decision plot (SI versus ρ), SI dendrogram, and two and three cluster solutions for the exclamation mark 1 **(I)** or 2 **(J)** datasets. Cluster centers and points are color coded blue-cyan-red in order of decreasing SI value as shown in legend. Black points mark cluster centers in distributions

(Fig. 1B). This problem is exacerbated by the fact that the originally proposed method for assigning points to clusters does not correctly partition the data, even when the cluster centers correctly identify both groups (Fig. 1B, third panel, dark blue points in the corner of the lower cluster). This failure to rank the cluster centers correctly does not occur in a dataset with similar characteristics, but in which the smaller cluster has higher density (Fig. 1C and D). To verify that this limitation was not dependent on parameter choice, we systematically varied d_c and plotted the locations of cluster centers for all cases where the correct number (2) was found. In all such cases, for the "exclamation mark 1" dataset, the cluster centers were both found in the larger cluster (Fig. 1E). The opposite result was observed for the 'exclamation mark 2' dataset where, if the algorithm found the correct number of clusters, they always spanned the two regions (Fig. 1F). These results suggest two key limitations of clusterdp: a sensitive dependence on parameter choice, and a failure to capture correctly the structure in certain simple data distributions.

3.2 The clusterdv algorithm

We set out to create a new clustering heuristic, based on the working principle of clusterdp, that was robust, general and automatic. That is to say, it should give repeatable results across many different data distributions, without the need for parameter fitting and should select the most suitable numbers of clusters in which to partition the data without human intervention. First, to calculate the local density at each point we used an adaptive Gaussian-based kernel density estimate (Breiman *et al.*, 1977), with the bandwidth at each point based on a heuristic that was applied identically, without any parameter tuning, to all datasets in this report (Fig. 2A and B). Second, we defined the separation of pairs of points in a way that aimed to capture a notion of distinct clusters. Specifically, two points should be considered well separated if you have to pass through a region of low density to get from one to the other, regardless of how close together they are in space. This separation was quantified by finding the path connecting two points, within a graph spanning the whole dataset, along which the minimum density was highest. We first estimated the minimum density along lines defined by single edges in the graph by sampling at discrete intervals from the previous kernel density estimate (Fig. 2C). After, we searched for the path joining each pair of points, following any of these lines, which dropped the

least in density along its length (density paths, Fig. 2D). To find such paths we used the single link algorithm (Sibson, 1973) on the previously calculated density lines. Therefore, the value of the 'density path' between two points is the lowest density you have to pass through to get from one point to the other. It should be possible to join two points that lie in the same unimodal cluster by a path whose density profile is always higher than the lower density point. Therefore, a point from which you cannot get to a higher density region without first passing through a region of lower density, is at a local maximum, and should be considered as a potential cluster center. So, for each point, we calculated the highest density path value between that point and any point of higher density (maximum density valley depth) (Fig. 2D).

Figure 2E shows this maximum density valley depth, plotted against the local density at each point. However, in order to be sensitive to clusters of very different densities, we should not consider the absolute value of the density drop, but rather how low it is relative to the associated peak. We chose to consider a low-density peak, separated by an empty region from the rest of the data, as a more salient feature of a dataset than a small dip separating two very high-density peaks. To capture this distinction, we calculated a SI by first dividing the maximum density valley depth of each point by its local density, in effect normalizing the drop of density to the density at each point, and then subtracting that value from 1 (see Section 2 for formula). A SI value of 1 indicates that there is a region of zero density between a point and any higher density region, and a negative value indicates that a point has a path to a denser point that never dips below the starting density and is therefore not a cluster center (Fig. 2F). All putative cluster centers (points with positive SI) can now be ranked according to their SI. These points are arranged in a hierarchical tree by connecting each new center point with the branch it would be assigned to, if the clusters were assigned without using that point (Fig. 2G). This SI tree reflects the hierarchical organization of clusters that exist in data. For datasets with complex clustering structure, e.g. with nested groups of clusters (Supplementary Fig. S4), the SI tree will capture this organization through groups of nodes at different levels. The total number of clusters is determined by distinguishing from the pool of putative cluster centers, which ones are 'real' and which arise due to stochastic variations in the density estimation of data with finite sample size. In practice, the selection of 'real' clusters is achieved by setting a cutoff on the SI values (blue line in Fig. 2F and G). We applied clusterdv to the 'exclamation mark' datasets and confirmed that the SI value ranks cluster centers correctly for both datasets (Fig. 2I and J). Additionally, if more than two cluster centers are selected (red lines in Fig. 2I and J), the data are partitioned in a manner that respects, as much as possible, the boundaries of the two true clusters (right panels of Fig. 2I and J), unlike clusterdp (right panels of Fig. 1B). In summary, clusterdv does not need any parameter optimization to work and reports the ranking of cluster centers correctly, even in difficult cases such as the 'exclamation mark 1' dataset.

3.3 Validation of clusterdv

Clusterdv is able to solve datasets with uneven clusters where the tightest cluster has lower density, but can the correct SI cutoff be determined automatically and does the method give correct solutions for data with other characteristics? To answer these questions, we applied clusterdv to 33 artificial and real-world datasets with known ground truth that were designed to present varying difficult challenges for clustering analysis (Supplementary Table S1). Three automatic criteria were tested to determine at which level the SI dendrogram should be cut and decide the number of clusters in each set. The more conservative method was to choose the largest jump in SI in the cluster tree. Alternatively, we estimated the SI distribution of clusters identified by chance in control datasets with similar density profiles, but with only a single peak in the underlying distribution ('onion') or similar spatial extent, but smoothing out variations in density ('simplex') (Supplementary Figs S5 and S6) and selected all clusters above these thresholds (see Supplementary Material). The three methods gave the same correct solution for 13 datasets, and, with the exception of the Olivetti face dataset (Samaria and Harter, 1994), spanned a range of cluster numbers encompassing the known correct value. The max SI jump, in every case, gave a higher SI cutoff than the other criteria and was able to solve correctly 29 of the 33 datasets (Fig. 3, for more examples see Supplementary Fig. S7). There were only four datasets for which the SI jump criteria failed to find the exact solution, and in three of those it identified a number of clusters just one step away from the correct answer. The methods based on reference distributions usually identified more structure, especially in the real-world datasets, as might be expected, but still also had high success rates (Supplementary Table S1). In the case of the Olivetti face dataset (Samaria and Harter, 1994), the clusters were fractured in the feature space that was used (Sampat *et al.*, 2009), so it may not be possible to improve on this result without a different representation of the data. Nevertheless clusterdv achieved higher success (Olivetti, SI jump, Fowlkes Mallows Index [FMI] = 0.77 and Supplementary Fig. S8), with much larger true associations for a given false positive

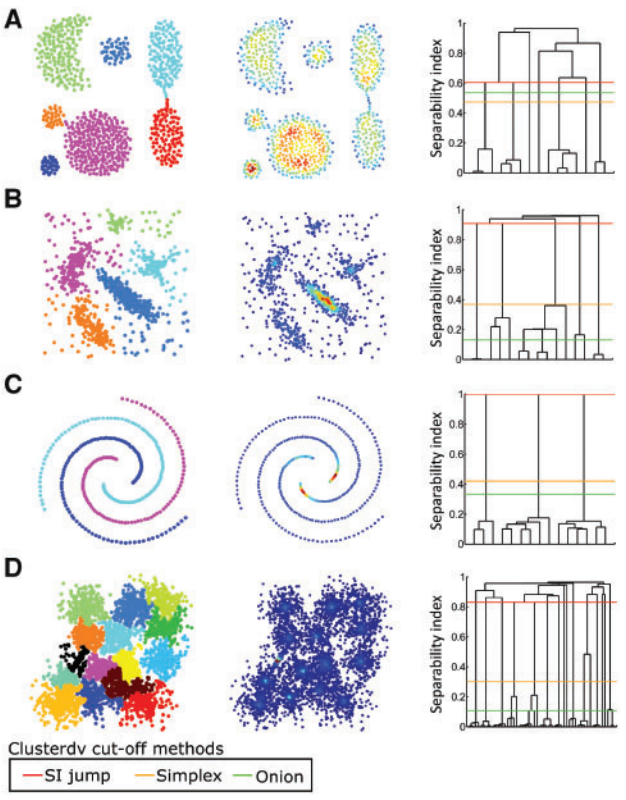

Fig. 3. Clusterdv gives the correct solution for a wide range of datasets. **(A)** (Gionis *et al.*, 2007), **(B)** (Rodriguez and Laio, 2014), **(C)** (Chang and Yeung, 2008), **(D)** (Fränti and Virmajoki, 2006). Left panels: cluster assignment by clusterdv using the SI jump criteria. Central panels: local densities calculated using adaptive Gaussian estimator. Right panels: SI dendrograms. Lines show cutoff criteria according to legend

rate, than previously published methods (Rodriguez and Laio, 2014) (78% at a false alarm rate of 1, versus ~65% previously reported for clusterdp). Another notable feature of these results is that the correct solution almost always existed within the dendrogram of cluster centers, and in 31 out of 33 cases could be obtained by cutting at a single level (shown in best cutoff column of Supplementary Table S1). In the case of the modified national institute of standards and technology (MNIST) handwritten number dataset (LeCun *et al.*, 1998), the 10-digit solution existed in the tree, but could not be found using a single SI cutoff, without also finding multiple clusters within digit groups. The 'correct' solution could however be found by choosing the most balanced partition into 10 clusters, and this could be used as a classifier which gave a 5.9% error rate for new data. This performance is comparable with the best unsupervised methods described for MNIST, that also use information about the construction of the dataset, but not individual data labels (5% error rate in Chen *et al.*, 2016) (Supplementary Fig. S9). Importantly, in all cases for which the correct number of clusters was identified, the assignment of cluster centers always matched the ground truth, and the assignment of points to clusters was largely correct (Supplementary Table S1).

We further tested the robustness of clusterdv by performing three tests: scaling the kernel bandwidths used for density estimation (Supplementary Fig. S10A–C), applying non-linear transformations to a simple point distribution (Supplementary Fig. S10D–G), and down sampling the number of points (Supplementary Fig. S11). Clusterdv gave correct solutions over most of the conditions of the three tests, showing that it is robust to variation in its density estimation, the shape of clusters and the number of points in data.

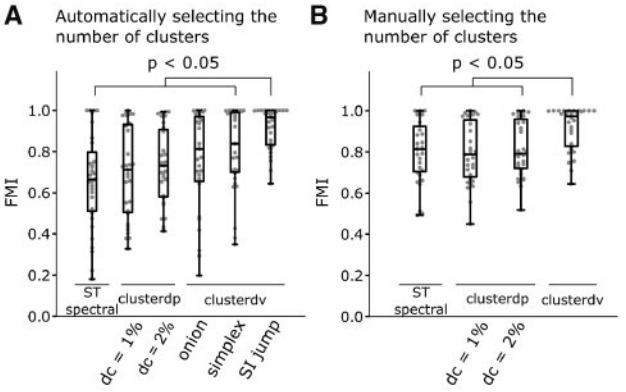

Fig. 4. Comparison between, ST-spectral clustering, clusterdp and clusterdv. The ST-spectral clustering, clusterdp and clusterdv methods (legends in x axis) were applied to 31 datasets that had known ground truth (for references of datasets see Supplementary Tables S1–S4) and the FMI was used to compare the clustering methods' performance with the ground truth. For clusterdp the d_c parameter was set to 1 and 2% (legend in x axis). **(A)** The number of clusters were set automatically by ST-spectral clustering (Perona and Zelnik-Manor, 2004), auto clusterdp (see Supplementary Fig. S1 for details) and in the case of clusterdv by using the simplex, onion or SI jump criteria. **(B)** The number of clusters were set manually (ST-spectral clustering) or by choosing the cutoff that corresponds to the known number of clusters, after sorting the cluster centers by the highest γ (clusterdp) or SI values (clusterdv). Gray points in plots are FMI values of each dataset. Boxplot indicates median with 25th and 75th percentile hinges, and whiskers extending to the smallest/largest value no less/more than 1.5 × interquartile range from the median. Black line marks the median. Paired Mann–Whitney test corrected for multiple comparisons by Holm-Bonferroni method, $n = 31$, comparisons with $P < 0.05$ were labeled in plots

3.4 Clusterdv outperforms clusterdp and ST-spectral clustering

Clusterdv is a robust automatic clustering method that solves a wide range of datasets, but how does its performance compare to other state of the art clustering algorithms? To make this comparison, we applied ST-spectral (Perona and Zelnik-Manor, 2004), a method that, as clusterdv, does not require any parameter tuning, and clusterdp, a cutting edge density-based clustering method, to the same array of datasets previously used to test clusterdv. For clusterdp we used the recommended range for the d_c parameter (Rodriguez and Laio, 2014). We started by testing the automatic versions of ST-spectral and clusterdp against the three automatic methods of clusterdv. Clusterdv, using the SI jump criteria, outperformed both automatic clusterdp and ST-Spectral, and also the two reference-based cutoffs (Fig. 4A), achieving significantly higher FMI values than any other method. Since the automatic rules to choose the number of clusters vary between ST-Spectral, clusterdp and clusterdv, we performed an additional test where the number of clusters is set, for each clustering method and dataset, according to the ground truth. In this case any problems in solving correctly the datasets are not due to the heuristic that chooses the cluster number but must come from problems in other steps of the algorithms. Figure 4B shows that, even in the situation of manually selecting the correct cluster number, clusterdv still clearly surpasses clusterdp and ST-Spectral, being able to correctly solve 31 of the 33 datasets (Supplementary Table S1), while clusterdp and ST- Spectral could only solve correctly 21 (Supplementary Tables S2–S4). Overall, clusterdv managed to outperform clusterdp and ST-Spectral in one of its automatic implementations (SI jump criteria), but also when the number of clusters was optimal, so the gain of performance for clusterdv likely comes from having a more reliable rule to identify the correct clustering structure than the other methods.

3.5 Clusterdv is able to categorize zebrafish startle behavior

The aim of clustering analysis is to identify groupings on real-world data that correspond to real natural phenomena. These datasets often have complex and noisy structure that differs from the datasets used to validate clustering algorithms. We used clusterdv to perform unsupervised categorization of movements of zebrafish larvae responding to acoustic stimuli. Larvae, which swim in short bouts of movement (Marques *et al.*, 2018), execute two types of escapes (C-starts) in response to acoustic startles: one bout type with short latency (SLC) and another with long latency (LLC) (Fig. 5A) (Burgess and Granato, 2007). Critically, both responses differ in a set of kinematic parameters and the neural circuits that produce these behaviors (Supplementary Figs S12 and S13) (Burgess and Granato, 2007). The proportion of the two types observed varies with the experimental conditions, creating datasets with uneven distributions of bouts. Also, the fish do not always respond to the stimulus with C-starts, so these datasets often have bouts that appear in kinematic space as outliers or that degrade cluster boundaries. In spite of these challenges, the SLC responses are stereotypical and form a tight cluster that appears distinct from the wide spread cluster that corresponds to the LLC bout type (Fig. 5B). These two swim types can be categorized by sorting them by the response latency to stimulus onset (red line in Fig. 5A) or by setting a threshold that separates them in kinematic space (green line in Fig. 5B) providing a 'ground truth' to the dataset. If the tail movements associated with swim bouts of each category are superimposed it becomes apparent that they correspond to two types of movements that are stereotypical within group (Supplementary Fig. S13B).

We applied five commonly used clustering methods to this dataset: k-means (Jain, 2010), agglomerative hierarchical clustering with

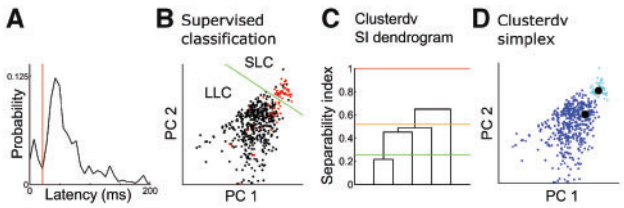

Fig. 5. Clusterdv enables unsupervised categorization of zebrafish acoustic startle behavior. **(A)** Swim bout latency from start of acoustic startle. Red line divides swims bouts with short latency (SLC) from bouts with long latency (LLC). **(B)** Principal component analysis (PCA) applied to swim bout kinematic parameters (see Supplementary Material for details). Red points correspond to SLC bouts, whereas black points correspond to LLC bouts categorized by red line in (A). Green line separates the small SLC cluster from the larger LLC cluster. **(C)** Clusterdv SI dendrogram for data in (B). Lines represent cutoff criteria used to choose number of clusters. Red line, SI jump. Orange line, simplex threshold; green line, onion threshold. **(D)** Clusterdv assignment according to the simplex threshold in (C). Black points mark cluster centers

unweighted pair group method with arithmetic mean (UPGMA) (Sokal and Michener, 1958), DBSCAN (Ester *et al.*, 1996), ST-Spectral (Perona and Zelnik-Manor, 2004) and clusterdp (Rodriguez and Laio, 2014). These algorithms weren't able to categorize the two types of movements, even when the correct number of cluster number were set by hand, or in the case of DBSCAN, when the parameter space was searched (Supplementary Figs S14 and S15). When we applied clusterdv to this dataset, the cluster center ranking was correct (Fig. 5C), and the algorithm was able, using an automatic criterion, to find essentially the same solution that was found by drawing a line in kinematic space or sorting the swims by latency (compare Fig. 5B with D). Clusterdv is thus able to categorize automatically animal behavior in a dataset with unevenly distributed data that are corrupted by noise and outliers.

4 Discussion

We have described here a novel, robust and simple density-based clustering algorithm, clusterdv, based on the density valleys between data points, which is applicable to a wide variety of data. It delivers better results than clusterdp, a state of the art density-based clustering algorithm (Rodriguez and Laio, 2014), and ST-spectral, a method that also does not need any parameter setting. In particular, clusterdv is able to find tight clusters of low density, where clusterdp fails, because its rule gives more importance to gaps between clusters than distances. Also, clusterdv's rule to find the number of clusters is more robust than the heuristic used by ST-spectral (Perona and Zelnik-Manor, 2004). Thus, a fully automatic version of clusterdv significantly outperforms clusterdp and ST-Spectral across a wide variety of datasets, even in the case where the number of cluster centers is manually set for these methods based on prior knowledge of the data structure (Supplementary Fig. S16).

All density-based clustering methods suffer from the problem that density estimation for data with finite sample size produces 'sporadic' local maxima that are not related to the 'real' structure present in data. Clusterdv produces a hierarchical tree of 'putative' cluster centers and uses an intuitive metric, the separability index or SI to rank their importance. The number of 'putative' cluster centers is often small, because all data points with negative SI values, which are not separated by density valleys, are *a priori* excluded from being cluster centers. To determine how many clusters exist in the data, it is necessary to decide which cluster centers are likely to be genuine, and which may occur sporadically due to sampling error, by setting an appropriate threshold on the SI value. We automated this step of the algorithm, by developing data-based criteria to choose the number of clusters. One such criterion, selection of the largest jump in SI, correctly determined the number of clusters in 29 out of 33 datasets, and is close to the correct solution in all cases, giving results that are comparable to setting the correct number of clusters by hand. It should be noted that this method is based on the assumption that any clustered organization is clearly distinct from noise in the data. For real world datasets, it is not clear that this assumption will always be met. Therefore, to be applied in real-world datasets, such as the zebrafish swimming data, we constructed, from the original data, references, termed 'onion' and 'simplex', which match the density profiles and spatial distributions of the data, respectively, but not the spatial structure, similar to the gap statistic method (Tibshirani *et al.*, 2001). These reference distributions are used to measure the probability that 'sporadic' cluster centers may arise in that particular set of data. Often the 'onion' and 'simplex' methods also gave the correct solution for many of the datasets we tested (14 and 21 correct datasets, respectively), but other times these methods overestimated the number of clusters. Even when the number of clusters did not match the ground truth, these methods still showed good performance, because the dendrogram reflected the true underlying data structure. It is likely that some of the datasets, in particular the real-world ones, contain other clustered organization that is not captured by the manual labeling. Thus, it is possible, in some cases, that the reference-based methods are uncovering meaningful structure.

Most clustering methods need one, or several, parameters to be set, so that correct results are obtained for different data distributions. These criteria introduce a subjective step in clustering analysis that may impact the particular solutions obtained. This is not the case for clusterdv. We benchmarked clusterdv on a set of 33 distributions with known ground truth that were chosen because they offer difficult challenges to clustering analysis such as: arbitrary cluster shape (Chang and Yeung, 2008; Fu and Medico, 2007; Gionis *et al.*, 2007; Jain and Law, 2005), number (Karkkainen and Franti, 2002) and spatial distribution (Veenman and Reinders, 2002), clusters with fuzzy edges (Fränti and Virmajoki, 2006; Monti *et al.*, 2003), data with many dimensions (Charytanowicz *et al.*, 2010; Zachary, 1977), corruption with noise (Karypis *et al.*, 1999; Rodriguez and Laio, 2014), and distributions with uneven proportions of clusters; and did not need to adjust any parameter to solve a particular dataset. Nevertheless, there are parameters that can be set in clusterdv, if desired. One such parameter, the number of edges used to calculate the density lines, allows a tradeoff of computational time versus accuracy, and needs to be set sufficiently large not to degrade the results. The density estimation by Gaussian mixture (Breiman *et al.*, 1977) may be performed using distinct methods or rules, but we found that the one simple heuristic used here always gave satisfactory results and it is robust to being scaled 10 fold and to data being down sampled. Other methods have been proposed that improve on, or automate, aspects of clusterdp (e.g. Courjault-Radé *et al.*, 2016; Mehmood *et al.*, 2017; Wang and Xu, 2017; Zhang *et al.*, 2016), but, to our knowledge, none has been demonstrated to work well across a similar range of datasets without parameter tuning.

Although clusterdv is able to solve a wide range of types of data it also has some limitations. It will not separate clusters that don't have separate peaks but are distinguished by sharp gradients in point density or where sharp gradients in density are not well captured by the KDE (Supplementary Fig. S17). The former is a class of clustering problems that are particularly challenging for density-based

methods, but can be solved by graph based algorithms (Zhang, 1971). The increased accuracy of clusterdv also comes with the trade-off of being slower than clusterdp (Supplementary Fig. S18). The rate-limiting step of the algorithm is the step where the density valley depth is computed, since the density has to be estimated along paths between all pairs of points. The computational time of this step scales with the square of the number of points. To mitigate this drawback of clusterdv we implemented two faster versions that limit the set of edges used for the density valley depth calculation (Kruskal, 1956), with minimal effect on the output, and set the option of using Matlab's parallel processing toolbox (Supplementary Material).

Clusterdv does not work directly with distance-based data, because it is necessary to embed this kind of data in a low dimensional space to calculate ρ and the density paths. We used two methods to create low-dimensional embeddings for high-dimensional data, T-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) (McInnes and Healy, 2018), and in most cases clusterdv outperformed clusterdp (Supplementary Table S5). Nevertheless, any other method to reduce dimensionality could be combined with clusterdv.

Finally, we applied clusterdv to the difficult problem of unsupervised behavioral categorization. We created a zebrafish larvae behavioral dataset that is sparse and composed of highly uneven clusters that are plagued with noise, but is known to contain two distinct swim categories (Burgess and Granato, 2007). This dataset proved to be challenging for commonly used clustering algorithms. Clusterdv could identify, in a completely automatic fashion, meaningful behavioral categories that these animals use when startled with acoustic stimuli, while clusterdp failed to provide correct results.

In many situations, it is important to determine the clusters that exist in a dataset, without a priori knowledge of their number or shape. To do this with confidence requires a method that delivers consistent results, and robustly selects the correct number and distribution of clusters. The systematic validation of clusterdv across many artificial and real world datasets, makes it suitable to apply to novel problems. We expect clusterdv to be useful in analyzing a wide range of data that has structure that reflects natural phenomena, but where the ground truth is unknown.

Acknowledgements

We thank Alfonso Renart, Christian Machens and Gonzalo de Polavieja for helpful discussions during the project and Drew Robson, Jennifer Li and Mattia Bergomi for critically reading the manuscript.

Funding

J.C.M. was supported by a Scholarship from the Portuguese Fundação para a Ciência e Tecnologia (FCT). The work was supported by grants to MBO from the Bial Foundation [185/12], Marie Curie [FP7-PEOPLE-2011-CIG] and FCT [PTDC/NEU-NMC/1276/2012], European Research Council [ERC-2017-COG-773012].

Conflict of Interest: none declared.

References

Ankerst,M. *et al.* (1999) OPTICS: ordering points to identify the clustering structure. In: *ACM Sigmod Record*, Vol. **28**, pp. 49–60.

Breiman,L. *et al.* (1977) Variable kernel estimates of multivariate densities. *Technometrics*, **19**, 135–144.

Burgess,H.A. and Granato,M. (2007) Sensorimotor gating in larval zebrafish. *J. Neurosci.*, **27**, 4984–4994.

Chang,H. and Yeung,D.-Y. (2008) Robust path-based spectral clustering. *Pattern Recogn.*, **41**, 191–203.

Charytanowicz,M. *et al.* (2010) Complete gradient clustering algorithm for features analysis of x-ray images. *Inform. Technol. Biomed.*, **69**, 15–24.

Chen,X. *et al.* (2016) Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: *Neural Information Processing Systems 29 (NIPS)*, pp. 2172–2180.

Courjault-Radé,V. *et al.* (2016) *Improved Density Peak Clustering for Large Datasets*. HAL Archives, pp. 1–28, https://hal.archives-ouvertes.fr/hal-01353574.

Donath,W.E. and Hoffman,A.J. (1973) Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, **17**, 420–425.

Ester,M. *et al.* (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231.

Fränti,P. and Virmajoki,O. (2006) Iterative shrinking method for clustering problems. *Pattern Recogn.*, **39**, 761–775.

Frey,B.J. and Dueck,D. (2007) Clustering by passing messages between data points. *Science*, **315**, 972–976.

Fu,L. and Medico,E. (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. A novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, **8**, 3.

Gionis,A. *et al.* (2007) Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, **1**, 4.

Jain,A.K. (2010) Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.*, **31**, 651–666.

Jain,A.K. *et al.* (1999) Data clustering: a review. *ACM Comput. Surv.*, **31**, 264–323.

Jain,A.K. and Law,M.H.C. (2005) Data clustering: a user's dilemma. In: Pal,S.K., Bandyopadhyay S., Biswas S. (eds) *Pattern Recognition and Machine Intelligence*. PReMI 2005. Lecture Notes in Computer Science. Vol. 3776. Springer, Berlin, Heidelberg.

Karkkainen,I. and Franti,P. (2002) Dynamic local search for clustering with unknown number of clusters. In: *16th International Conference on Pattern Recognition*. Vol. 2, Quebec City, Quebec, Canada, pp. 240–243.

Karypis,G. *et al.* (1999) Chameleon: hierarchical clustering using dynamic modeling. *IEEE Comput.*, **32**, 68–75.

Kaufmann,L. and Rousseeuw,P.J. (1987) Clustering by means of medoids. In: Dodge, Y. (ed.) *Statistical Data Analysis Based on the L1–Norm and Related Methods*, North-Holland, Elsevier, pp. 405–416.

Kruskal,J.B. (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.*, **7**, 48–50.

LeCun,Y. *et al.* (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2334.

Lloyd,S.P. (1982) Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, **28**, 129–136. IT

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

Marques,J.C. *et al.* (2018) Structure of the zebrafish locomotor repertoire revealed with unsupervised behavioral clustering. *Curr. Biol.*, **28**, pp. 181–195.E5.

McInnes,L. and Healy,J. (2018) UMAP: uniform manifold approximation and projection for dimension reduction. ArXiv:1802.03426.

Mehmood,R. *et al.* (2017) Clustering by fast search and merge of local density peaks for gene expression microarray data. *Sci. Rep.*, **7**, 45602. doi: 10.1038/srep45602.

Monti,S. *et al.* (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.*, **52**, 91–113.

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

Samaria,F.S. and Harter,A.C. (1994) Parameterisation of a stochastic model for human face identification. In: *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, USA, pp. 138–142.

Sampat,M.P. *et al.* (2009) Complex wavelet structural similarity: a new image similarity index. *IEEE Trans. Image Process*, **18**, 2385–2401.

Shi,J. and Malik,J. (1997) Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**, 888–905.

Sibson,R. (1973) SLINK: an optimally efficient algorithm for the single-link cluster method. *Comput. J.*, **16**, 30–34.

Sokal,R.R. and Michener,C.D. (1958) A statistical method for evaluating systematic relationships. *Univ. Kansas Sci. Bull.* **28**, 1409–1438.

Tibshirani,R. *et al.* (2001) Estimating the number of clusters in a data set via the gap statistic. *J. Roy. Stat. Soc. B.* **63**, 411–423.

Veenman,C.J. *et al.* (2002) A maximum variance cluster algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**, 1273–1280.

Wang,X.F. and Xu,Y. (2017) Fast clustering using adaptive density peak detection. *Stat. Methods Med. Res.*, **26**, 2800–2811.

Wiwie,C. *et al.* (2015) Comparing the performance of biomedical clustering methods. *Nat. Methods*, **12**, 1033–1038.

Xu,R. and Wunsch,D. (2005) Survey of clustering algorithms. *IEEE Trans. Neural Netw.*, **16**, 645–678.

Zachary,W.W. (1977) An information flow model for conflict and fission in small groups. *J. Anthropol. Res.*, **33**, 452–473.

Zelnik-Manor,L. and Perona,P. (2004) Self-tuning spectral clustering. In: Saul,L.K., Weiss,Y. and Bottou,L. (eds) *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04)*, MIT Press, Cambridge, MA, USA, pp. 1601–1608.

Zhang,J. *et al.* (2016) A robust density-based clustering algorithm for multi-manifold structure. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 832–838. doi: 10.1007/978-3-662-46248-5_34.

Zhang,C.T. (1971) Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, **100**, 68–86.