Title

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

Short title

Clustering by search of density valleys

Author names and affiliations

Joao C. Marques[1,2], Michael B. Orger[1]

[1] Champalimaud Research, Champalimaud Centre for the Unknown, Avenida Brasília, Doca de Pedrouços, Lisboa 1400-038, Portugal.

[2] Currently at the Rowland Institute at Harvard, 100 Edwin H. Land Boulevard, Cambridge, MA 02142, USA.

Correspondence: michael.orger@neuro.fchampalimaud.org

Abstract

How to partition a data set into a set of distinct clusters is a ubiquitous and challenging problem. The fact that data sets 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, 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. We propose an alternative approach, which estimates density dips between points, and allows robust determination of cluster number and distribution across a wide array of data, without any manual parameter adjustment. We show that this method is able to solve a range of synthetic and experimental data sets, where the underlying structure is known, and identifies consistent and meaningful clusters in new behavioural data.

Significance

It is common that natural phenomena produce groupings, or clusters, in data, that can reveal the underlying processes. However, the form of these clusters can vary arbitrarily, making it

challenging to find a single algorithm that identifies their structure correctly, without prior knowledge of the number of groupings or their distribution. We describe a simple clustering algorithm that is fully automatic and is able to correctly identify the number and shape of groupings in data of many types. We expect this algorithm to be useful in finding unknown natural phenomena present in data from a wide range of scientific fields.

Introduction

A notable feature in many data sets is that the data points, rather than being evenly distributed, are more densely clustered in some regions of space than others. These clusters may be distributed around single points, or more extended complex shapes. It is often necessary to determine how many clusters exist in a particular data set, and how points are distributed in these clusters, since this structure may reflect the natural processes underlying the data being collected. Unsupervised computational methods that can 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.

Although clustering analysis has a long history, there is no universal consensus on the definition of a cluster or on which clustering algorithm is the most effective (1). In fact, "an impossibility theorem for clustering" was formally proven, showing that there is no single clustering function that can satisfy three fundamental criteria, scale-invariance, richness and consistency (2). In spite of that, it is possible to construct clustering algorithms that have broad applicability by relaxing the proposed criteria. 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 (3, 4).

Whereas some widely used clustering approaches depend on assumptions about the cluster shape, some density-based clustering methods have been proposed that allow clusters of arbitrary shape to be discovered (5). One clustering method, based on search for density peaks (named clusterdp or DensityClust), is fast, resilient to noise, captures clusters of arbitrary shapes, and provides an intuitive method to select the number of clusters that exist in data (6). This method was applied successfully to artificial data sets contaminated with noise, and clusters with fuzzy edges and varying shapes. Also, it performed well in classifying faces from images and, in a meta-study of clustering methods, to solve real world biomedical data (4).

However, in very 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 inherently fails when applied to certain spatial 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 features. We use an adaptive Gaussian density estimator to compute the local densities, in common with other variants of the method (7), 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 centres. Lastly, we implemented methods, based on statistical comparison with reference distributions, to select the number of clusters automatically.

We validated the clusterdv method by applying it, without parameter tuning, to a wide variety of artificial and real-world test data for which the ground truth cluster identity, or underlying data structure, is known. 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 behavioural categories in experimental data from larval zebrafish. Altogether clusterdv is an automatic unsupervised method for density cluster identification, which is generally applicable to different types of data, while simultaneously achieving state-of-the-art performance on a variety of different benchmarks.

Results

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) (6). Both of these values can depend on the choice of d_c, a free parameter in the clustering method which determines the spatial scale for local density calculations. Since the clusterdp method requires the user to select a number of cluster centres based on the distributions of ρ and δ, we developed an automated heuristic for this choice (see Materials and Methods and Fig. S1). In most cases, this method performed as well or better than human observers in selecting the correct number of clusters, when applied to data sets for which ground truth was available (Fig. S2). We took advantage of the good performance of the automatic clusterdp to sample

the d_c parameter for 8 data sets used in the original study and found that the results of automatic clusterdp are highly dependent on the d_c parameter, for all data sets we tested, and there was no single value or range of values that consistently gave correct results (Fig. S3).

Another potential pitfall of the clusterdp is that, if δ uses Euclidean distance as its metric, it will tend to favour 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 data set, 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. 1*A*). This situation is commonly observed in real experimental data sets, 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 clusterdv algorithm is unable to find the low-density cluster without also splitting the single dense cluster into several parts. 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 centres correctly identify both groups (Fig. 1*B*-*D*). This failure to rank the cluster centres correctly does not occur in a data set with similar characteristics, but in which the smaller cluster has higher density (Fig. 1*E*-*H*). To verify that this limitation was not dependent on parameter choice, we systematically varied d_c and plotted the locations of cluster centres for all cases where the correct number (2) was found. In all such cases, for the "exclamation mark 1" data set, the cluster centres were both found in the larger cluster (Fig 1*I*). The opposite result was observed for the "exclamation mark 2" data set where, if the algorithm found the correct number of clusters, they always spanned the two regions (Fig. 1*J*). These results suggest two key limitations of the clusterdp method: a sensitive dependence on parameter choice, and a failure to capture correctly the structure in certain simple data distributions.

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 (8), with the bandwidth at each point based on a heuristic that was applied identically to all data sets (Fig. 2*A*-*B*). Similar adaptive approaches have been used in other clusterdp adaptations (7). This method was applied in a uniform way to different data sets, with no parameter estimation required. Secondly, 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 data set, along which the minimum density was highest. We first estimated the minimum density along a subset of single edges by sampling at discrete intervals from the previous kernel density estimate (Fig. 2*C*). After, we searched for the path joining each pair of points, following any of these lines, that had the smallest drop in density (Fig. 2*D*). It should be possible to join two points that lie in the same cluster by paths whose density profile is always higher than the lower density point. For each point, we can calculate the lowest density that must be passed through to reach a second point of higher density. If this value is lower than the density of the starting point, then that point is at a local maximum, and is a putative cluster centre. Figure 2*E* shows this 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 data set than a small dip separating two very high-density peaks. To capture this distinction, we calculated a separability index (SI), where a 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 centre (Fig. 2*F*). All putative cluster centres (points with positive SI) can now be ranked according to their SI. These points are arranged in a hierarchical tree by connecting each new centre point with the branch it would be assigned to, if the clusters were assigned without using that point (Fig. 3*G*). This SI tree reflects the hierarchical organization of clusters that exist in data. For data sets with complex clustering structure, for example with nested groups of clusters (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 centres, which are "real" (high SI values) and which arise sporadically due to the density estimation of data with finite sample size (low SI values) (blue

line in Fig. 2*F*). We applied clusterdv to the "exclamation mark" data sets and confirmed that the SI value ranks cluster centres correctly for both data sets (Fig. 2*I*-*R*). Additionally, if more than two cluster centres are shown, the data is partitioned in a manner than respects the true cluster boundaries (Fig. 2*M*, *R*), unlike clusterdp (Fig. 1). In summary, clusterdv does not need any parameter optimization to work and reports the ranking of cluster centres correctly, even in difficult cases such as the "exclamation mark 1" data set.

Validation of clusterdv

Clusterdv is able to solve data sets with uneven cluster proportions where the tightest cluster has lower density, but can the correct SI cut-off be determined automatically, and does the method give correct solutions for data with other characteristics? To answer this question, we applied clusterdv to 33 artificial and real-world data sets with known ground truth that were designed to present varying difficult challenges for clustering analysis (Table S1). Two criteria were tested for deciding 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 data sets with similar spatial extent or density profiles, but with only a single density peak in the underlying distribution (Fig. S5 and Fig. S6), and selected all clusters above this threshold (see Materials and Methods for details). The two methods frequently gave the same solution, and, with the exception of the Olivetti face data set (9), always encompassed the known correct solution. The max SI jump, in every case, gave a higher SI cut off than the other criteria and was able to solve correctly 29 of the 33 data sets (Fig. 3). There were only four data sets that 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. It is important to note that, since several of the data sets are real world data sets, it is not clear either that the labeled groups are the only clusters that should exist in the data, or that there should exist a clear distinction between clustered and non-clustered organization, which the SI jump assumes (Table S1). The methods based on reference distributions usually identified more structure, especially in the real-world data sets, as might be expected, but still also had very high success rates (Table S1). In the case of the Olivetti face data set (9), simple observation showed that many of the clusters were fractured in the feature space that was used (10), so it may not be possible to improve on this result without a different representation of the data. Nevertheless clusterdv was able to achieve higher success (Olivetti FMI = 0.75 and Fig. S7), with much larger true associations for a given false positive rate, than previously published results (6) (78 percent at

191 a false alarm rate of 1, versus approximately 65 percent previously reported for clusterdp).
192 Another notable feature of these results is that the correct solution almost always existed
193 within the dendrogram of cluster centres, and in 31 out of 33 cases could be obtained by
194 cutting at a single level (shown in 'manual' column of Table S1). In the case of the MNIST
195 handwritten number data set (11), the ten digit solution existed in the tree, but could not be
196 found using a single SI cut off, without also finding clusters within digits. The 'correct'
197 solution could however be found by choosing the most balanced partition into ten clusters,
198 and this solution could be used as a classifier which gave a 5.9% error rate for new data. This
199 performance is comparable with the best unsupervised methods described for MNIST, that
200 also use information about the construction of the data set, but not individual data labels (5%
201 error rate in (12)) (Fig. S8). Importantly, when the correct number of clusters was identified,
202 the assignment of cluster centres always matched the ground truth, and the assignment of
203 points to clusters was largely correct (Table S1). As an additional comparison with clusterdp
204 we applied this algorithm to the same data sets under the recommended range for the d_c
205 parameter (6) and could only solve correctly 13 (d_c = 1%) and 11 (d_c = 2%) of the 33 data
206 sets (Table S2**,** Table S3). In sum, clusterdv is able to solve correctly a wide range of data
207 without any parameter fitting in a completely automatic fashion. The SI jump criteria shows a
208 performance similar to setting manually the correct number of clusters and all three automatic
209 criteria for clusterdv performed better than clusterdp even when the number of clusters for
210 clusterdp were set manually based on knowledge of the correct value (Fig. S9).

211 **Clusterdv is able to categorize zebrafish startle behaviour**

212 The aim of clustering analysis is to identify groupings on real-world data that correspond to
213 real natural phenomena. We used clusterdv to perform unsupervised categorization of
214 movements of zebrafish larvae responding to acoustic startles. Larvae, which swim in short
215 bouts of movement, execute two types of escapes (C-starts) in response to acoustic startles:
216 one bout type with short latency (SLC) and another with long latency (LLC) (Fig. 4*A*) (13).
217 Critically, both responses differ in a set of kinematic parameters and the neural circuits that
218 produce these behaviours (Fig. 4*B* and Fig. S10) (13). The proportion of both C-starts that
219 fish execute varies with the experimental conditions, creating data sets with very uneven
220 distributions of bouts. Also, the fish do not always respond to the stimulus with C-starts, so
221 these data sets often have bouts that appear in kinematic space as outliers or that degrade
222 cluster boundaries. In spite of these challenges, the SLC responses are normally extremely
223 stereotypical and form a tight cluster that appears distinct from the wide spread cluster that

corresponds to the LLC bout type (Fig. 4*B*). These two swim types are relatively easy to categorize by sorting them by the latency to stimulus onset (red line in Fig. 4*A*) or by setting a threshold that separates them in kinematic space (green line in Fig. 4*B*). If swim bouts of each category are superimposed it becomes apparent that they correspond to two types of movements that are stereotypical within group (Fig. 4*C-D*). We applied clusterdp to this data set and found that it consistently failed to find the 2 correct clusters (Fig. 4*E-H*). As with the "exclamation mark 1" data set the clusterdp rule fails to rank the cluster centres correctly, not identifying the small SLC cluster centre before splitting the larger LLC into several clusters, regardless of the choice of d_c (Fig. 4*I-L*). When we applied clusterdv to this data set, the cluster centre ranking was correct, and the algorithm was able to find essentially the same solution that was found by drawing a line in kinematic space or sorting the swims by latency (compare Fig. 4*B-D* with Fig. 4*N-P*). Clusterdv is thus able to categorize automatically animal behaviour in a data set with uneven sparse data that is corrupted by noise.

Discussion

We have described here a novel, robust and simple density based clustering algorithm, clusterdv, based on the density valleys between data points, that is applicable to a wide variety of data. It delivers consistently better results than an automatic version of clusterdp (6). In particular, it is able to reliably find tight clusters of low density, where clusterdp fails because its rule gives more importance to distance than to gaps between clusters. Thus, our fully automatic version of clusterdv outperforms clusterdp across a wide variety of data sets, even in the case where the number of cluster centres is chosen manually for clusterdp based on prior knowledge of the data structure.

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. To deal with this problem clusterdv produces a hierarchical tree of "putative" cluster centres and uses an intuitive metric, the separability index or SI to rank their importance. The number of "putative" cluster centres 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 centres. To determine how many clusters exist in the data, it is necessary to decide which cluster centres are likely to be genuine, and which may occur sporadically due to sampling error, by setting an appropriate threshold on the SI value, or to cut the SI dendrogram at a particular set of nodes. We automated this step of the algorithm by

developing data-based criteria to choose the number of clusters. One such criteria, selection of the largest jump in separability index, correctly determined the number of clusters in 29 out of 33 data sets, 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 data sets, it is not clear that this assumption will always be correct. Therefore, to be applied in real-world data sets, such as the zebrafish swimming data we have described here, we developed two other criteria, termed onion and simplex, that are based on constructing reference distributions based on the original data that only have one density peak, similar to the gap statistic method (14). These reference distributions are used to measure the probability that "sporadic" cluster centres may arise in that particular set of data. Often the "onion" and "simplex" methods also gave the correct solution for many of the data sets we tested (14 and 21 correct data sets 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 data sets, 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 (15-18), number (19) and spatial distribution (20), clusters with fuzzy edges (21, 22), data with multiple dimensions (23, 24), corruption with noise (6, 25), and distributions with uneven proportions of clusters; and did not need to adjust any parameter to solve a particular data set. Nevertheless, there are parameters that can be set in clusterdv, if desired. One such parameter, the number of divisions of the density lines, allows a trade-off of computational time vs accuracy, and needs to be set sufficiently large not to degrade the results. The density estimation by Gaussian mixture (8) may be performed using distinct methods or rules, but we found that the one simple heuristic used here always gave satisfactory results. Other methods have been proposed that aim to improve on, or automate,

aspects of clusterdp, (eg (7, 26)), but, to our knowledge, none has been demonstrated to work consistently well across different data sets without parameter tuning.

Finally, we applied clusterdv to the difficult problem of unsupervised behavioural categorization. We created a zebrafish larvae behavioural data set that is sparse and composed of highly uneven clusters that are plagued with noise, but is known to contain two distinct swim categories (13). Clusterdv could identify, in a completely automatic fashion, meaningful behavioural 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 data set, 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 data sets, 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.

Materials and Methods

Animal care

Fish were reared on a 14/12 hr light/dark cycle at 28 °C on sets of 20 in E3 water as described previously (27). Wildtype Tübingen zebrafish larvae (6-7 days post fertilization) were used for behavioural experiments. Animal handling and experimental procedures were approved by the Champalimaud Foundation Ethics Committee and the Portuguese Direcção Geral Veterinária and were performed according to the European Directive 2010/63/EU.

Behavioural assays

Behavioural recordings were performed in acrylic transparent 5 x 5 cm arenas in groups of 7 fish. 100 ms tones with fixed frequency (200 to 2000 hz) where delivered, every 2 min, by two HP mini speakers (Hewlett-Packard Company) attached to the arena. The fish's movements were recorded at 700 frames per second using a custom made behaviour tracking system composed of a high-speed infrared sensitive camera (MC1362, Mikrotron) fitted with a Schneider apo-Xenoplan 2.0/24 lens. Fish were illuminated by a 10 x 10 cm LED-based diffusive backlight (850 nm, Nerlite) placed below the fish arena and the camera was fitted with a 790 nm long pass filter that blocked visible light. Fish and tail tracking were

performed online by custom written programs using C# (Microsoft) and the OpenCV imaging processing library as described previously (28). Kinematic parameters and swim bout detection were performed using custom written Matlab made programs (29). A subset of 550 bouts were randomly picked to construct the "beep20" data set.

Density valley clustering algorithm. Matlab code for running the clusterdv algorithm, together with example data sets, is available at https://github.com/jcbmarques/clusterdv. The local densities are estimated from a KDE (Kernel density Estimate) calculated with Gaussian kernels, using a Matlab-based toolbox developed by Alexander Ihler (http://www.ics.uci.edu/~ihler/code/kde.html). The bandwidth for each data point was chosen using a leave-one-out maximum likelihood estimate, with all the bandwidths constrained to be proportional to the kth nearest neighbour distance where:

$$k = \sqrt{N} \qquad (1)$$

and N is the number of data points. We calculate the density valley depth between points by using the KDE to estimate the density at fixed intervals along straight lines connecting pairs of points and finding the minimum value along each line. We limited the computational time of this step for larger data sets, by restricting the calculation to three sets of edges: 1) the minimal spanning tree found using Kruskal's algorithm (30), 2) edges connecting each point to the nearest point of higher density, and 3) edges connecting each point to the nearest $\sqrt{N}$ points. To identify the smallest drop in density on any path connecting two points, we applied the single link algorithm to the matrix of values (31). Based on that value we calculate the separability index (SI) for each point by:

$$SI = \frac{-Maximum\ density\ valley}{\rho} + 1 \qquad (2)$$

where ρ is the local density at that point. Points with positive SI values represent putative cluster centres, with values ranging from 0 (not separated from other cluster centres) to 1 (completely separated from other clusters).

Generation of reference distributions. We used two methods to generate reference distributions that aimed to match features of the original distributions, but should only have a single density peak. The "simplex" method aims at creating a reference distribution within the same convex hull as the original data but without multiple density peaks. Briefly, every set of D + 1 points in the data set, where D is the number of dimensions of the data, defines a simplex. We sample each reference point randomly from within one of these simplexes,

which is itself chosen randomly with a weighting proportional to its volume. This has the effect of sampling each simplex at equal density, resulting in a summed distribution which has only one peak. This produces a distribution that lies within the convex hull of the original data, and therefore has a similar range of pairwise point distances. However, the range of local point densities can be very different, which may not be ideal for the clusterdv method, for which these densities define the x-axis range of the decision plot. Therefore, for the "onion" method we aim to match not the shape, but the distribution of point densities in the original data. We make the "onion" reference distribution by first calculating the local density at each of the original data points, and arranging them in descending order. We then choose the first point of the reference distribution randomly from within an n-sphere, whose radius is chosen so that the local density matches that of the densest point. Subsequent points are then chosen from n-spherical shells surrounding the previous level, whose outer radii are chosen to match the local density within each shell with the density of the corresponding point in the original distribution. The "onion" method creates a spherically symmetrical reference distribution with a single local maximum, and a distribution of density that is very similar to the original data.

Determination of number of clusters for clusterdv.

Data points with positive SI value were considered 'putative' cluster centres. Their SI value was used to construct dendrograms that reflect the ranking of cluster centres. As cluster centres with decreasing SI are added to the dendrogram, they are connected to the cluster centre with which they co-partition at the next higher level. For all data sets 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 separability index (SI jump) between successive cluster centres. Alternatively, we take the 95th percentile of the second highest SI value from 100 reference distributions computed using the "simplex" or the "onion" methods. All cluster centres with higher SI values than each of these criteria formed the associated solutions for each data set.

Automatic cluster centre identification for clusterdp. The local densities (ρ) and δ value where calculated for every data point as described in (6). By definition $\gamma_i = \rho_i\delta_i$ will be large for cluster centres that belong to clusters that are well separated and have many points. We rely on this quantity to find a "jump" on the data that signals when data points stop being cluster centres and start being data points belonging to an already identified cluster, or

outliers. 100 unimodal reference distributions using the "simplex" resampling method were created. This method was used to create unimodal distributions because clusterdp uses pairwise distances to calculate its two measures and the distances between points are highly influenced by the shape of the distribution. Next, we calculate the γ measure for the real data and the reference distributions, sort these values, average the γ sorted values of the reference distributions and subtract those from the γ sorted values of the original data. To find the appropriate number of clusters we determined how much the γ difference values deviate from the reference distributions. To determine this the γ difference values are divided by the values of the original distribution with all values made positive by subtracting the minimum (normalized γ differences). The cluster centres were found by choosing the points before the largest jump in the normalized γ values.

Analysis of the MNIST data set. The test data of the MNIST digit data set was placed in a set into a three-dimensional space obtained by parametric t-SNE (perplexity = 30) (32). For manual selection of the SI cut-off, two methods were used. First the tree was cut at a level which gave 10 clusters. Alternatively, the 'balanced' cut was found by starting with the 10 cluster solution and lowering the threshold on the branch with the largest number of points, until another centre was found, and pruning the cluster centre with the smallest number of points. This process was repeated until the standard deviation of cluster sizes was at a minimum. The resulting cluster assignments were then used to classify the 60000 remaining digits via a nearest neighbour method.

Data point assignment. For all data sets the non-cluster centre points were assigned, sequentially and in decreasing density order, to the same cluster of the nearest neighbour of higher density (6).

Automatic clusterdp validation. To determine the effectiveness of the automatic clusterdp we compared the algorithm's success in finding the correct number of clusters to the ability of people in manually picking cluster centres. d_c values from 1 to 100 % were applied to nine synthetic data sets and their decision plots (ρ vs δ) were shown to 9 unpaid volunteers by using a custom made Matlab script (MathWorks). The participants were asked to pick the cluster centres that they thought were present in the decision plots by drawing a square around them. The order of display of the decision plots was randomized. The people were not given feedback over their choices or informed of the number of clusters present in the data sets. To each person it was explained how the clusterdv algorithm works and how to draw a

square using the Matlab custom program (MathWorks). For comparison of two independent groups, significance of difference was tested with the Mann–Whitney test. Statistical analysis was performed using the Statistic Toolbox (MathWorks). Differences were considered significant when $P < 0.05$.

Figure Legends

Fig. 1. Density peak clustering fails with uneven clusters. (*A*, *E*) The "exclamation mark 1 and 2" data sets were drawn from two part probability distributions (left). White represents 2.5-fold higher probability than grey and black is probability 0. (*B*, *F*) Clusterdp decision plot (ρ vs δ) of the distribution in (*A*, *E*). (*C*, *G*) Clusterdp solutions of data in (*A*, *E*) by picking the two (left) or three (right) cluster centres with highest γ (δ*ρ). (*D*, *H*) Density profile of data in (*A*, *E*) (d_c = 9%). (*I*-*J*) Left: Number of cluster centres picked by automatic clusterdp in function of d_c value for the "exclamation mark 1" data set (*I*) and the "exclamation mark 2" data set (*J*). Red outlines mark the ground truth (2 clusters). Right: the cluster centres obtained by automatic clusterdp whenever the two cluster solution was selected. Blue is cluster centre with higher γ value, cyan is cluster centre with 2nd γ value.

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 are calculated, in a set of discrete steps, along straight lines between pairs of points. (*D*) The single link algorithm is used to determine the highest minimum density (white point) to get from one point to another, via this set of straight lines. (*E*) Maximum density valley versus local density (ρ). For each point, the maximum density valley is the highest such value connecting that point to a point of higher density. (*F*) Separability index (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. (*G*) Dendrogram computed from separability index. (*H*) Cluster assignment of the point distribution in (*A*) obtained by choosing the cluster centres with higher SI value than the blue line in (*G*). (*I*-*R*) (left to right) Distribution, clusterdv decision plot (SI vs ρ), dendrogram and two and three cluster solutions for the exclamation mark 1 (*I*-*M*) or 2 (*N*-*R*) data sets. Cluster centres and points are color coded blue-cyan-red in order of decreasing SI value.

Fig. 3. Clusterdv gives the correct solution for artificial and real-world data sets. (*A*-*U*) Synthetic data sets: (*A*-*C*, *T*) data sets from this study, (*D*-*F*) (19), (*G*-*J*) (22), (*K*-*L*) (6), (*M*) (15), (*N*) (4), (*O*) (16), (*P*) (17), (*Q*) (4), (*R*) (18), (*S*) (4), (*U*) (20). (*V*-*X*) Real-world data

sets: (*V*) seeds (23), (*W*) bone marrow (21), (*X*) Zachary (24). Left panels: cluster assignment of the data sets according to the SI jump criteria. Right panels: SI dendrograms. Red line, SI jump threshold. Orange line, simplex threshold. Green line, onion threshold.

Fig. 4. Clusterdv enables unsupervised categorization of zebrafish acoustic startle behaviour. (*A*-*D*) Manual categorization of larval acoustic startle swim bouts. (*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*) PCA of set of swim bout kinematic parameters. Red points correspond to short latency bouts, while black points correspond to long latency bouts categorized by red line in (*A*). Green line separates the small SLC cluster from the larger LLC cluster. (*C*-*D*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked SLC (*C*) and LLC bouts (*D*) categorized according to green line in (*B*). Cyan lines are the average of all bouts in each category. (*E*) Clusterdp decision plot (ρ vs δ) of bout distribution in (*B*). (F) Assignment of bout distribution in (*B*) by manually picking cluster centres (red square in (*E*)). (*G*-*H*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked bouts from solution in (*F*). (*I*) Locations of cluster centres in data picked manually by humans for different d_c values. Colors represent d_c values according to legend. (*J*) Clustering solutions for automatic clusterdp in function of d_c values. Red rectangle shows the known 2-cluster solution. (*K*-*L*) Locations of cluster centres in data set obtained from automatic clusterdp solutions for two cluster solutions (*K*) and three cluster solutions (*L*). Colors in (*K*) represent d_c values as in legend and in (*L*) indicate the γ ranking. (*M*-*P*) Clusterdv finds the solution that is consistent with experimental observations. (M) Clusterdv SI dendrogram for data in (*B*). Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. (N) Clusterdv assignment according to the simplex threshold in (*M*). (*O*-*P*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked SLC (*O*) and LLC bouts (*P*) categorized according to orange line in (*M*).

Fig. S1. Automatic clusterdp method. Our automatic method to determine the number of cluster centres for the density peak algorithm (see Materials Methods for details) (6). (*A*) Point distribution drawn from a mixture of five gaussians (black). In green, a reference distribution from within the convex hull of the original distribution obtained by the "simplex" method. (*B*-*F*) Red points correspond to cluster centres found by the algorithm. (*B*) Decision plot (δ vs ρ) of the two distributions in (*A*). (*C*) Difference between the γ values of the gaussian mixture distribution and the average values of 100 reference distributions, the points are ordered for decreasing values of γ difference values. (*D*) Normalized γ difference values.

(*E*) Difference of values in (*D*), or "jump". The minimum point is used to choose the number of cluster centres. (*F*) Point assignment of black data set in (*A*) according to the number of clusters found in (*E*). The five clusters found in (*F*) correspond to the five gaussians that form the (*A*) distribution.

Fig. S2. Automatic clusterdp performance. We compared the performance of automatic clusterdp to pick the correct number of clusters to the human ability to manually pick cluster centres (see Materials and Methods for details). (*A*, *C-D*) This study, (*B*) (17), (*E-F*) (6), (*G*) (16), (*H*) (18), (*I*) (20), and (*J*) all data sets together. Error bars are standard error of the mean. Mann–Whitney test, * $p < 0.05$, auto n = 10, manual n = 9.

Fig. S3. Accuracy of the clusterdp algorithm is critically dependent on the choice of parameter d_c. Left panels represent synthetic point distributions used in (6): (*A*) (20), (*B*) (17), (*C-D*) (6), (*E*) (16), (*F)* (18), (*G*) (22), and (*H*) (15). White lines in right panels correspond to number of clusters found using automatic clusterdp for a particular d_c value expressed in percentile of all pairwise distances. The red outlines mark the column with the correct number of clusters.

Fig. S4. Solution of clusterdv for data set with nested groups of clusters. (*A*) Point distribution drawn from mixture of nine gaussians. (*B*) Local densities are calculated using adaptive gaussian density estimator. (*C*) SI versus local density (ρ). Green points are from example reference distribution calculate using the onion method. (*D*) SI dendrogram. (*C-D*) Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. (*E*) SI jump versus cluster centres sorted by decreasing SI value. (*F*) Cluster assignment of the point distribution in (*A*) using the SI jump or onion threshold cut-offs.

Fig. S5. Methods to calculate reference distributions. (*A*) Synthetic point distribution drawn from a mixture of five gaussians. (*B*) Reference distribution obtained by resampling the distribution in (*A*) using the simplex method. (*C*) Reference distribution obtained by resampling the distribution in (*A*) using the onion method. (*A-C*) Colours represent local densities calculated by using an adaptive mixture of gaussians density estimate. (*D*) Average local density between the three distributions in (*A-C*). (*E*) Distance between points plotted in descending order. (*F*) Local densities plotted in descending order. (*E-F*) Black, original distribution. Green, simplex distribution. Red, onion distribution.

Fig. S6. Simplex method example reference distributions. Black points are original point distributions. Green points are "simplex" reference distributions. Synthetic point distributions from: (*A*-*B*) (6), (*C*-*D*, *H*) this study, (*E*) (15), (*F*) (16), (*G*) (22), (*I*) (18), and (*J*) (20).

Fig. S7. Density valley clustering of the Olivetti face data set. Complex wavelet structural similarity (CW-SSIM) index (10) was applied to the Olivetti face data set (9) to calculate similarities between pairs of images. Since clusterdv is not a distance based clustering method we applied t-SNE to reduce the dimensionality of the similarity matrix (perplexity 30). (*A*-*B*) CW-SSIM embedded in a two-dimensional T-SNE space. Point colours correspond to ground truth (*A*) and clusterdv solution of 40 clusters (*B*). (*C*) SI dendrogram of data in (*A*-*B*). (*D*) Fraction of pair of images correctly associated to the same cluster (rTrue) as a function of the fraction of pair of images erroneously assigned to the same cluster (rFalse), using different cut-off values. (*C*-*D*) Red, SI jump. Orange, simplex threshold. Green, onion threshold. Cyan, manually picking 40 clusters with highest SI value. (*E*) Histogram of percent matching faces for the largest group in each cluster for the 40 cluster centres solution. (*F*) Assignment of the Olivetti data set for the 40 cluster centres solution. Each picture is labelled with the colour of the cluster it was assigned to.

Fig. S8. Density valley clustering of the MNIST handwritten digit data set. Parametric t-SNE was used to embed the raw data of the test MNIST data set into a three-dimensional space (perplexity = 30) (32). (*A*) SI dendrogram of data in (*B*-*E*). Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. Cyan line manually picking 10 clusters with higher SI values. (*B*) Colors represent ground truth according to legend. (*C*) Solution given by SI jump. Most clusters are correct with the exception of clusters of digits 9 and 4 that were grouped together because these clusters are not well separated in the t-SNE space. Compare clusters 4 and 9 in (*B*) with dark orange cluster in (*C*). (*D*) Solution by finding clusters in dendrogram by minimizing standard deviation of number of points per cluster (see Materials and Methods for details). (*E*) Solution given by manually picking 10 clusters with a single cut-off. Similar to the SI jump solution but cluster 2 was divided into two clusters (compare cluster 2 in (*B*) with blue and magenta cluster in (*E*)). (*F*) Fifty digits were picked randomly from solution in (*D*).

Fig. S9. Performance comparison between clusterdp and clusterdv. The clusterdp and clusterdv were applied to 33 data sets (for references of datasets see Tables S1-3). For clusterdp the d_c parameter of 1% and 2% were used (numbers in legends) and the number of

clusters were chosen automatically (auto, see Fig. S1 for details) or set by hand to the known number of clusters by choosing the points with the highest γ values (manual). The number of clusters for clusterdv were chosen automatically by the largest SI jump, simplex or onion criteria or manually by choosing the number of clusters in ground truth with the highest SI value. 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.

Fig. S10 | Short latency and long latency bout types have distinct distributions of kinematic parameters. (*A*) Swim bout data set obtained by larvae responding to acoustic stimuli. Colors represent bout categorization obtained by clusterdv (Fig. 4*M-P*). (*B-H*) Distributions were obtained by using all bouts in (*A*). SLC and LLC bout types show different kinematic parameter values that are in accordance with previous reports (13). Colours of distributions correspond to clusters in (*A*).

Author contributions

JCM and MBO designed the project, developed the clusterdv algorithm, collected data, analysed data and wrote the manuscript.

Acknowledgements

JCM 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). We thank Alfonso Renart, Megan Carey, Christian Machens and Gonzalo de Polavieja for helpful discussions during the project.

References

1. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM computing surveys (CSUR)*.
2. Kleinberg J (2002) An impossibility theorem for clustering. *NIPS*.
3. Xu R, WunschII D (2005) Survey of Clustering Algorithms. *IEEE Trans Neural Netw*

16(3):645–678.

4. Wiwie C, Baumbach J, Röttger R (2015) Comparing the performance of biomedical clustering methods. *Nat Meth* 12(11):1033–1038.

5. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*.

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

7. Wang XF, Xu Y (2015) Fast clustering using adaptive density peak detection. *Statistical Methods in Medical Research*.

8. Breiman L, Meisel W, Purcell E (1977) Variable Kernel Estimates of Multivariate Densities. *Technometrics* 19(2):135–144.

9. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*.

10. Sampat MP, Wang Z, Gupta S, Bovik AC, Markey MK (2009) Complex wavelet structural similarity: a new image similarity index. *IEEE Trans Image Process* 18(11):2385–2401.

11. LeCun Y, Bottou L, Bengio Y (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

12. Chen X, Duan Y, Houthooft R, Schulman J (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *NIPS.*

13. Burgess HA, Granato M (2007) Sensorimotor Gating in Larval Zebrafish. *Journal of Neuroscience* 27(18):4984–4994.

14. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society B*.

15. Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. *ACM Trans Knowl Discov Data* 1(1):4–es.

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

17. Jain A, Law M (2005) Data clustering: A user's dilemma. *Pattern Recognition and Machine Intelligence*. pp. 1-10

18. Chang H, Yeung DY (2008) Robust path-based spectral clustering. *Pattern Recognition*. 41(1):191-203

19. Karkkainen I, Franti P (2002) Dynamic local search for clustering with unknown number of clusters. *Pattern Recognition*.

20. Veenman CJ, Reinders M (2002) A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(9):1273–1280.

21. Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*. 52: 91-113

22. Fränti P, Virmajoki O (2006) Iterative shrinking method for clustering problems. *Pattern Recognition* 39(5):761–775.

23. Charytanowicz M, Niewczas J, Kulczycki P (2010) Complete gradient clustering algorithm for features analysis of x-ray images. *Information Technologies in Biomedicine.* 69: 15-24

24. Zachary WW (1977) An information flow model for conflict and fission in small groups. *Journal of anthropological research*. 33(4): 452-473

25. Karypis G, Han EH, Kumar V (1999) Chameleon: Hierarchical clustering using dynamic modeling. IEEE *Computer*. 32(8): 68-75

26. Zhang J, Pechenizkiy M, Pei Y, Efremova J (2016) A robust density-based clustering algorithm for multi-manifold structure. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. pp 832–838.

27. Martins S, et al. (2016) Toward an Integrated Zebrafish Health Management Program Supporting Cancer and Neuroscience Research. *Zebrafish* 13(S1):S–47–S–55.

28. Severi K, et al. (2014) Neural control and modulation of swimming speed in the larval zebrafish. *Neuron*. 83(3): 692-707

29. Marques J, Lackner S, Félix R, Orger M Hierarchical structure of the zebrafish locomotor repertoire revealed with unsupervised behavioural clustering. *Submitted*.

30. Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*. 7(1): 48-50.

31. Sibson R (1973) SLINK: an optimally efficient algorithm for the single-link cluster method. *The computer journal*. 16(1): 30-34

32. van der Maaten L (2009) Learning a parametric embedding by preserving local structure. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS). pp.* 384-391.

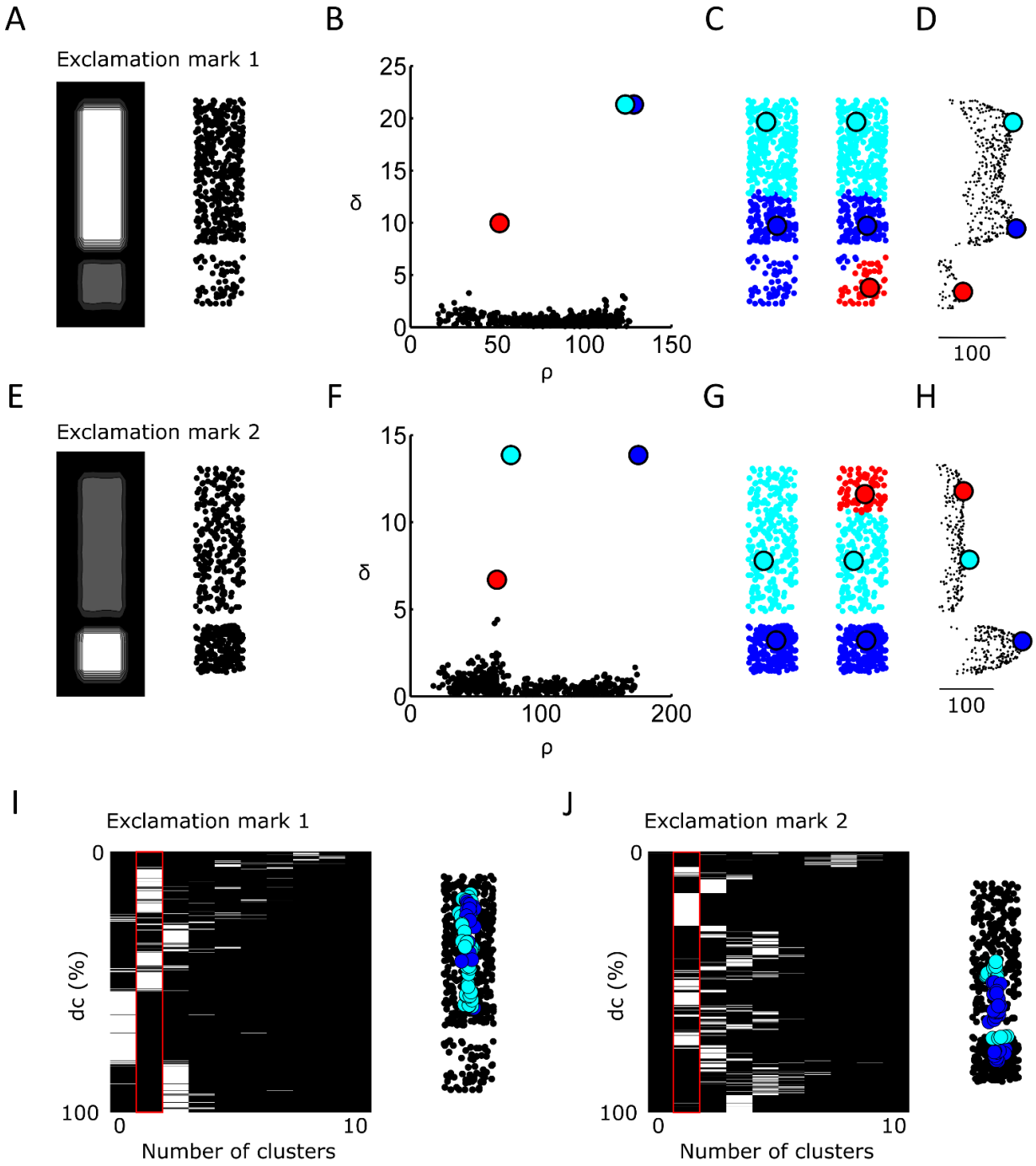

Fig. 1. Density peak clustering fails with uneven clusters. (*A*, *E*) The "exclamation mark 1 and 2" data sets were drawn from two part probability distributions (left). White represents 2.5-fold higher probability than grey and black is probability 0. (*B*, *F*) Clusterdp decision plot (ρ vs δ) of the distribution in (*A*, *E*). (*C*, *G*) Clusterdp solutions of data in (*A*, *E*) by picking the two (left) or three (right) cluster centers with highest γ (δ*ρ). (*D*, *H*) Density profile of data in (*A*, *E*) (d_c = 9%). (*I*-*J*) Left: Number of cluster centers picked by automatic clusterdp in function of d_c value for the "exclamation mark 1" data set (*I*) and the "exclamation mark 2" data set (*J*). Red outlines mark the ground truth (2 clusters). Right: the cluster centers obtained by automatic clusterdp whenever the two cluster solution was selected. Blue is cluster center with higher γ value, cyan is cluster center with 2nd γ value.

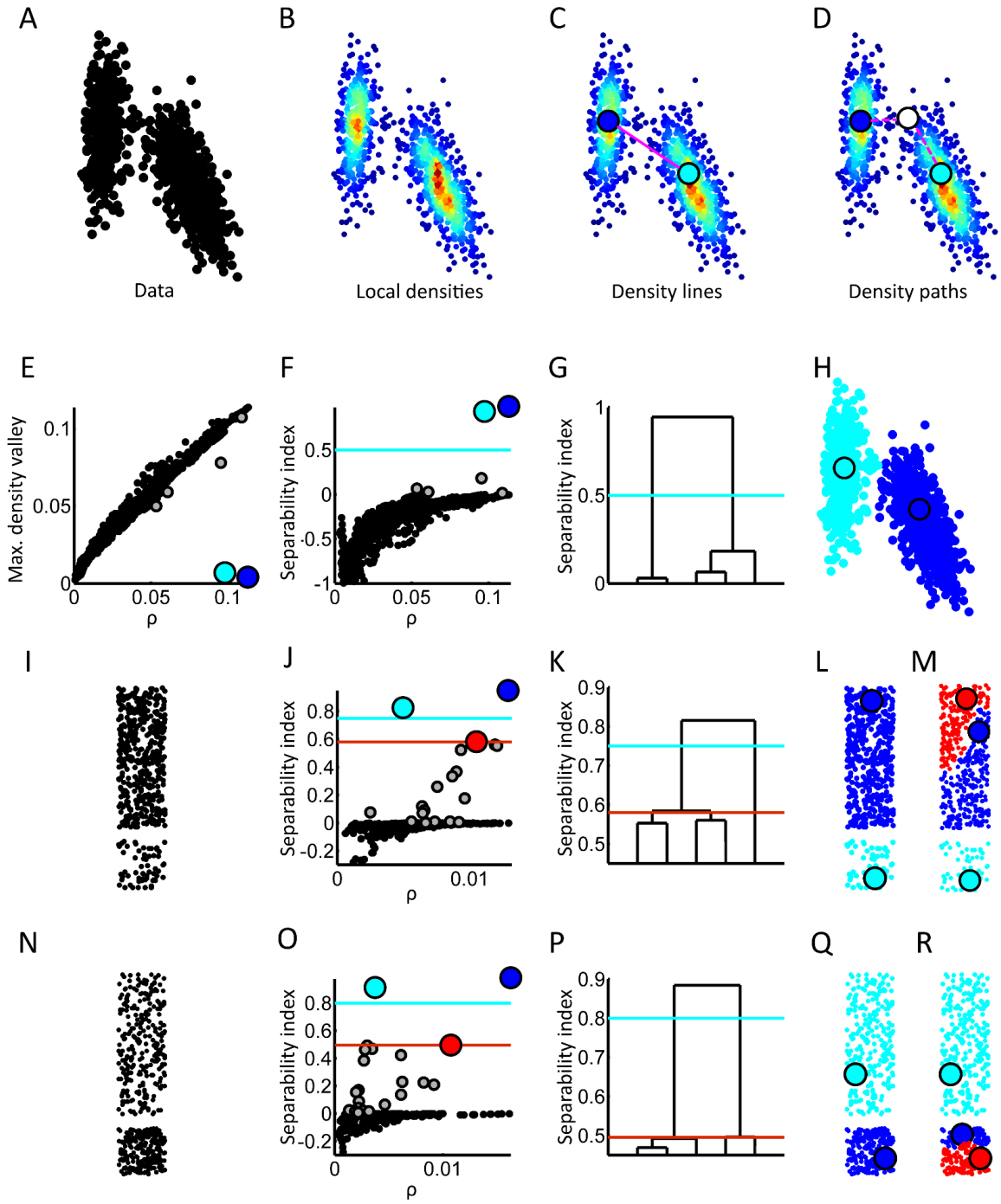

Fig. 2. The clusterdv method. (*A*) Point distribution drawn from a mixture of two gaussians. (*B*) Local densities are calculated using an adaptive gaussian density estimator. (*C*) Density profiles are calculated, in a set of discrete steps, along straight lines between pairs of points. (*D*) The single link algorithm is used to determine the highest minimum density (white point) to get from one point to another, via this set of straight lines. (*E*) Maximum density valley versus local density (ρ). For each point, the maximum density valley is the highest such value connecting that point to a point of higher density. (*F*) Separability index (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. (*G*) Dendrogram computed from separability index. (*H*) Cluster assignment of the point distribution in (*A*) obtained by choosing the cluster centers with higher SI value than the blue line in (*G*). (*I-R*) (left to right) Distribution, clusterdv decision plot (SI vs ρ), dendrogram and two and three cluster solutions for the exclamation mark 1 (*I-M*) or 2 (*N-R*) data sets. Cluster centers and points are color coded blue-cyan-red in order of decreasing SI value.

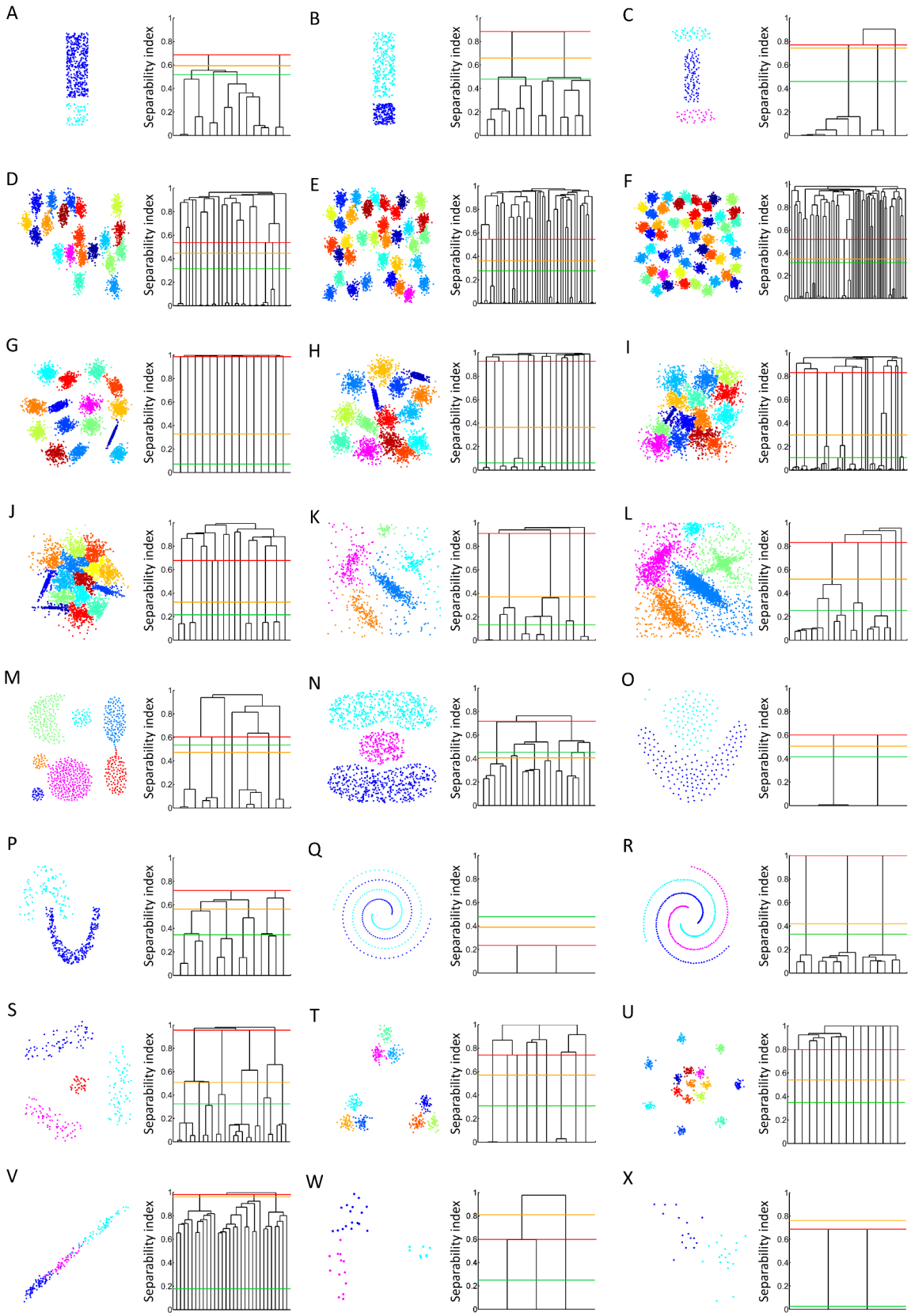

Fig. 3. Clusterdv gives the correct solution for artificial and real-world data sets. (*A*-*U*) Synthetic data sets: (*A*-*C*, *T*) data sets from this study, (*D*-*F*) (19), (*G*-*J*) (22), (*K*-*L*) (6), (*M*) (15), (*N*) (4), (*O*) (16), (*P*) (17), (*Q*) (4), (*R*) (18), (*S*) (4), (*U*) (20). (*V*-*X*) Real-world data sets: (*V*) seeds (23), (*W*) bone marrow (21), (*X*) Zachary (24). Left panels: cluster assignment of the data sets according to the SI jump criteria. Right panels: SI dendrograms. Red line, SI jump threshold. Orange line, simplex threshold. Green line, onion threshold.

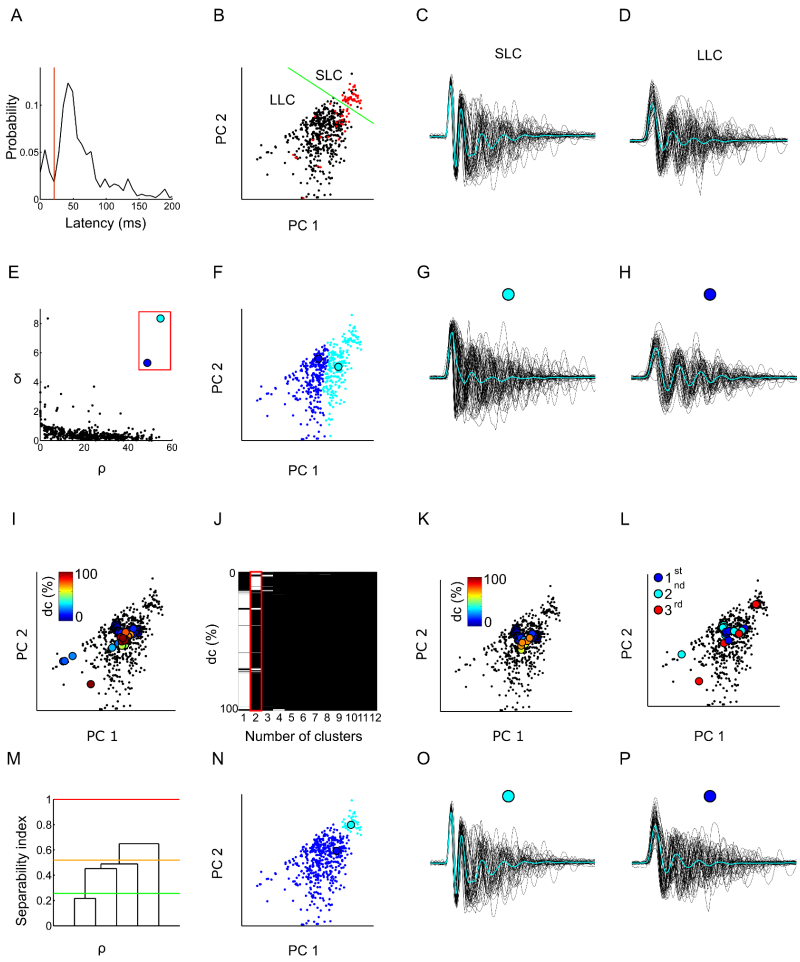

Fig. 4. Clusterdv enables unsupervised categorization of zebrafish acoustic startle behaviour. (*A-D*) Manual categorization of larval acoustic startle swim bouts. (*A*) Swim bout latency from start of acoustic startle. Red line divides swims bouts with short latency (SLC) from bouts with long latency (LLC). (*B*) PCA of set of swim bout kinematic parameters. Red points correspond to short latency bouts, while black points correspond to long latency bouts categorized by red line in (*A*). Green line separates the small SLC cluster from the larger LLC cluster. (*C-D*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked SLC (*C*) and LLC bouts (*D*) categorized according to green line in (*B*). Cyan lines are the average of all bouts in each category. (*E*) Clusterdp decision plot (ρ vs δ) of bout distribution in (*B*). (F) Assignment of bout distribution in (*B*) by manually picking cluster centers (red square in (*E*)). (*G-H*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked bouts from solution in (*F*). (*I*) Locations of cluster centers in data picked manually by humans for different d_c values. Colors represent d_c values according to legend. (*J*) Clustering solutions for automatic clusterdp in function of d_c values. Red rectangle shows the known 2-cluster solution. (*K-L*) Locations of cluster centers in data set obtained from automatic clusterdp solutions for two cluster solutions (*K*) and three cluster solutions (*L*). Colors in (*K*) represent d_c values as in legend and in (*L*) indicate the γ ranking. (*M-P*) Clusterdv finds the solution that is consistent with experimental observations. (M) Clusterdv SI dendrogram for data in (*B*). Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. (N) Clusterdv assignment according to the simplex threshold in (*M*). (*O-P*) Angle of caudal tail segment (°) versus time (ms) of fifty randomly picked SLC (*O*) and LLC bouts (*P*) categorized according to orange line in (*M*).

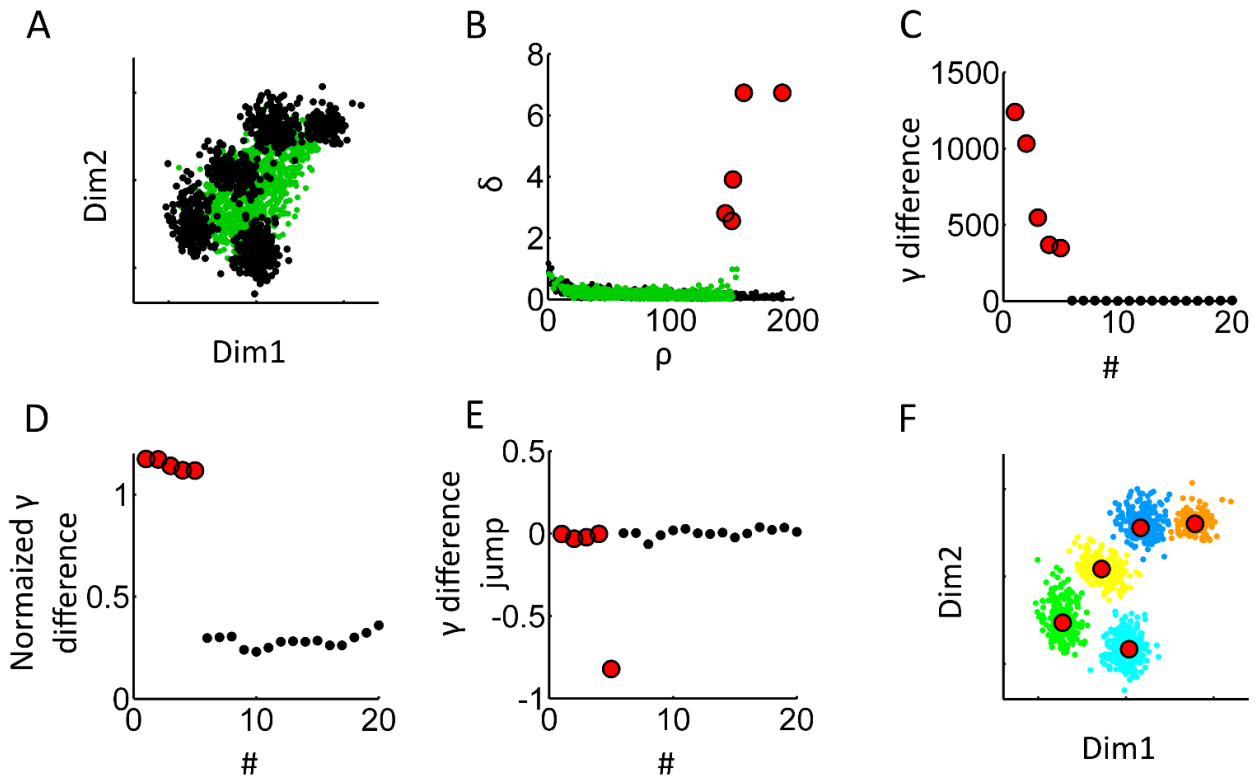

Fig. S1. Automatic clusterdp method. Our automatic method to determine the number of cluster centres for the density peak algorithm (see Materials Methods for details) (6). (*A*) Point distribution drawn from a mixture of five gaussians (black). In green, a reference distribution from within the convex hull of the original distribution obtained by the "simplex" method. (*B-F*) Red points correspond to cluster centres found by the algorithm. (*B*) Decision plot (δ vs ρ) of the two distributions in (*A*). (*C*) Difference between the γ values of the gaussian mixture distribution and the average values of 100 reference distributions, the points are ordered for decreasing values of γ difference values. (*D*) Normalized γ difference values. (*E*) Difference of values in (*D*), or "jump". The minimum point is used to choose the number of cluster centres. (*F*) Point assignment of black data set in (*A*) according to the number of clusters found in (*E*). The five clusters found in (*F*) correspond to the five gaussians that form the (*A*) distribution.

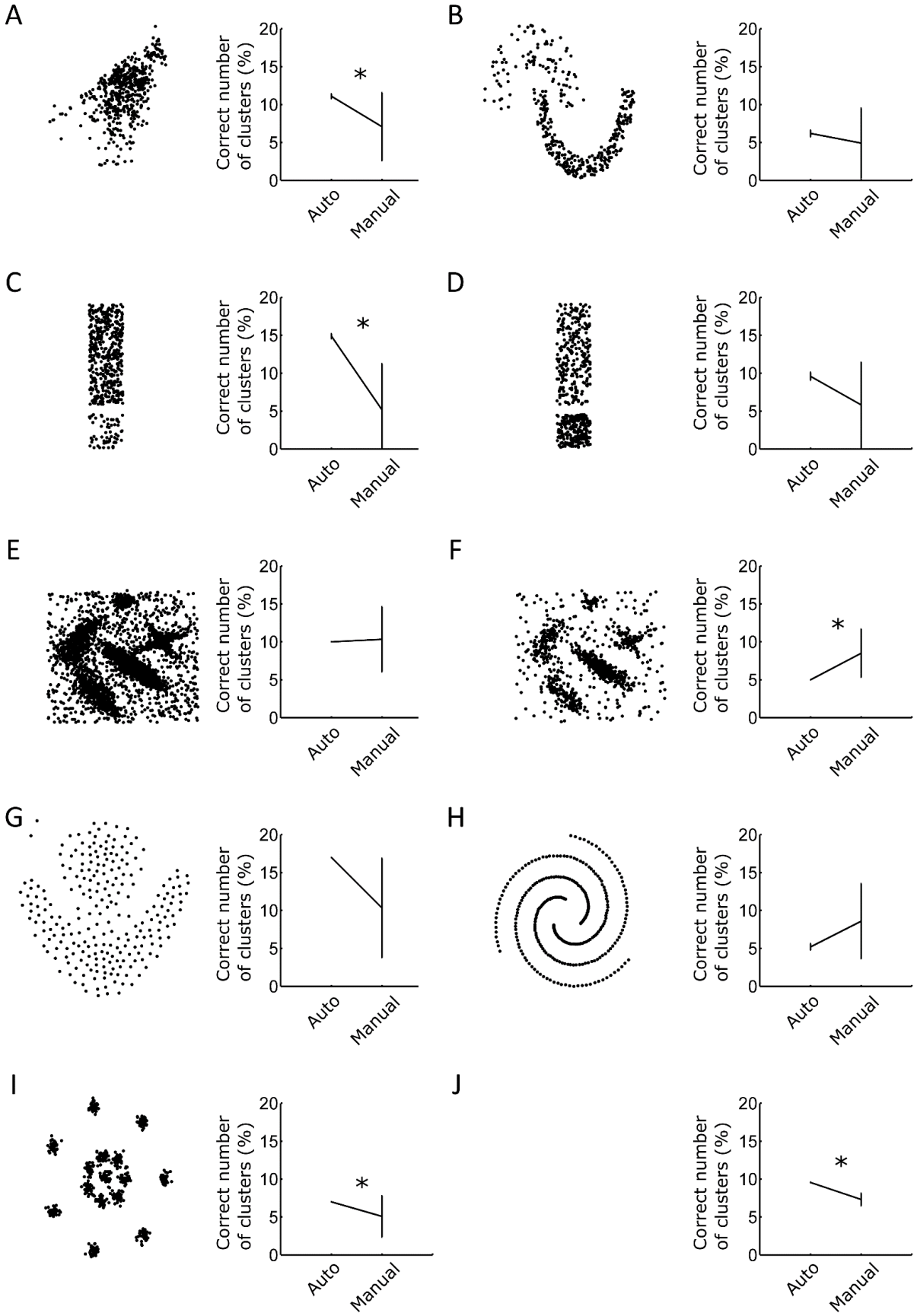

Fig. S2. Automatic clusterdp performance. We compared the performance of automatic clusterdp to pick the correct number of clusters to the human ability to manually pick cluster centers (see Materials and Methods for details). (*A*, *C-D*) This study, (*B*) (17), (*E-F*) (6), (*G*) (16), (*H*) (18), (*I*) (20), and (*J*) all data sets together. Error bars are standard error of the mean. Mann–Whitney test, * $p < 0.05$, auto n = 10, manual n = 9.

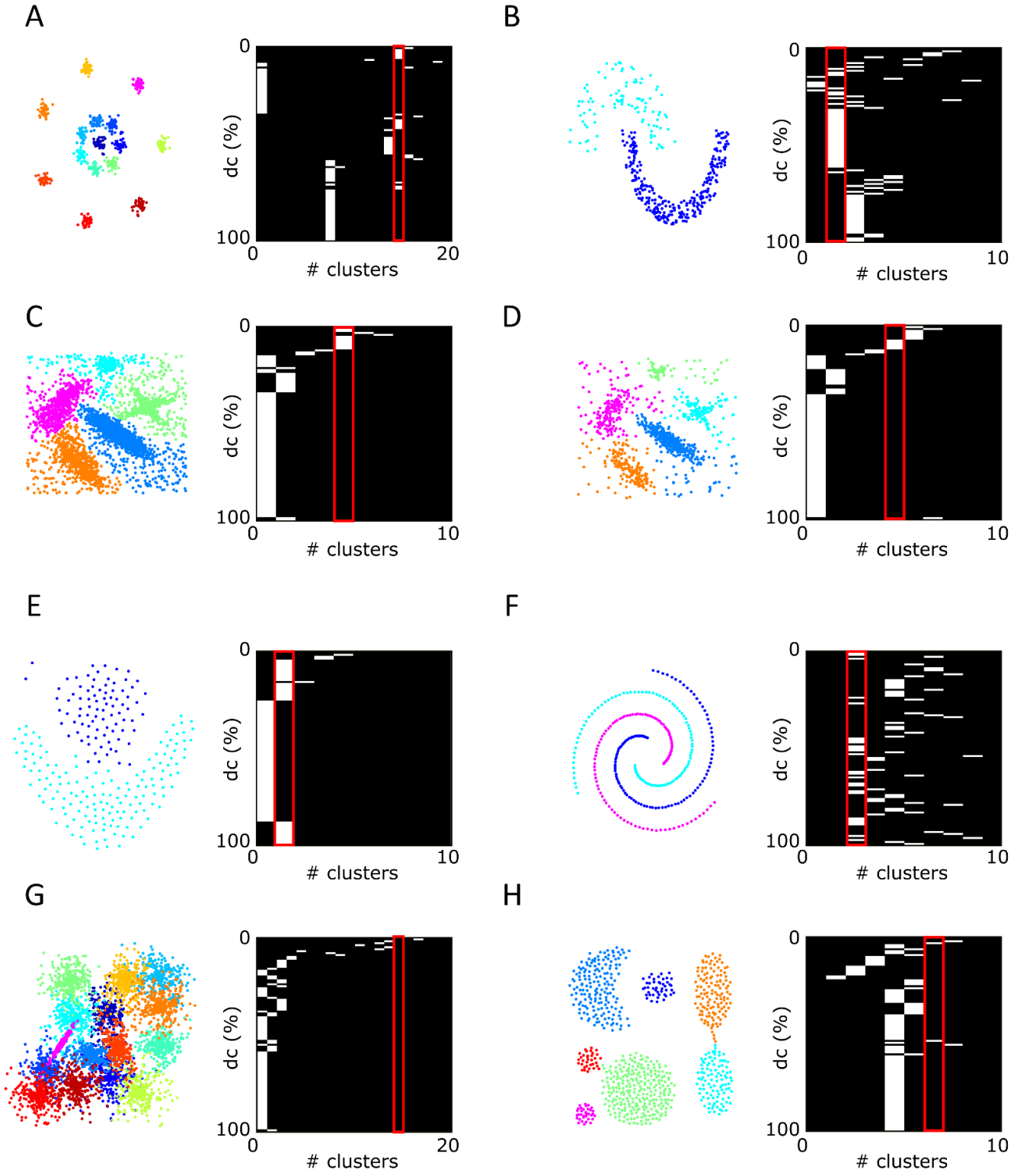

Fig. S3. Accuracy of the clusterdp algorithm is critically dependent on the choice of parameter d_c. Left panels represent synthetic point distributions used in (6): (*A*) (20), (*B*) (17), (*C-D*) (6), (*E*) (16), (*F*) (18), (*G*) (22), and (*H*) (15). White lines in right panels correspond to number of clusters found using automatic clusterdp for a particular d_c value expressed in percentile of all pairwise distances. The red outlines mark the column with the correct number of clusters.

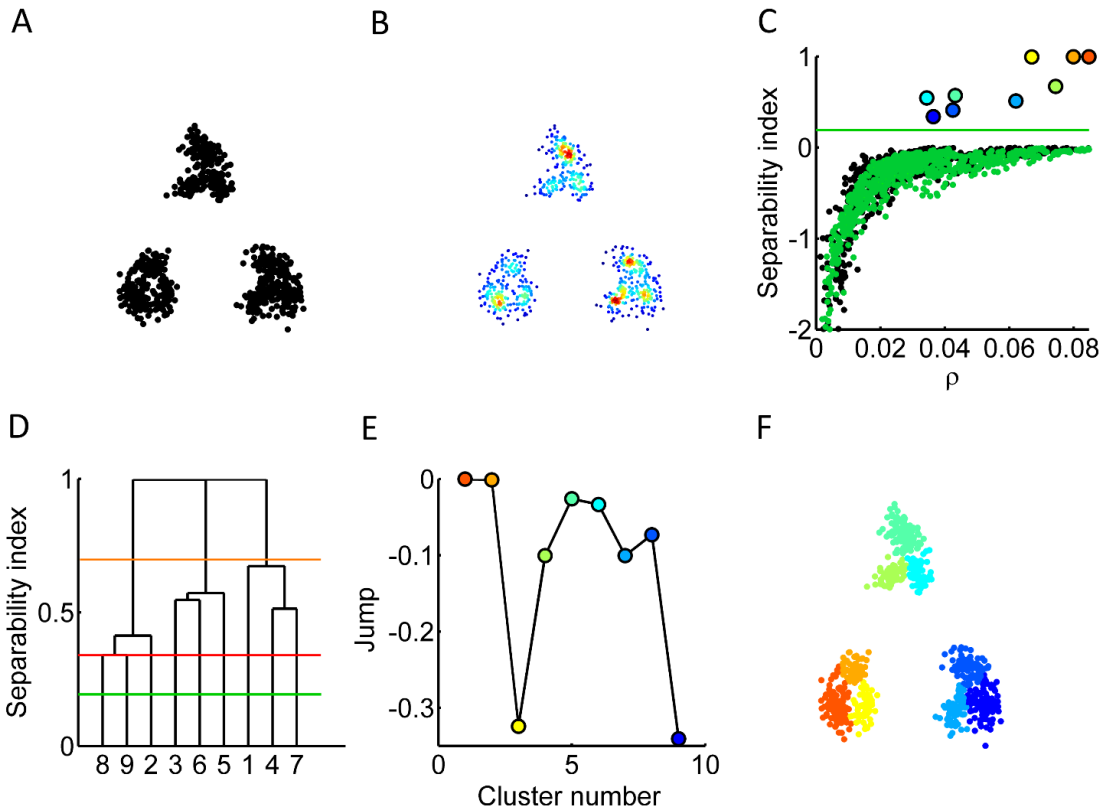

Fig. S4. Solution of clusterdv for data set with nested groups of clusters. (*A*) Point distribution drawn from mixture of nine gaussians. (*B*) Local densities are calculated using adaptive gaussian density estimator. (*C*) SI versus local density (ρ). Green points are from example reference distribution calculate using the onion method. (*D*) SI dendrogram. (*C*-*D*) Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. (*E*) SI jump versus cluster centers sorted by decreasing SI value. (*F*) Cluster assignment of the point distribution in (*A*) using the SI jump or onion threshold cut-offs.

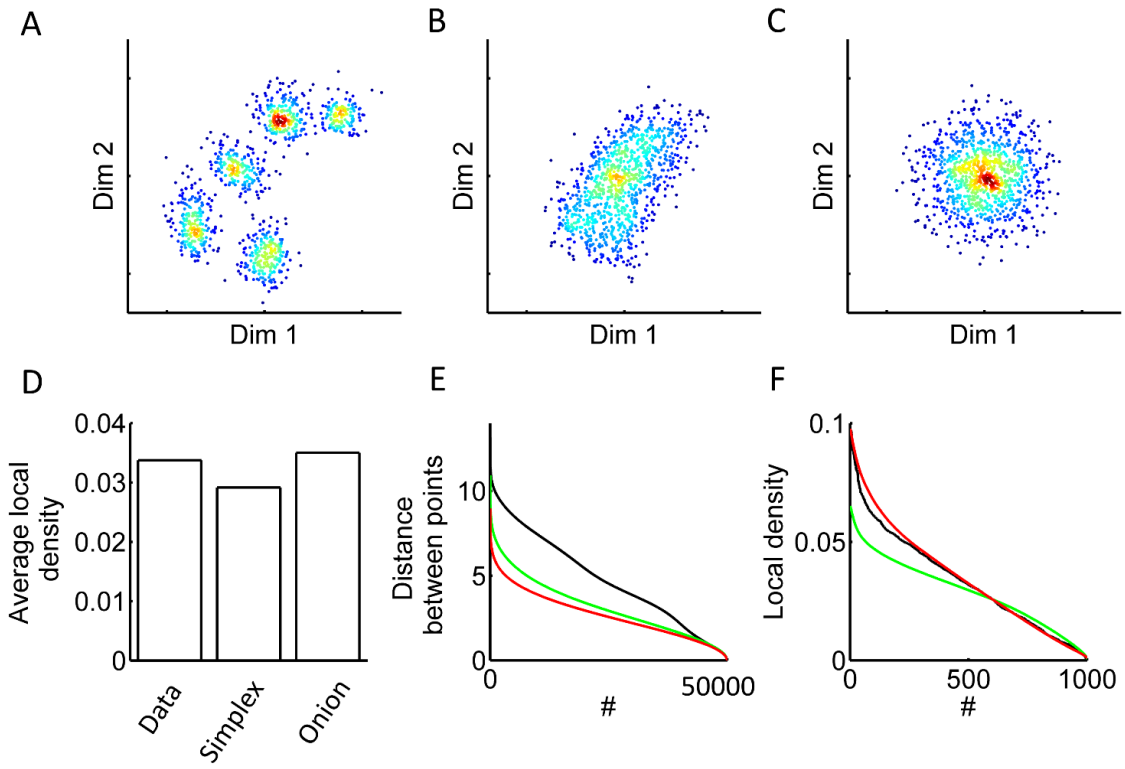

Fig. S5. Methods to calculate reference distributions. (*A*) Synthetic point distribution drawn from a mixture of five gaussians. (*B*) Reference distribution obtained by resampling the distribution in (*A*) using the simplex method. (*C*) Reference distribution obtained by resampling the distribution in (*A*) using the onion method. (*A*-*C*) Colours represent local densities calculated by using an adaptive mixture of gaussians density estimate. (*D*) Average local density between the three distributions in (*A*-*C*). (*E*) Distance between points plotted in descending order. (*F*) Local densities plotted in descending order. (*E*-*F*) Black, original distribution. Green, simplex distribution. Red, onion distribution.

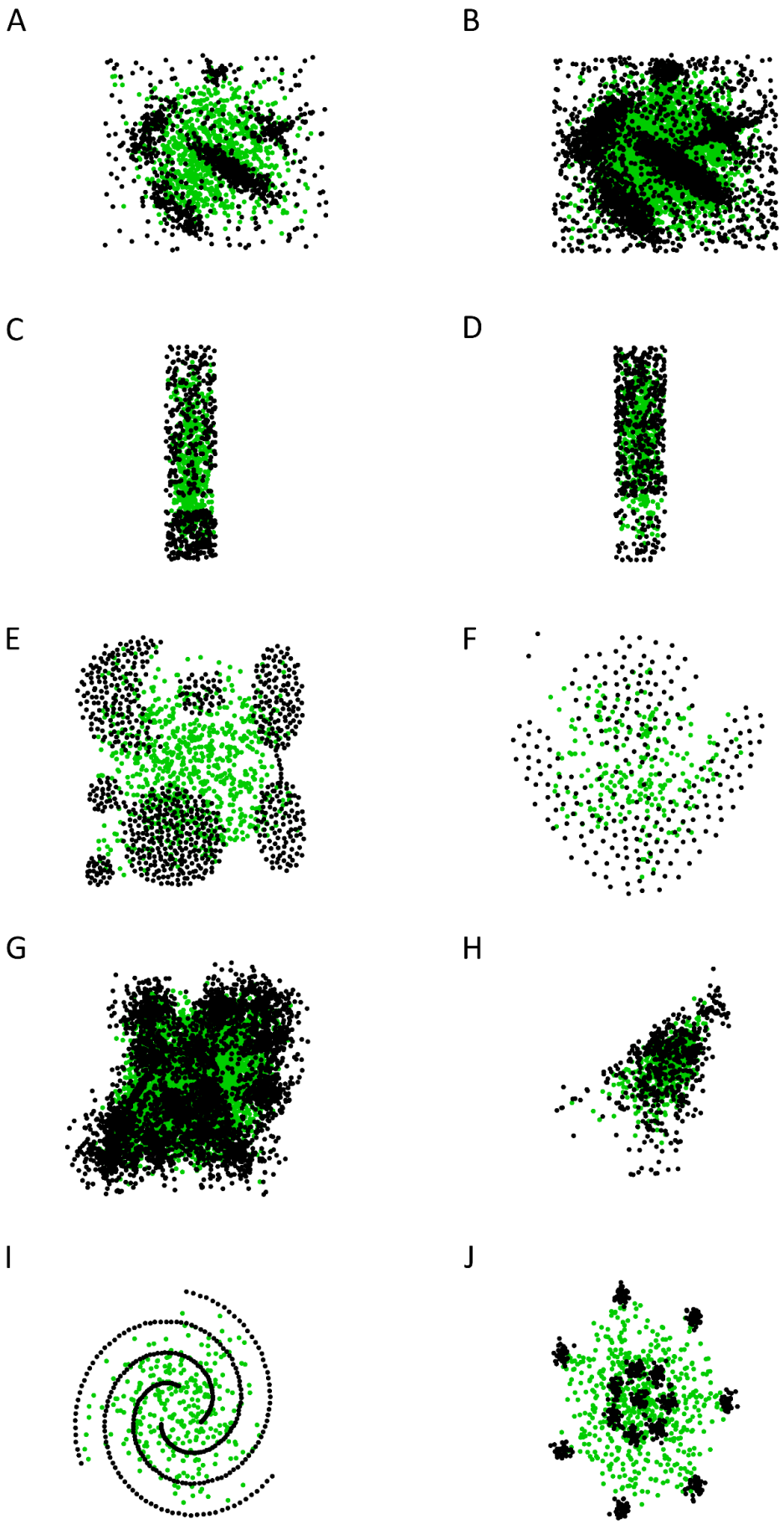

Fig. S6. Simplex method example reference distributions. Black points are original point distributions. Green points are "simplex" reference distributions. Synthetic point distributions from: (*A*-*B*) (6), (*C*-*D*, *H*) this study, (*E*) (15), (*F*) (16), (*G*) (22), (*I*) (18), and (*J*) (20).

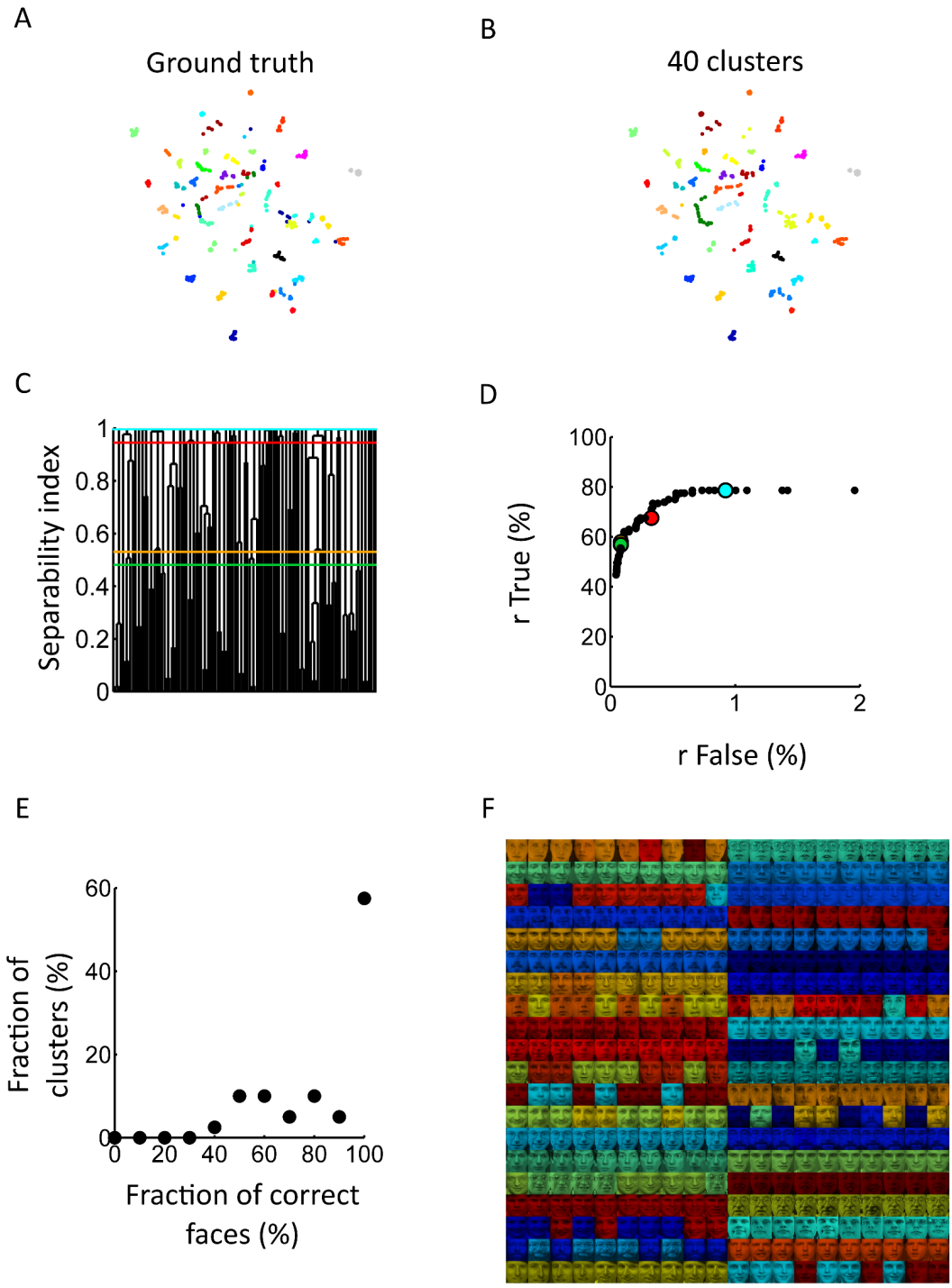

Fig. S7. Density valley clustering of the Olivetti face data set. Complex wavelet structural similarity (CW-SSIM) index (10) was applied to the Olivetti face data set (9) to calculate similarities between pairs of images. Since clusterdv is not a distance based clustering method we applied t-SNE to reduce the dimensionality of the similarity matrix (perplexity 30). (*A*-*B*) CW-SSIM embedded in a two-dimensional T-SNE space. Point colours correspond to ground truth (*A*) and clusterdv solution of 40 clusters (*B*). (*C*) SI dendrogram of data in (*A*-*B*). (*D*) Fraction of pair of images correctly associated to the same cluster (rTrue) as a function of the fraction of pair of images erroneously assigned to the same cluster (rFalse), using different cut-off values. (*C*-*D*) Red, SI jump. Orange, simplex threshold. Green, onion threshold. Cyan, manually picking 40 clusters with highest SI value. (*E*) Histogram of percent matching faces for the largest group in each cluster for the 40 cluster centres solution. (*F*) Assignment of the Olivetti data set for the 40 cluster centres solution. Each picture is labelled with the colour of the cluster it was assigned to.

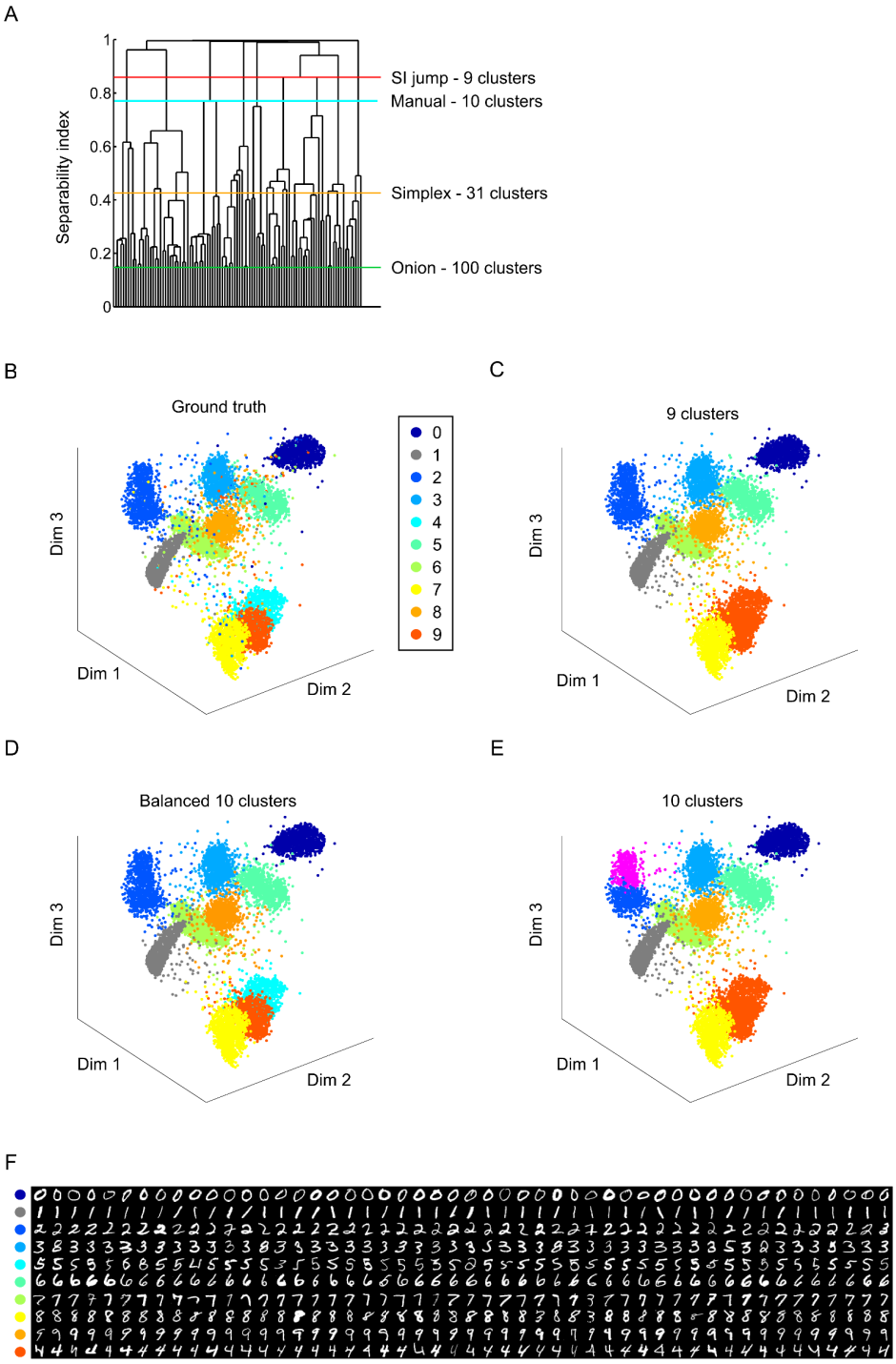

Fig. S8. Density valley clustering of the MNIST handwritten digit data set. Parametric t-SNE was used to embed the raw data of the test MNIST data set into a three-dimensional space (perplexity = 30) (32). (*A*) SI dendrogram of data in (*B*-*E*). Red line, SI jump. Orange line, simplex threshold. Green line, onion threshold. Cyan line manually picking 10 clusters with higher SI values. (*B*) Colors represent ground truth according to legend. (*C*) Solution given by SI jump. Most clusters are correct with the exception of clusters of digits 9 and 4 that were grouped together because these clusters are not well separated in the t-SNE space. Compare clusters 4 and 9 in (*B*) with dark orange cluster in (*C*). (*D*) Solution by finding clusters in dendrogram by minimizing standard deviation of number of points per cluster (see Materials and Methods for details). (*E*) Solution given by manually picking 10 clusters with a single cut-off. Similar to the SI jump solution but cluster 2 was divided into two clusters (compare cluster 2 in (*B*) with blue and magenta cluster in (*E*)). (*F*) Fifty digits were picked randomly from solution in (*D*).

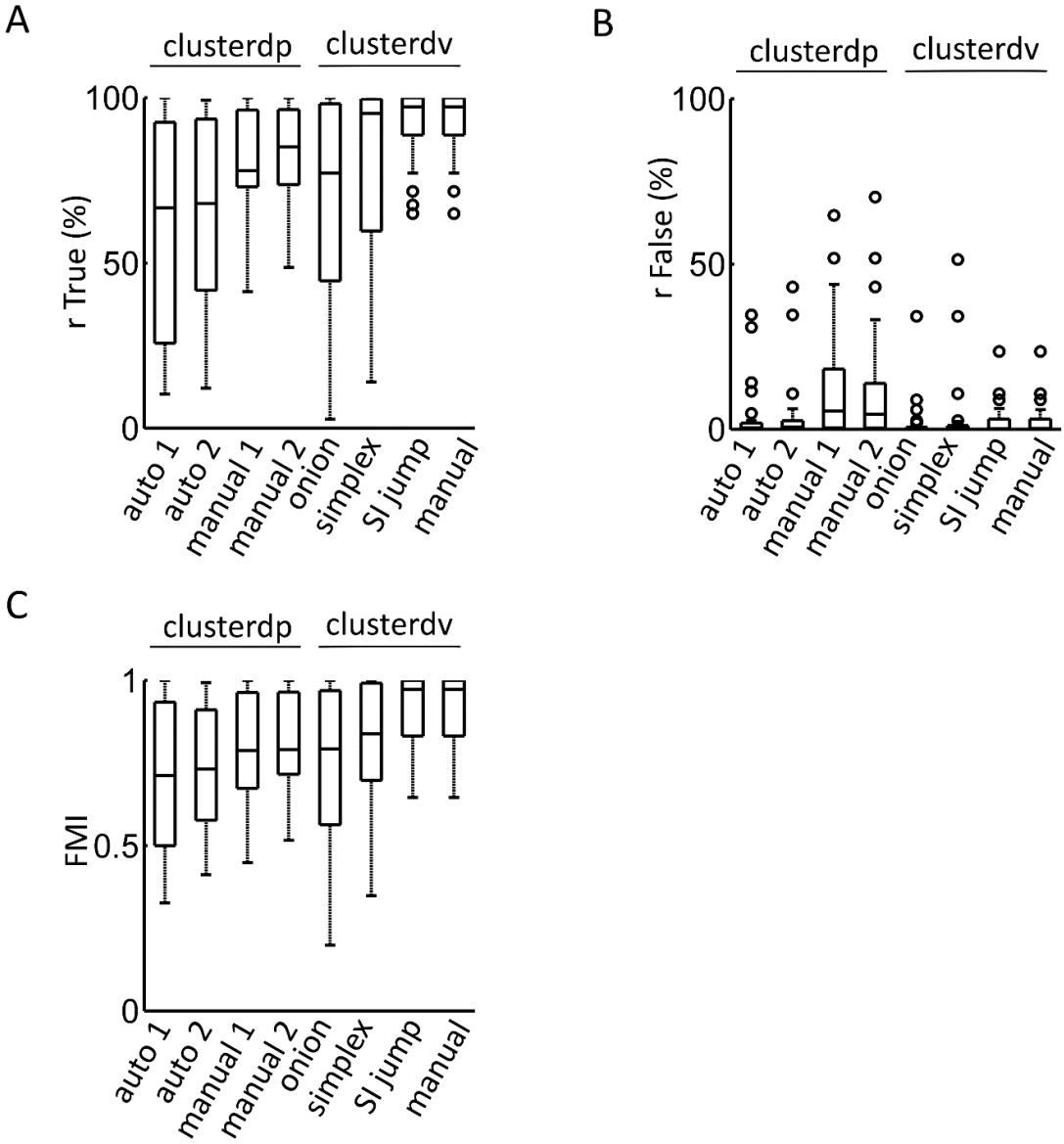

Fig. S9. Performance comparison between clusterdp and clusterdv. The clusterdp and clusterdv were applied to 33 data sets (for references of datasets see Tables S1-3). For clusterdp the d_c parameter of 1% and 2% were used (numbers in legends) and the number of clusters were chosen automatically (auto, see Fig. S1 for details) or set by hand to the known number of clusters by choosing the points with the highest γ values (manual). The number of clusters for clusterdv were chosen automatically by the largest SI jump, simplex or onion criteria or manually by choosing the number of clusters in ground truth with the highest SI value. 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.

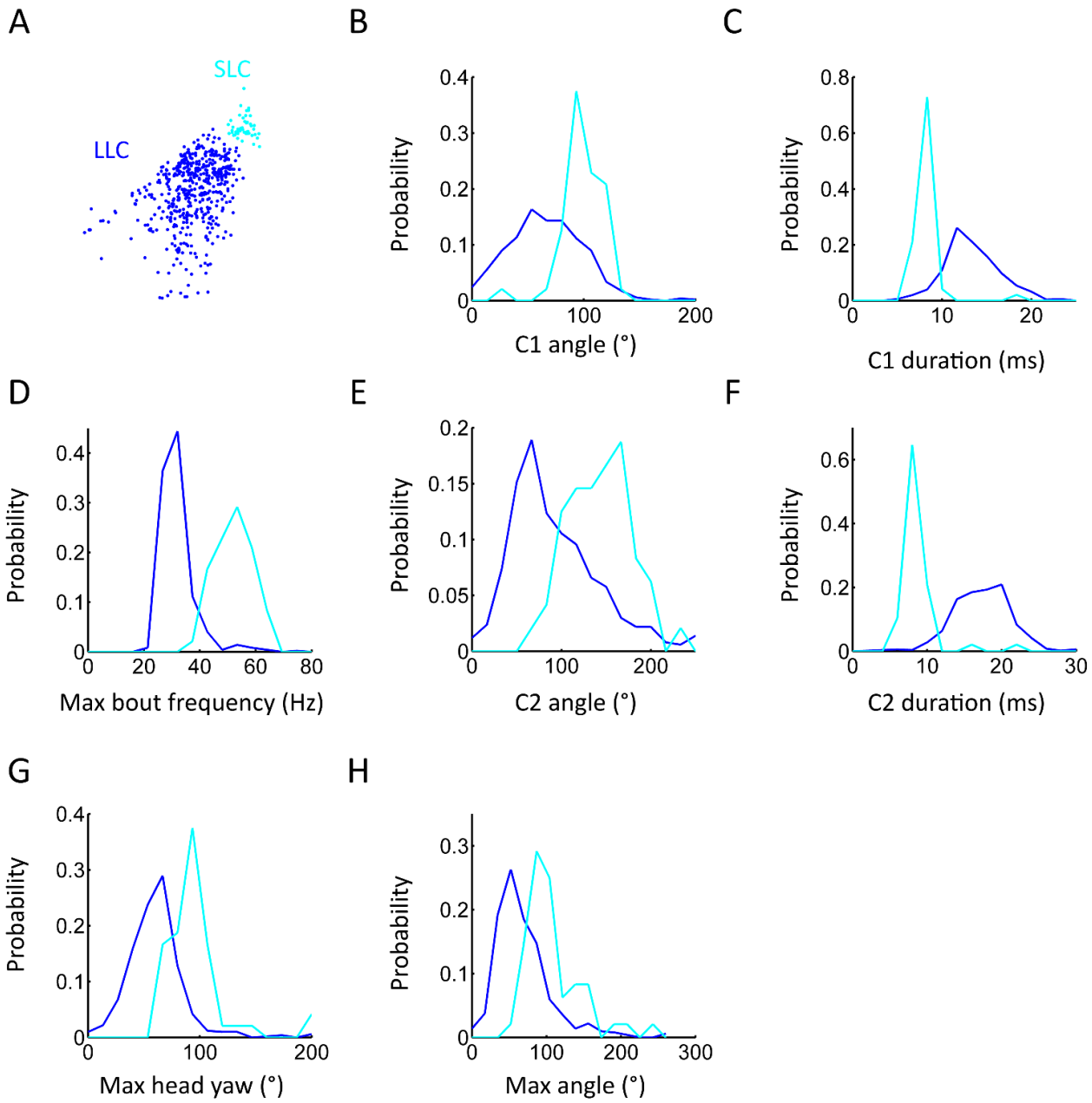

Fig. S10 | Short latency and long latency bout types have distinct distributions of kinematic parameters. (*A*) Swim bout data set obtained by larvae responding to acoustic stimuli. Colors represent bout categorization obtained by clusterdv (Fig. 4*M-P*). (*B-H*) Distributions were obtained by using all bouts in (*A*). SLC and LLC bout types show different kinematic parameter values that are in accordance with previous reports (13). Colours of distributions correspond to clusters in (*A*).

Table S1. Clusterdv performance

Clusterdv performance[1].

Data set name	# points[2]	t-SNE[3]	Onion	Simplex	SI	Manual[4]	r True (%)	r False (%)	FMI	Ref.
2 level 1	504 (2)	×	9	9	9	9	98.42	0.20	0.98	t.s.[6]
2 level 2	700 (2)	×	9	3	9	9	82.78	2.08	0.84	t.s.[6]
5 easy	1000 (2)	×	5	5	5	5	99.14	0.16	0.99	t.s.[6]
A1	3000 (2)	×	20	20	20	20	95.28	0.26	0.95	19
A2	5000 (2)	×	35	35	35	35	96.64	0.10	0.97	19
A3	7500 (2)	×	50	50	50	50	97.30	0.06	0.97	19
Aggregation	788 (2)	×	7	7	7	7	99.81	0.04	0.99	15
Beeps 20	550 (2)	×	4	2	1	2	99.57	2.27	0.99	t.s.[6]
Bone marrow	38 (3)	✓ (7)	3	2	3	3	77.19	8.86	0.82	21
Cassini	1250 (2)	×	7	9	3	3	100	0	1.00	4
Cuboid	250 (3)	×	10	8	4	4	100	0	1.00	4
Exclamation 1	500 (2)	×	6	2	2	2	100	0	1.00	t.s.[6]
Exclamation 2	500 (2)	×	4	2	2	2	100	0	1.00	t.s.[6]
Exclamation 3	186 (2)	×	3	3	3	3	100	0	1.00	t.s.[6]
Rodriguez 1	4000 (2)	×	8	5	5	5	n. d.	n. d.	n. d.	6
Rodriguez 2	1000 (2)	×	11	5	5	5	n. d.	n. d.	n. d.	6
Flame	240 (2)	×	2	2	2	2	100	0	1.00	16
Jain	373 (2)	×	9	5	2	2	100	0	1.00	17
MNIST test	10000 (3)	✓ (30)	100	31	9	10	90.67 (9)[5]	3.23 (9)[5]	0.83	11
Olivetti	400 (2)	✓ (30)	81	79	60	40	78.67	0.92	0.75	9,10
Path based	300 (2)	×	2	2	3	3	87.38	23.53	0.76	18
R15	600 (2)	×	15	15	15	15	98.68	0.10	0.99	20
S1	5000 (2)	×	22	15	15	15	98.88	0.08	0.99	22
S2	5000 (2)	×	16	15	15	15	92.24	0.58	0.92	22
S3	5000 (2)	×	19	15	15	15	71.71	2.20	0.71	22
S4	5000 (2)	×	15	15	15	15	65.00	2.62	0.65	22
Seeds	210 (7)	×	103	3	3	3	79.89	10.73	0.80	23
Spiral	312 (2)	×	3	3	3	3	100	0	1.00	18
Spirals	250 (2)	×	1	1	2	2	100	0	1.00	4
T4	8000(2)	×	68	38	6	6	95.63	3.87	0.89	25
T5	8000(2)	×	66	33	6	6	88.19	4.87	0.81	25
T7	8000(2)	×	87	54	8	9	97.03	3.41	0.91	25
Zachary	34 (4)	✓ (30)	2	1	2	2	94.12	5.88	0.94	24

[1] Number of clusters identified by cutting dendogram at given criteria. Green: correct clusters were found by analysis. Red: incorrect number of clusters were found by analysis. Yellow: correct number of clusters found but cluster centers do not correspond to correct clusters.
[2] In brackets is the number of dimensions of the data set.
[3] In brackets is perplexity used in t-SNE analysis.
[4] Dendrogram was cut on the number of clusters that exist in ground truth.
[5] In brackets is number of clusters used to calculate r True and r False.
[6] This study.

Table S2. Clusterdp performance at d_c 1%.

Clusterdp performance (1% d_c)[1].

Data set name	# points[2]	t-SNE[3]	Automatic	Manual[4]	r True (%)	r False (%)	FMI	Ref.
2 level 1	504 (2)	×	11	9	97.27	0.35	0.97	t.s.[5]
2 level 2	700 (2)	×	9	9	89.07	1.18	0.90	t.s.[5]
5 easy	1000 (2)	×	5	5	99.14	0.16	0.99	t.s.[5]
A1	3000 (2)	×	20	20	97.59	0.13	0.98	19
A2	5000 (2)	×	35	35	98.23	0.04	0.98	19
A3	7500 (2)	×	50	50	98.06	0.04	0.98	19
Aggregation	788 (2)	×	17	7	79.89	5.49	0.80	15
Beeps 20	550 (2)	×	3	2	57.61	30.88	0.72	t.s.[5]
Bone marrow	38 (3)	✓ (7)	7	3	77.19	8.86	0.82	21
Cassini	1250 (2)	×	20	3	77.90	13.60	0.77	4
Cuboid	250 (3)	×	10	4	73.67	37.24	0.55	4
Exclamation 1	500 (2)	×	10	2	55.23	34.92	0.69	t.s.[5]
Exclamation 2	500 (2)	×	12	2	50.87	51.76	0.63	t.s.[5]
Exclamation3	186 (2)	×	5	3	61.16	19.55	0.65	t.s.[5]
Rodriguez 1	4000 (2)	×	5	5	n. d.	n. d.	n. d.	6
Rodriguez 2	1000 (2)	×	6	5	n. d.	n. d.	n. d.	6
Flame	240 (2)	×	12	2	62.51	43.79	0.63	16
Jain	373 (2)	×	19	2	79.72	13.04	0.85	17
MNIST test	10000 (3)	✓ (30)	15	10	85.83	3.25	0.80	11
Olivetti	400 (2)	✓ (30)	38	40	77.56	0.77	0.76	9,10
Path based	300 (2)	×	2	3	73.35	24.04	0.67	18
R15	600 (2)	×	16	15	99.33	0.05	0.99	20
S1	5000 (2)	×	15	15	98.88	0.08	0.99	22
S2	5000 (2)	×	15	15	93.69	0.46	0.94	22
S3	5000 (2)	×	15	15	74.20	1.95	0.74	22
S4	5000 (2)	×	15	15	66.66	2.58	0.66	22
Seeds	210 (7)	×	3	3	74.91	14.11	0.74	23
Spiral	312 (2)	×	3	3	79.70	11.50	0.79	18
Spirals	250 (2)	×	2	2	100	0	1.00	4
T4	8000(2)	×	19	6	73.04	6.29	0.71	25
T5	8000(2)	×	33	6	87.96	4.84	0.81	25
T7	8000(2)	×	32	9	41.32	8.52	0.45	25
Zachary	34 (4)	✓ (30)	13	2	75.73	64.70	0.65	24

[1] Number of clusters identified by cutting dendogram at given criteria. Green: correct clusters were found by analysis. Red: incorrect number of clusters were found by analysis. Yellow: correct number of clusters found but cluster centers do not correspond to correct clusters.

[2] In brackets is the number of dimensions of the data set.

[3] In brackets is perplexity used in t-SNE analysis.

[4] Dendrogram was cut on the number of clusters that exist in ground truth.

[5] This study.

Table S3. Clusterdp performance at d_c 2%.

Clusterdp performance (2% d_c)[1].

Data set name	# points[2]	t-SNE[3]	Automatic	Manual[4]	r True (%)	r False (%)	FMI	Ref.
2 level 1	504 (2)	×	9	9	98.42	0.20	0.98	t.s.[5]
2 level 2	700 (2)	×	9	9	88.44	1.22	0.90	t.s.[5]
5 easy	1000 (2)	×	5	5	99.14	0.16	0.99	t.s.[5]
A1	3000 (2)	×	20	20	97.38	0.14	0.97	19
A2	5000 (2)	×	35	35	97.20	0.08	0.97	19
A3	7500 (2)	×	25	50	89.91	1.07	0.75	19
Aggregation	788 (2)	×	11	7	99.81	0.04	1.00	15
Beeps 20	550 (2)	×	2	2	51.31	43.03	0.67	t.s.[5]
Bone marrow	38 (3)	✓ (7)	4	3	80.99	33.18	0.71	21
Cassini	1250 (2)	×	6	3	77.81	11.75	0.78	4
Cuboid	250 (3)	×	16	4	85.17	11.42	0.79	4
Exclamation 1	500 (2)	×	8	2	58.87	70.29	0.67	t.s.[5]
Exclamation 2	500 (2)	×	8	2	50.87	51.76	0.63	t.s.[5]
Exclamation3	186 (2)	×	5	3	62.94	14.58	0.69	t.s.[5]
Rodriguez 1	4000 (2)	×	5	5	n.d.	n.d.	n.d.	6
Rodriguez 2	1000 (2)	×	7	5	n.d.	n.d.	n.d.	6
Flame	240 (2)	×	5	2	74.81	20.92	0.78	16
Jain	373 (2)	×	8	2	72.66	18.84	0.79	17
MNIST test	10000 (3)	✓ (30)	12	10	85.70	3.27	0.80	11
Olivetti	400 (2)	✓ (30)	34	40	78.83	0.99	0.74	9,10
Path based	300 (2)	×	2	3	73.35	24.04	0.67	18
R15	600 (2)	×	15	15	99.33	0.05	0.99	20
S1	5000 (2)	×	15	15	99.05	0.07	0.99	22
S2	5000 (2)	×	15	15	94.00	0.43	0.94	22
S3	5000 (2)	×	17	15	75.00	1.89	0.75	22
S4	5000 (2)	×	17	15	66.28	2.61	0.65	22
Seeds	210 (7)	×	3	3	81.67	10.78	0.81	23
Spiral	312 (2)	×	3	3	92.07	4.57	0.92	18
Spirals	250 (2)	×	4	2	100.00	0.00	1.00	4
T4	8000(2)	×	16	6	77.69	7.00	0.73	25
T5	8000(2)	×	22	6	87.97	4.84	0.81	25
T7	8000(2)	×	22	9	48.69	7.89	0.52	25
Zachary	34 (4)	✓ (30)	16	2	94.12	5.88	0.94	24

[1] Number of clusters identified by cutting dendogram at given criteria. Green: correct clusters were found by analysis. Red: incorrect number of clusters were found by analysis. Yellow: correct number of clusters found but cluster centers do not correspond to correct clusters.

[2] In brackets is the number of dimensions of the data set.

[3] In brackets is perplexity used in t-SNE analysis.

[4] Dendrogram was cut on the number of clusters that exist in ground truth.

[5] This study.