A trainable clustering algorithm based on shortest paths from density peaks

Diego Ulisse Pizzagalli1,2, Santiago Fernandez Gonzalez2*†, Rolf Krause1*†

Clustering is a technique to analyze empirical data, with a major application for biomedical research. Essentially, clustering finds groups of related points in a dataset. However, results depend on both metrics for point-to-point similarity and rules for point-to-group association. Non-appropriate metrics and rules can lead to artifacts, especially in case of multiple groups with heterogeneous structure. In this work, we propose a clustering algorithm that evaluates the properties of paths between points (rather than point-to-point similarity) and solves a global optimization problem, finding solutions not obtainable by methods relying on local choices. Moreover, our algorithm is trainable. Hence, it can be adapted and adopted for specific datasets and applications by providing examples of valid and invalid paths to train a path classifier. We demonstrate its applicability to identify heterogeneous groups in challenging synthetic datasets, segment highly nonconvex immune cells in confocal microscopy images, and classify arrhythmia heartbeats in electrocardiographic signals.

INTRODUCTION

Clustering algorithms aim at automatically grouping points. This task is critical for extracting knowledge from complex datasets of unlabeled data, which are commonly produced by empirical studies (1) and are impossible to analyze manually (2). The identified groups can indicate the presence of multiple phenomena (i.e., populations) in the dataset. Moreover, the structure of each group (hereafter, shape) can indicate the relationship between the included points. Because clustering is a problem without a unique solution, results depend on different metrics for point-to-point similarity and rules for point-to-cluster association. However, metrics and rules generally restrict the set of admissible solutions. As a consequence, when metrics and rules are not chosen appropriately, clustering artifacts are introduced. This is especially relevant when analyzing datasets that have points from populations with arbitrary and heterogeneous structure.

For example, common implementations of the broadly used K-means algorithm (3) use the Euclidean distance as a metric for point-to-point similarity and point-to-cluster association. Despite the efficiency of K-means in clustering linearly separable groups (i.e., globular-like), artifacts can be introduced when attempting to separate nonconvex and nested shapes.

Methods based on density overcome these limitations. Arbitrary shapes are reconstructed according to a density-connectivity criterion. For example, DBSCAN (density-based spatial clustering of applications with noise) (4) considers two points belonging to the same cluster if a sufficient number of points in a neighborhood are common (density reachable). As a result, the association rule of DBSCAN correctly identifies clusters with any shape having sufficient density. However, when two or more clusters are present and in close proximity, a wide density reachability threshold may join them, while an excessively strict threshold may fail in detecting the clusters. Generally, algorithms like DBSCAN depend on a density-connectivity threshold, which can be difficult to adapt for specific dataset and applications.

Recently, Rodriguez and Laio (5) proposed a remarkable strategy that achieves clustering by finding density peaks (CDP). CDP addresses the limitations of DBSCAN by initially finding density peaks and using them to separate clusters. Density peaks are considered as points surrounded by enough other points with lower density. Hence, the distance between two density peaks should be higher than the distance between another arbitrary point and its closer point with a higher density. On the basis of this intuition, CDP identifies density peaks by detecting the outliers of a density-distance plot. This task can be performed either automatically or manually by inspecting the density-distance plot. Because of the simplicity of this rule, CDP can be applied whenever density can be measured, which holds in a wide range of applications. Once peaks are identified, CDP assigns them a unique label representing a cluster.

However, the association rule of CDP for the remaining points consists of inheriting the label of the closest point with a higher density, which ends in being a density peak. Although this generic rule is applicable to any cluster shape, cases remain where such a local criterion is not optimal (i.e., it associates a point disregarding how the other points are associated).

In this work, we propose to substitute the local association rule of CDP, with the solution of a global optimization problem on a graph, bringing two main contributions. First, a global vision on the dataset allows finding globally optimal solutions not identifiable with a local choice. Second, we evaluate path properties rather than point-to-point differences. This allows defining path-cost functions that can be generically meaningful (i.e., large gap penalization). In addition, by providing examples of valid and invalid paths, a path classifier can be trained. This is used for easily adopting and adapting the proposed method to specific tasks and datasets, by providing examples rather than tuning parameters, within a semisupervised approach for data analysis.

We evaluate the proposed method (both with a generic path-cost function and with a trainable path classifier) on synthetic datasets designed to challenge clustering algorithms (2). Then, we provide two applications for analyzing real biomedical data. The first is the segmentation of highly nonconvex immune cells in confocal microscopy (grouping pixels in space). The second is the identification of different types of arrhythmia in electrocardiographic (ECG) signals. Our method obtained clustering results not obtainable by DBSCAN nor CDP.

In conclusion, our method exploits network information (6) and the integration of possibly existing knowledge in line with a theory-guided
data science paradigm (7). We consider it as a step toward machines (fast, operation based) capable of working collaboratively with humans (slow, practice based) (1).

RESULTS
Comparison with CDP on a simplified dataset
In CDP, a point is associated with the closest point with a higher density. This local rule takes place regardless of how the other points are associated, and there exist cases where artifacts are introduced. An example is illustrated in Fig. 1A, where a long-thin projection of a nonglobular cluster and a second globular-like cluster are in close proximity. The estimated point density (Fig. 1B) and the distance to the closest point with a higher density (Fig. 1C) are used to find density peaks in a density-distance plot (Fig. 1D). For a generic point \( x \), the association rule of CDP connects it to the wrong density peak \( q \) (Fig. 1D, white arrows). By contrast, our algorithm computes the path optimally connecting \( x \) to a density peak, which turns out to be \( p \) (Fig. 1F, white arrows). Although local heuristics would not consider that path (i.e., longer, nonsmooth density variation), our algorithm identifies it by solving a shortest path problem in which paths having multiple small gaps rather than fewer but larger gaps are preferred. This has been expressed using a minimax path-cost function (Fig. 1G). As a result, the thin portion of the cluster is correctly associated by our algorithm (Fig. 1, H and I). We consider the differences in this example relevant. Analyzing the cuts introduced by the two algorithms (Fig. 1, dashed
Algorithm for global optimization

We consider clustering as the problem of associating each point to a density peak with the optimal set of paths. This problem is modeled as a single-source shortest path (SSSP) on a graph (Fig. 1F and fig. S1). A globally optimal solution is computed by Dijkstra’s SSSP algorithm (8), adopting a customized path-cost function.

To explain this in more detail, let us define the following quantities.

Let \( G(V, E, w) \) be a weighted graph where a node \( v \in V \) is connected to another node \( u \in V, v \neq u \) through an edge \( e_{vu} \in E \) with an associated cost \( w_{vu} : E \rightarrow \mathbb{R}_+ \). \( w_{vu} \) is a distance metric (i.e., Euclidean distance).

Let \( \rho \) be the density of a given node \( v \in V \), defined as the number of points in a weighted neighborhood. See (5) for an implementation where a Gaussian kernel is used to estimate \( \rho \).

Let \( \delta_i = \min ||v - u|| \forall u \in V \) such that \( \rho_u > \rho_v \), which is the distance of \( v \) to the closest point \( u \) with a higher density.

Let \( P_i = \{v \in V | \rho_v > \rho_i \cap \delta_i > \delta_{i+1}, 0 < \delta_i < \min(\delta) \} \) be a path from \( s \) to \( x_n \) with associated cost \( c_i \). The cost of the extended path \( \Gamma_i = \{s, x_1, x_2, ..., x_{w+1}, x_n\} \) be a path from \( s \) to \( x_n \) with associated cost \( c_i \). The cost of the extended path \( \Gamma_i = \{s, x_1, x_2, ..., x_{w+1}, x_n\} \) can be evaluated by means of a function that considers the cost until \( x_1 \) (corresponding to \( c_1 \)), and the path fragment \( x_1, ..., x_w \) where \( w + 1 \) is the number of nodes in the fragment. In the original formulation of the Dijkstra algorithm, this is computed by adding to \( c_1 \) the cost of the edge from \( x_{w+1} \) to \( x_1 \).

However, our method outperforms the other algorithms when \( x_1 \) is fully connected, the usage of a path-cost function that sums the cost of edges in the path does not penalize large gaps nor the problem of associating \( x \) to the closest but possibly not optimal peak. Therefore, we describe the position of \( x \) with respect to a density peak by evaluating the path connecting them. Properties of the path, such as mediating elements (9), textures (10), density patterns, or user-defined metrics, can be used to associate a meaningful cost.

Dijkstra’s SSSP algorithm can be adopted to compute the shortest paths on graphs if the path cost is not decreasing when extending a path. Being \( \xi : G \rightarrow \mathbb{R}, \xi(\Gamma) = c \) the cost of a path \( \Gamma \in G \), the following should be guaranteed

\[
\xi(\Gamma = \{s, x_1\}) \leq \xi(\Gamma' = \{s, x_1, x_{i+1}\}) \tag{1}
\]

Several functions respect Eq. (1). Among these, cumulative functions sum the weights of edges included in the path \( \xi_p(\Gamma) = \sum_{i,j \in \Gamma} (|w_{ij}|)^p \).

Values of \( p \geq 1 \) penalize edges with high cost, preferring (if possible) alternative multiple edges with lower cost. Generally, \( p > 1 \) acts as a regularizer, preferring edges with similar value.

When \( p \rightarrow +\infty \), the total cost of a path is given by the edge with maximum cost in the path. This metric is used in the minimax formulation, which defines for a generic path \( \Gamma \) the cost \( \xi_{min}(\Gamma) \) as max \( \{w_{ij}, i, j \in \Gamma \} \). \( \Gamma \) is “minimax” if its cost is minimum with respect to all the existing paths from the same source to the same destination

\[
\Gamma \ "\text{minimax}" \text{ if } \xi_{min}(\Gamma) \leq \xi_{min}(\Lambda) \quad \forall \Lambda \neq \Gamma
\]

Trainable path-cost function

The Dijkstra algorithm incrementally extends a set of shortest paths by evaluating the cost of the possible extensions. We exploited this behavior and used a custom function for computing the cost of an extended path, based on the last path fragment. Let \( \Gamma_1 = \{s, x_1, x_2, ..., x_{w+1}, x_n\} \) be a path from \( s \) to \( x_n \) with associated cost \( c_1 \). The cost of the extended path \( \Gamma_2 = \{s, x_1, x_2, ..., x_{w+1}, x_n, x_{w+2}\} \) can be evaluated by means of a function that considers the cost until \( x_1 \) (corresponding to \( c_1 \)), and the path fragment \( x_1, ..., x_w \) where \( w + 1 \) is the number of nodes in the fragment.

Among the possible features to describe such a path fragment, we propose the density profile as the ordered density values of the nodes in the fragment. This opens the possibility to integrate existing knowledge in a supervised learning approach to prefer certain paths with respect to others by training a path classifier providing examples of admissible and nonadmissible paths.

Reconstruction of heterogeneous synthetic shapes

To benchmark the proposed algorithm, we benefit from the availability of the ClustEval platform (2), which provides synthetic datasets specifically designed to challenge clustering algorithms, together with an evaluation methodology. Among these datasets, we report the comparison with respect to other common methods on an example categorized as difficult, which visually contains heterogeneous structures (i.e., two globular-like clusters, surrounded by one thin-elongated cluster) (Fig. 2A). Our method achieves comparable performances to state-of-the-art algorithms when the generic minimax path-cost function is used (Fig. 2, A and B, orange columns).

However, our method outperforms the other algorithms when trained with examples of valid and invalid paths (Fig. 2, A and B, blue columns). In this example, 25 fragments of good paths and 25 fragments of bad paths composed by five points, respectively (Fig. 2C), were automatically generated from the ground truth and were described using the density of points as feature vector (Fig. 2D). In comparison to CDP, our method allowed correct clustering of the outermost elongated structure while still keeping separate the two central globular clusters. We provide two scripts to train the algorithm. In demo_trainable.m (data file S2), we provide the code for training the algorithm from an available (but not necessarily complete) ground truth. In demo_interactive.m instead, a GUI (graphical user interface) allows a user to visually inspect the dataset and annotate valid/invalid paths. A video showing an example of interactive training is provided in movie S1. Quantitative results on the remaining datasets included in ClustEval are provided...
Robustness to density peak detection

Results of the proposed algorithm depend on the initial identification of density peaks. Figure 3 includes a performance assay, evaluating the F1 score of the proposed method with respect to an artificial perturbation on the position of density peaks (Fig. 3A). In the example 04_fu_flame from ClustEval (2), the proposed method tolerates perturbations up to 30% of the dataset size without decrease in performance (Fig. 3B). Above this level, the algorithm exhibited a variable and generally decreasing behavior that was dependent on the position of the peaks with respect to the conformation of the dataset. More precisely, when two seeds were randomly assigned to points in the same cluster, an undesired partition was introduced, lowering the performances. Perturbations keeping seeds in two separated clusters are better tolerated (Fig. 3C).

The identification of density peaks depends on the setting of the threshold parameters \( t_p \) and \( t_b \). These two parameters can be set to select only the outliers of a density-distance plot [see (5)] (Fig. 3D). Variations from the optimal values that do not select additional points as density peaks keep the maximum achievable performance. Variations that include additional points result in the identification of an increased number of clusters, lowering the performance with respect to the ground truth labels (Fig. 3E).

Noise handling

The proposed method can identify points that are suitable to be considered as noise by analyzing the path connecting them to a cluster. This can be achieved, for instance, by introducing a cutoff threshold on the path cost. When using a trainable path-cost function, examples of paths including noisy points can be used to penalize these connections. Automatic thresholding methods such as Otsu thresholding (11) can be used to set the cutoff distance. Considering that a density profile can be used as path descriptor, this strategy is in line with the criterion of DBSCAN, which considers as noise those points not density reachable from a cluster.

A benchmark of the proposed method in the presence of background noise is reported in Fig. 4. To this end, we generated a synthetic probability distribution with three density peaks and a uniform background with a different probability (Fig. 4A). The proposed method exhibited average better results than CDP, with a decreased slope of the F1 score with respect to the background noise level (Fig. 4B). With a low level of background noise, both a minimax path-cost function and a trained path-cost function achieved similar performances (Fig. 4, C to E).
Obtained with algorithm. The position of seeds is perturbed by a displacement of max ±1 (black line). (A) Color-coded ground truth including two clusters in close proximity (red and blue) from the synthetic dataset 04_fu_flame provided in ClustEval. Points colored in black, at the intersection of the arrows, are the identified density peaks, which are used as seeds by the proposed algorithm. The position of seeds is perturbed by a displacement of max ±1 (black line). (B) Performance degradation with respect to the maximum perturbation. Results are obtained with n = 5 replicas with a random perturbation direction. CDP* refers to a modified version of CDP, forced to associate points to the perturbed seeds. (C) Color-coded results of the proposed method using the seeds colored in black. (D and E) Robustness to the setting of τ_r, and τ_p parameters. The optimal settings of the parameters (D) identifies two outliers of the δ – ρ graph. (E) Variations from the optimal settings include additional subclusters, lowering the performances of the algorithm.

For higher levels of noise, a path-cost function trained to associate high cost to paths including noise (Fig. 4, F to H) yielded an improved recognition of background.

Application on biomedical data

Segmentation of immune cells in confocal microscopy

Microscopy images are considered challenging spatiotemporal biomedical datasets (12, 13). This is particularly relevant for images of immune cells, which exhibit high plasticity, both globular and nonglobular shapes, possibly in contact (14, 15). Considering segmentation as a grouping task, the aforementioned properties represent challenges for clustering algorithms (i.e., heterogeneous shapes in close proximity). Moreover, they remain challenging for state-of-the-art bioimaging software, which requires manual correction of results, lowers usability (16), and compromises reproducibility.

Here, we apply our clustering algorithm for grouping pixels. We assume that other image-processing tasks such as background removal have been already executed with specific software such as Ilastik (17), Trainable Weka Segmentation (18), or Imaris (Bitplane). Qualitative results are presented in Fig. 5. Similar to the example in Fig. 1, in Fig. 5A, the local association rule of CDP yields to the wrong association of a thin dendrite (i) but correctly separates touching cells (ii). In Fig. 5B, DBSCAN correctly reconstructs the shape of dendritic cells but does not separate touching cells. In Fig. 5C, our algorithm, penalizing large gaps in the paths, connects the pixels to the correct density peaks in both the cases.

In this example, training the algorithm does not increase performances substantially. These are limited because more than necessary density peaks are identified as starting points. As a consequence, some cells are split into multiple parts. To merge these subclusters, a different metric can be used for computing the δ measure for each peak. Alternatively, methods for merging multiple clusters such as (19) can be applied for postprocessing.

Identification of different heart rhythms

Long-term ECG records are widely used to capture the dynamic behavior of the heart, which may change over time and exhibit different rhythms. Here, we applied clustering analysis to automatically identify different rhythms in a complex ECG record showing various types of arrhythmia (20). Arrhythmia affects the distance between heartbeats (Fig. 6A). To this end, we described each heartbeat with two parameters corresponding to the distance (in seconds) from the previous (RR1) and the next (RR2) heartbeat. These two parameters are similar in sinus rhythms (Fig. 6B) and vary linearly according to the heart rate. However, in the presence of extrasystoles, these may change. Extrasystoles are beats that stay closer to the previous beat and are possibly followed by a long pause. There are two main types of extrasystoles: atrial and ventricular. The second has an increased alteration of the distances. Life-threatening arrhythmias such as ventricular tachycardia and flutter result in extremely short distances between beats (Fig. 6, C and D), compromising the pump function of the heart. In results presented in Fig. 6E, our algorithm identified six different rhythms.

Similar to the ground truth (manually annotated by a group of cardiologists), our algorithm identified groups corresponding to ventricular flutter (brown), isolated ventricular extrasystoles (red), isolated atrial extrasystoles (orange), and right bundle branch block (blue). By contrast, the algorithm identified two different subtypes (instead of one) of sinus beats, one associated with lower heart rate (gold), and one associated with higher heart rate (green).
Fig. 4. Robustness to background noise. (A) Synthetic probability distribution from which point distributions are drawn. Lighter areas have the maximum probability. The area in blue is a background with uniform probability from 0 to 0.1, corresponding to a number of points from 10 to 52% of the total. (B) Quantitative performance degradation assay with respect to the value of background probability. A total of 1000 points are drawn from the probability distribution in (A), varying the background uniform probability from 0 to 0.1. For each level of background probability, $n = 5$ replicates are generated. Bold lines refer to the mean of the F1 score among the replicates, while the shaded lines correspond to the range. Three peaks were selected using F1 ranking (5). (C to E) Qualitative results with a background probability of 0.01, corresponding to 900 points from the clusters and 100 points in the background. Cluster labels are color coded. Red x correspond to noise. (F to H) Qualitative results with a background probability of 0.1, corresponding to 480 points from the true clusters and 520 points in the background. Arrows indicate an aggregation of points in the background. The trained algorithm detects those points as noise.

With respect to CDP, our algorithm removed an unrealistic boundary that was cutting a piece of the ventricular extrasystole cluster (red). In CDP, these beats were joined by mistake to the normal beats (green), while in our algorithm, they are correctly joined with the ventricular extrasystole group (red). The trained version further allowed a group of mixed atrial-ventricular extrasystoles to be found, which were associated by mistake to the normal beats by the untrained version.

DISCUSSION

The proposed method inherits the advantages of density-based clustering methods but, using a global association rule, is able to overcome the limitations of local connectivity rules. This is in line with the work by Tenenbaum et al. (21), which uses global optimization (i.e., shortest path computation) to represent the global structure of a dataset in a lower dimensional coordinate system. Clustering can be interpreted as a special case of dimensionality reduction, where points are represented by a single discrete coordinate corresponding to the assigned label.

The application of the proposed method requires the initial detection of density peaks as described in (5). Although this step may hamper the application to datasets where density peaks do not exist, CDP demonstrated broad applicability to challenging datasets (2) where density peaks become easily identifiable as outliers, with high density and high distance, in a density-distance plot. Outliers of the density-distance plot can be selected as points above two thresholds, $\tau_d$ and $\tau_q$. These thresholds can be set either manually by inspecting the density-distance plot, or automatically [i.e., via Otsu thresholding (11)], or to meaningful values for applications where the expected cluster size and density are known. Alternatively, points can be ranked according to a score and the top $N$ points selected as density peaks ($N =$ number of desired clusters). In addition, a combined approach based on thresholding and ranking can be adopted.

There might be cases in which density peaks are not obvious to identify. This can happen in the presence of clusters composed of multiple
subclusters. Peaks of these subclusters may arise in the density-distance plot. In this case, a hierarchical approach is suggested.

Recently, Du et al. (22) proposed an approach that facilitates the identification of density peaks using the graph distance to compute \( d \). Two points in the same cluster are more likely to be connected by a path with low cost. Density peaks that are in distinct clusters instead are more likely to be connected by a path with high cost. Hence, the graph distance of density peaks is likely to be followed by the algorithm, from the cell centroid (density peak) to a point in the dendrite and in the touching region, respectively. (D) Quantitative performances. The F1 score and the Jaccard index are computed with respect to the ground truth. Here, the trained version of the algorithm achieved similar performances to the generic version. These are bounded because of the separation of cells with increased size into multiple subclusters.

Arbitrary nonnegative metrics for edge cost are supported by the proposed methods. Figure S5 reports an application on real high-dimensional data from the bone marrow leukemia dataset (2, 24). Such a dataset includes the gene expression level of 999 genes from 38 samples of patients with three different types of leukemia. The proposed method correctly identified three groups with comparable performances to CDP. In this case, the graph was built with an edge cost derived from the Spearman correlation.

Alternatively, graph pruning can be used together with a cumulative Euclidean path-cost function to limit the connectivity of each node within a restricted neighborhood. Methods for dimensionality reduction can further be used for data preprocessing. Among these, global and nonlinear approaches, such as Isomap (21), are preferable with respect to principal components analysis for highly nonconvex datasets.

The proposed path-based formulation supports the expression of a path cost by evaluating path properties. A nondecreasing path-cost function allows solving the proposed clustering problem using Dijkstra’s SSSP algorithm. Among the possible path-cost functions, minimax penalizes paths with large gaps, which is a generically
desired property. Other nonlinear functions can be designed for specific tasks.

Dijkstra’s SSSP algorithm exhibits a lower than quadratic computational complexity that can be reduced for sparse graphs. Moreover, it does not require storage of the adjacency matrix on memory and can be speeded up by adopting heuristics for optimal path-finding or graph-partitioning techniques. Another advantage of the Dijkstra algorithm is that it supports graphs with multiple edges between two nodes. This opens the possibility to express the distance between two nodes using multiple edges and metrics. During the optimization process, the edge and the metric that are more convenient for the solution will automatically selected. Therefore, we consider this method suitable to be adapted and optimized for specific applications.

Last, existing knowledge can be integrated by defining an appropriate path-cost function, which can be obtained, for instance, by training a path classifier with examples provided by humans or available ground truths. Therefore, it represents an intermediate method between machine and human computation for tasks that would benefit from a collaborative effort (1, 25).

MATERIAL AND METHODS

Data generation

Confocal microscopy

For data shown in Fig. 5, immunofluorescence confocal microscopy was performed using a Leica TCS SP5 microscope, with sequential acquisition to limit signal cross-talking. A murine trachea was collected and fixed in 4% paraformaldehyde at 4°C for 1 hour. Tracheas were cut in two halves along the long axis. Pieces were placed on a microscopy slide and embedded in Fluoromount Aqueous Mounting Medium (Sigma-Aldrich). Three-dimensional (3D) images were acquired using an HCX PL APO CS 20×/0.7 oil immersion objective, and z-step was 1 μm for a total depth ranging from 50 to 112 μm. The bioimaging software Bitplane Imaris (v 9.2.1) was used for computing the 2D maximum intensity projections and for manual cell segmentation. The same
software (Coloc functionality) was used to exclude the background and to create a channel containing only the points belonging to Cd11c-GFP+ cells. This dataset along with the ground truth is provided in data file S1 using the same format described in (2).Mice were maintained in specific pathogen–free facilities at the Institute for Research in Biomedicine, Bellinzona. Experiments were performed in accordance with the Swiss Federal Veterinary Office guidelines, and animal protocols were approved by the veterinarian local authorities.

**Synthetic datasets**
The dataset presented in Fig. 1 is provided in data file S1. The synthetic point distributions in Fig. 4 were generated by the script `benchmark_noise.m` in data file S2. More precisely, three bivariate distributions were overimposed to a uniform background. This was achieved by creating a probability matrix $M$ of size 200 x 200. Then, 1000 samples were extracted from the matrix as follows. For each sample, a pair of random coordinates $(x, y)$ and a random number $p$, from 0 to 1 with uniform probability, were generated. If $M(x, y) > p$ and the point was not already selected, then it was included into the dataset.

Points in the background were selected if the probability of the background was higher than the randomly generated number. Controlling the probability of the background allowed selection of a different number of points in the background with respect to the points in the other distributions.

The synthetic high-dimensional datasets in fig. S4 were generated using MDCGen (Multidimensional Dataset Generator for Clustering) (26), configured to generate two multivariate Gaussian distributions. The number of dimensions ranged from 2 to 100. For each dimension, 15 different distributions (mean and variance) were randomly computed. Code is provided in the script `benchmark_dimensions.m` (data file S2).

**Implementation**
The proposed algorithm was implemented in MATLAB r2017b. A demo for both the generic (minimax) and trainable algorithms are provided respectively in `demo_generic.m` and `demo_trainable.m` in (data file S2). Moreover, an interactive example where the user can visually draw examples of valid paths and invalid paths on a 2D dataset is provided in `demo_interactive.m` and in movie S1.

Dijkstra’s SSSP algorithm, which is at the core of the proposed method, was implemented using vectorization and outperforming the `graphshortestpath()` routine distributed with the Bioinformatics Toolbox of MATLAB r2017b.

The following libraries have been used:
- `DensityClust.m`—implementation of the CDP clustering algorithm (fileexchange/53922-densityclust)
- `dbscan.m`—implementation of the DBSCAN clustering algorithm (www.peterkovceti.com/matlabs/Misc/dbscan.m)
- `FitCSVM`—implementation of a supported vector machine classifier (MATLAB Bioinformatics Toolbox)
- `mdcgen-matlab` (26)—generator of high-dimensional datasets for clustering (https://github.com/CN-TU/mdcgen-matlab).

**Path classifier**
In the example provided in Fig. 2 (C to E), a support vector machine with radial basis function kernel was trained on 25 desired paths and 25 nondesired path fragments. Results can be reproduced using the MATLAB script `demo_trainable.m` provided in data file S2.

**Evaluation**
For the quantitative benchmarking of the proposed method with respect to the ground truth, the F1 score and the Jaccard index ($J$) were computed according to (2). These were defined as follows: $F1 = 2 \times (Recall \times Precision)/(Recall + Precision)$, where $TP$ is the true positive, $FN$ is the false negative, $FP$ is the false positive, and $TN$ is the true negative. Recall $= (TP)/(TP + FN)$, Precision $= TP/(TP + FP)$.

**SUPPLEMENTARY MATERIALS**
Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/5/10/eaax3770/DC1

Proof S1. Unique cluster assignment.

Fig. S1. Graph structure.

Fig. S2. Results on the 12 synthetic datasets provided by ClustEval.

Fig. S3. Performance degradation with respect to number of training paths.

Fig. S4. Benchmark on high-dimensional synthetic datasets.

Fig. S5. Results on the bone marrow leukemia dataset.

Data file S1. Additional datasets used in this article.

Data file S2. MATLAB source code of both the generic and trainable algorithm.

Movie S1. Demo showing the training procedure.

View/request a protocol for this paper from Bio-protocol.

**REFERENCES AND NOTES**
1. M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects. Science 349, 255–260 (2015).
2. C. Wivie, J. Baumbach, R. Röttger, Comparing the performance of biomedical clustering methods. Nat. Methods 12, 1033–1038 (2015).
3. J. MacQueen, Some methods for classification and analysis of multivariate observations, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (University of California Press, 1967), vol. 1, pp. 281–297.
4. M. Estor, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD’96) (AAAI Press, 1996), vol. 96, pp. 226–231.
5. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks. Science 344, 1492–1496 (2014).
6. G. Gerber, I. Horenko, Improving clustering by imposing network information. Sci. Adv. 1, e1500163 (2015).
7. A. Karpate, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Saitahata, V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. 29, 2318–2331 (2017).
8. E. W. Dijkstra, A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).
9. B. Fischer, J. M. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 25, 513–518 (2003).
10. S. Ghidoni, L. Nanni, S. Brahm, E. Menegatti, in Studies in Health Technology and Informatics (IOS Press, 2014), pp. 74–82.
11. N. Otsu, A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66 (1979).
12. V. Ulman, M. Maška, K. E. G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jäldén, H. M. Blau, O. Dzyubachyk, B. Lieveldt, P. Xiao, Y. Li, S.-Y. Cho, A. C. Dufour, J.-C. Olivo-Marin, C. C. Reyes-Aldasoro, J. A. Solís-Lemus, R. Bensch, T. Brox, J. Stegmaier, R. Mikut, S. Wolf, F. A. Hamprecht, T. Esteves, P. Quehas, O. Demirel, L. Malmström, F. Jug, P. J. Pomp, E. Meijering, A. Muñoz-Barrutia, M. Kuzube, C. Ortiz-de-Zerolo, An objective comparison of cell-tracking algorithms. Nat. Methods 14, 1141–1152 (2017).
13. M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Edena, A. Urbiola, T. España, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolcková, M. Štreitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jäldén, H. M. Blau, O. Dzyubachyk, P. Klüzer, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kuzube, C. Ortiz-de-Zerolo, A benchmark for comparison of cell-tracking algorithms. Bioinformatics 30, 1609–1617 (2014).
14. D. U. Pizzagalli, Y. Farsakoglu, M. Palomino-Segura, E. Palladino, M. Marangoni, T. R. Mempel, W. H. Koh, T. T. Murooka, F. Thelen, J. V. Stein, G. Pozzi, M. Thelen, R. Krause,

Pizzagalli et al., Sci. Adv. 2019;5:eaax3770 30 October 2019
S. F. Gonzalez, Leukocyte tracking database, a collection of immune cell tracks from intravital 2-photon microscopy videos. Sci. Data 5, 180129 (2018).

15. J. B. Beltman, A. F. M. Marée, R. J. de Boer, Analysing immune cell migration. Nat. Rev. Immunol. 9, 789–798 (2009).

16. A. E. Carpenter, L. Kamentsky, K. W. Eliceiri, A call for bioimaging software usability. Nat. Methods 9, 666–670 (2012).

17. C. Sommer, C. Strähle, U. Köthe, F. A. Hamprecht, in 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (IEEE, 2011), pp. 230–233.

18. I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, H. S. Seung, Trainable Weka segmentation: A machine learning tool for microscopy pixel classification. Bioinformatics 33, 2424–2426 (2017).

19. R. Mehmood, S. El-Ashram, R. Bie, H. Dawood, A. Kos, Clustering by fast search and merge of local density peaks for gene expression microarray data. Sci. Rep. 7, 45602 (2017).

20. A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101, e215–e220 (2000).

21. J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000).

22. M. Du, S. Ding, X. Xu, Y. Xue, Density peaks clustering using geodesic distances. Int. J. Mach. Learn. Cybernet. 9, 1335–1349 (2018).

23. C. C. Aggarwal, A. Hinneburg, D. A. Keim, in International Conference on Database Theory (Springer, 2001), pp. 420–434.

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

25. G. Re Calegari, G. Nasi, I. Celino, Human computation vs. machine learning: An experimental comparison for image classification. J. Hum. Comput. 5, 13–30 (2018).

26. F. Iglesias, T. Zseby, D. Ferreira, A. Zimek, MDCGen: Multidimensional dataset generator for clustering, in Journal of Classification (Springer, 2019).

Acknowledgments: We are thankful to D. Eynard, M. Thelen, B. Thelen, I. Arini, L. Karagyaur, A. Puffer, G. Pozzi, F. Maffessanti, G. Rovi, P. Zulian, Y. Farsakoglou, and N. Chatziandreu for technical discussion and support; R. D’Antuono and D. Morone for microscopy assistance; and T. Virgilio and M. Palomino-Segura for having generated microscopy data.

Funding: This project was supported financially by the Swiss National Science Foundation (SNF) with the SystemsX.ch iPhD grant 2013/124 to D.U.P., by the SNF with the grant Ambizione 148183 and 176124 to S.F.G., by the Marie Curie Reintegration Grant 612742 to S.F.G., and by the Center for Computational Medicine in Cardiology (CCMC) at ICS.

Author contributions: D.U.P. conceptualized and implemented the algorithm and wrote the manuscript. S.F.G. supervised the biological aspects of the project, provided microscopy data, acquired funding, and wrote the manuscript. R.K. supervised the computational aspects of the project, acquired funding, and wrote the manuscript.

Competing interests: The authors declare that they have no competing interests.

Data and materials availability: All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the authors. This work has been tested on the publicly available datasets included in ClustEval (2). Datasets used in Figs. 1, 5, and 6 are provided in data file S1. MATLAB code is provided in data file S2.