InfoClus: Informative Clustering of High-dimensional Data Embeddings

Fuyin Lai (✉), Edith Heiter, Guillaume Bied, and Jefrey Lijffijt

Ghent University, Belgium {fuyin.lai, edith.heiter, guillaume.bied, jefrey.lijffijt}@ugent.be

Abstract. Developing an understanding of high-dimensional data can be facilitated by visualizing that data using dimensionality reduction. However, the low-dimensional embeddings are often difficult to interpret. To facilitate the exploration and interpretation of low-dimensional embeddings, we introduce a new concept named *partitioning with explanations*. The idea is to partition the data shown through the embedding into groups, each of which is given a sparse explanation using the original high-dimensional attributes. We introduce an objective function that quantifies how much we can learn through observing the explanations of the data partitioning, using information theory, and also how complex the explanations are. Through parameterization of the complexity, we can tune the solutions towards the desired granularity. We propose `InfoClus`, which optimizes the partitioning and explanations jointly, through greedy search constrained over a hierarchical clustering. We conduct a qualitative and quantitative analysis of `InfoClus` on three data sets. We contrast the results on the Cytometry data with published manual analysis results, and compare with two other recent methods for explaining embeddings (RVX and VERA). These comparisons highlight that `InfoClus` has distinct advantages over existing procedures and methods. We find that `InfoClus` can automatically create good starting points for the analysis of dimensionality-reduction-based scatter plots.

Keywords: dimensionality reduction · clustering · explainability.

1 Introduction

Background. Dimensionality-reduction (DR) methods are widely employed to project high-dimensional data into a two-dimensional space, such that it can be visualized. Popular methods are t-SNE [12] and UMAP [13], which are DR methods that are distance-based, local-focused, non-linear, and transductive (terms explained in following paragraphs). Their popularity stems from their effectiveness to retain the local structure of the data, often in the form of cluster structure, which they manage by flexibly warping larger distances. However, this flexibility bring several challenges in the interpretation of embeddings [6,9,18,20]:

The lack of interpretability has its origin in several factors: (1) the methods are *distance-based*, meaning they operate only on the distance matrix and not directly on the feature values. The distances summarize all features with equal weight and thus there is no preference for sparsity in relations between points in the embedding. (2) They are *non-linear and transductive* (i.e., non-parametric): the placement of points in the embedding is entirely free and not a projection. The axes are not interpretable, the embeddings are sensitive to specificities of the data and many XAI methods do not apply (requiring, e.g., the existence of gradients). (3) They are *local-focused* meaning that the distances are distorted. For example, t-SNE also uses a point-specific parameter σ_i to factor out density differences across the data, which can lead to counterintuitive distortions.

Idea behind InfoClus. Rather than aiming to fix these problems, we want to facilitate interpretation of the embeddings, agnostic to the dimensionality reduction algorithm. More specifically, InfoClus is designed to provide a useful starting point for exploring and interpreting of low-dimensional embeddings.

As discussed more in-depth in the related work section below, existing systems enable users to 'explain' hand-selected clusters, or give an overview of the impact of a specific attribute. They rely on the user selecting what to look at. In contrast, the idea behind `InfoClus` is to automatically partition the embeddings into coherent clusters that have a sparse explanation. In other words, to find a clustering that is (1) cohesive on the embedding and (2) such that the points in each cluster share a relatively simple explanation. The explanations that we consider here provide information about how the points in each cluster are different from the data as a whole. Figure 1 shows a diagram that illustrates the `InfoClus` approach.

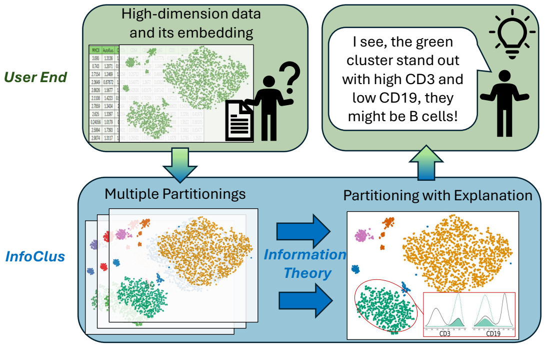

User End
High-dimension data and its embedding
I see, the green cluster stand out with high CD3 and low CD19, they might be B cells!
InfoClus
Multiple Partitionings
Information Theory
Partitioning with Explanation
CD3
CD19

Fig. 1: `InfoClus` workflow

`InfoClus` is a parametrized approach for finding good *partitionings with explanations* (*PwX*s), given a high-dimensional dataset and a low-dimensional embedding. The explanations for a cluster are those that provide most information, as quantified through information theory.

How InfoClus works. We first formalize the problem setting (Section 3) and define *partitioning with explanations*. We then use information theory to quantify the amount of information communicated with a *PwX* and we propose a quantification for the 'complexity' of a *PwX* (Section 3.1). A *PwX* that conveys more information and/or has a lower complexity is better. The complexity is parameterized, enabling users to tune the algorithm to yield simpler or more complex solutions, as desired.

In Section 3.2, we consider how to algorithmically find good *PwX*s. Maximizing the informativeness will lead to *PwX*s that are cohesive in the high-dimensional space, but we need to also ensure that the clustering is cohesive on the two-dimensional embedding. Hence, we constrain possible solutions from InfoClus to clusterings on the embedding that are generated by existing clustering methods. We consider the use of hierarchical and k-means clustering. Such constraints on solutions also have the advantage of limiting the search space of possible partitionings.

For hierarchical clustering, the number of possible partitionings is still very large and optimizing informativeness is not straightforward. We show this problem can be optimized efficiently and effectively using a greedy search algorithm that cuts off branches from the tree induced by the hierarchical clustering. This yields an algorithm that empirically runs in linear time in the number of data points and attributes, per iteration that adds a cluster to the solution.

Evaluation. We evaluate the approach using three different datasets, and study the results qualitatively and quantitatively (Section 4). We include a comparison with the two closest existing methods (RVX [21] and VERA [15]). We consider a Cytometry dataset for which we have a manual expert labeling available based on visualization of the data over feature pairs. We compare the features used in the experts' approach and by InfoClus. We additionally study a dataset with demographic and voting statistics, and the UCI Mushroom data.

Finally, we study the impact of the hyperparameters (Section 4.2) and the runtime of InfoClus (Section 4.3). We conclude the paper with a discussion of the capabilities and limitations of the approach (Section 5).

Reproducibility. All code and results are available on GitHub at https://github.com/aida-ugent/InfoClus.

2 Related work

For brevity, we discuss only the most important examples of work on the more general topic of explaining embeddings and then also the most similar methods.

Related work on explaining embeddings. Several approaches have been developed for explaining the specific positions of points in an embedding, e.g., using inverse projections [5,8] or counterfactuals [1]. Another popular approach to explaining embeddings is feature-centric. We can show the values for an individual feature, but patterns may be found more easily through aggregates such as contour lines [9]. Similarly, one may construct 'rangesets' for class labels [19].

The above approaches can provide insight into the impact of each attribute on the embedding, but do not lead to discovery of the most important features. The second type of approach that has been explored is to start from a user-selected cluster and show how this cluster stands out against the overall data or against another cluster. Stahnke et al. introduced this as 'probing' of projections [20]. In their paper, they showed the distributions of all attributes to users. Recent papers have considered the use of decision trees [2], boosting machines [17], and predicate logic [14] to provide sparse explanations for manually-selected clusters (i.e., explanations using a small subset of features). The latter approach named

DimBridge [14] is notable for also supporting the explanation of hand-drawn lines, by constructing a regression model for the progression along the line.

Finally, there are other DR algorithms, including approaches that aim to directly create coherent—and thus arguably more interpretable—clusters in the embedding. For example, by integrating label information in classification settings [3] or contrastive DR to visualise a clustering [23]. However, in this paper we do not aim to evaluate DR approaches and the approach is agnostic to the DR algorithm. We aim to facilitate the interpretation of any given embedding.

Methods similar to `InfoClus`. There are no approaches directly similar to `InfoClus` that partition an embedding of the data and generate explanations for the partitioning. The most closely related work is Visual Explanations via Region Annotation (VERA) by Policar and Zupan [15]. VERA provides two different types of 'explanations' for embeddings: descriptive and contrastive. They both are built up by finding clusters on the embedding for each data attribute separately. Contrastive explanations select an attribute and value splits that lead to cohesive clusters on the embedding. Descriptive explanations are constructed by merging clusters across attributes that are similar, and then selecting a set that has small overlap.

Another method that does not explicitly seek clusters but may end up giving a similar result to `InfoClus` is by Thijssen et al. [21]. We will refer to it as Relative Value eXplanations (RVX) and RV$_\sigma$X for the variance-based variant. RVX provides an explanation in the form of a feature per data point that stands out most, in terms of value or variance. It employs smoothing, leading to visual clusters often sharing a single most-outstanding attribute. We include an empirical comparison between `InfoClus`, VERA, and RVX in Section 4.

3 InfoClus – Step by Step

Preliminaries. We assume given a dataset with n points $X = \{x_1, x_2, \ldots, x_n\}$, m attributes $A = \{a_1, a_2, \ldots, a_m\}$, and its corresponding two-dimensional embedding Y. We refer to a *partitioning* of Y as a list of sets $\mathcal{C} = [c_1, \ldots, c_r]$ such that each point in Y belongs to exactly one c_i. We refer to an *explanation* for a *partitioning* as choice of representative attributes and their corresponding distribution for each cluster in the partitioning.

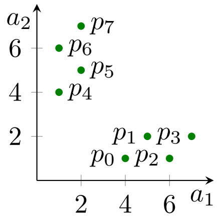


Fig. 2: Example dataset.

Definition 1. *A* partitioning with explanations *(PwX) of dataset X is a partitioning $\mathcal{C}$ of its embedding Y with associated explanations, represented as a tuple $(\mathcal{C}, \mathcal{E})$, where $\mathcal{C} = [c_1, \ldots, c_r], \mathcal{E} = [e_1, \ldots, e_r]$. $\mathcal{E}$ denotes the list of explanations, where $e_i = \{a_i^1, a_i^2, \ldots\} \subseteq A$ is the set of attributes selected to explain cluster c_i.*

Fig. 2 shows a small dataset (8 points, 2 attributes) that will serve as a running example to illustrate the concepts in this section (note here that $Y = X$).

Example. $(\mathcal{C}', \mathcal{E}')$ with $\mathcal{C}' = [c_1 : \{p_0, p_1, p_2, p_3\}, c_2 : \{p_4, p_5, p_6, p_7\}]$, $\mathcal{E}' = [e_1 : \{a_2\}, e_2 : \{a_1\}]$ is a *PwX* of the dataset shown in Fig. 2.

Section outline. Section 3.1 presents measures for how much information users may learn from a *PwX* and the complexity of a *PwX*, which together define what is an informative clustering. Section 3.2 introduces `InfoClus`, a methodology aimed at finding coherent and informative *PwX*s.

3.1 Explanation Ratio

We assume that the goal of interpreting the embedding is to learn about the distribution of the data. The visualization of the embedding enables users to identify groups of points that form a cluster. For example, t-SNE indeed tends to produce embeddings with meaningful cluster structure [10]. Hence, we can model the interpretation as a communication process where the user learns about the data through visualizations of distributions for clusters present in the embedding.

To arrive at a quantification for what a user learns, we take inspiration from the framework on subjective interestingness [7]. We can quantify how much a user learns using information theory, by quantifying the reduction in uncertainty a user has about the data. This means we have to make an assumption about what the user knows already—which is known as the *prior distribution*—and what the user knows after observation of a *PwX*.

Following De Bie [7], the *information content* of an observation is given by the surprisal of this observation: $IC(X) = -\log P(X)$. Equivalently, but from a different perspective, if we consider how much we learn by updating a prior distribution to a posterior, this is given by the KL-divergence between the two distributions. The amount of information that we learn by updating a prior to a more specific posterior distribution is quantified as follows:

Definition 2. *The* information content *I for a PwX is the sum of* information content *for each cluster c_i and each attribute a_i^j in the PwX, which is given by*

$$I_i^j = |c_i| * D_{KL}\big(P_i^j \| Q^j\big)\,, \tag{1}$$

where $|c_i|$ is the size of the cluster, P_i^j is the distribution of attribute a^j in cluster c_i, and Q^j is the distribution of attribute a^j in the full dataset.

Here we assumed that the prior information a user has about the data are the marginal distributions for each attribute in the data. We present the distributions for clusters in visual form, but we assume users learn about the mean and variance statistics only, but with infinite precision. In practice, we may not know either, but the approach may still be helpful even if the details of how we model the beliefs of the user are wrong.

In `InfoClus`, we model real-valued attributes with Gaussian distributions, and categorical attributes with a categorical distribution as a default. The implementation of `InfoClus` supports both numerical and categorical attributes, but not mixed data, because the KL-divergence is defined for discrete and continuous variables, but they are not comparable.

Example. We assume continuous attribute a_2 follows a Gaussian distribution. By computing the empirical mean and variance in the full dataset and the cluster c_1 respectively, we have a_2 follows $\mathcal{N}(3.5, 4.75)$ on the full dataset and $\mathcal{N}(1.5, 0.25)$ in c_1 of *PwX* $(\mathcal{C}', \mathcal{E}')$. The information carried by a_2 in c_1 is: $I_1^2 = 4 \times D_{KL}(\mathcal{N}(1.5, 0.25) \| \mathcal{N}(3.5, 4.75)) = 5.68$.

Yet, the more attributes are selected to inform a user about a partitioning, the more effort is needed for users to understand them. This leads us to define the following notion of complexity, with α and β parameterizing a user's learning sensitivity for information:

Definition 3. *The* complexity *of a PwX* $(\mathcal{C}, \mathcal{E})$ *is*

$$\alpha + (\sum_{i=1}^{r} \sum_{j=1}^{|e_i|} |a_i^j|)^\beta \,, \tag{2}$$

where a_i^j *is an attribute belonging to* $e_i \in \mathcal{E}$*, and* $|a_i^j|$ *refers to the count of statistics needed to describe the distribution* a_i^j *follows.* α *and* β *are hyper-parameters.*

Example. Assuming attributes a_1, a_2 in Fig. 2 follow a Gaussian distribution, we have $|a_1| = |a_2| = 2$, since for a Gaussian distribution, the mean and variance statistics define the probability distribution. Then, given $\alpha = 1, \beta = 2$, the *complexity* for *PwX* $(\mathcal{C}', \mathcal{E}')$ is $1 + 4^2 = 17$.

To balance the information carried by a *PwX* and the human cognitive effort necessary to interpret it, we introduce the concept of an explanation ratio:

Definition 4. *The* explanation ratio $R_{\alpha,\beta}$ *of a PwX* $(\mathcal{C}, \mathcal{E})$ *with* r *clusters is*

$$R_{\alpha,\beta}(\mathcal{C}, \mathcal{E}) = \frac{\sum_{i=1}^{r} \sum_{j=1}^{|e_i|} I_i^j}{\alpha + (\sum_{i=1}^{r} \sum_{j=1}^{|e_i|} |a_i^j|)^\beta} \,, \tag{3}$$

where i *indexes the clusters and* j *the attributes of explanation* e_i*.*

Example. Given $\alpha = 1, \beta = 2$, the *explanation ratio* for *PwX* $(\mathcal{C}', \mathcal{E}')$ is $R_{1,2}(\mathcal{C}', \mathcal{E}') = (I_1^2 + I_2^1)/(1 + (|a_2| + |a_1|)^2) = 0.67$.

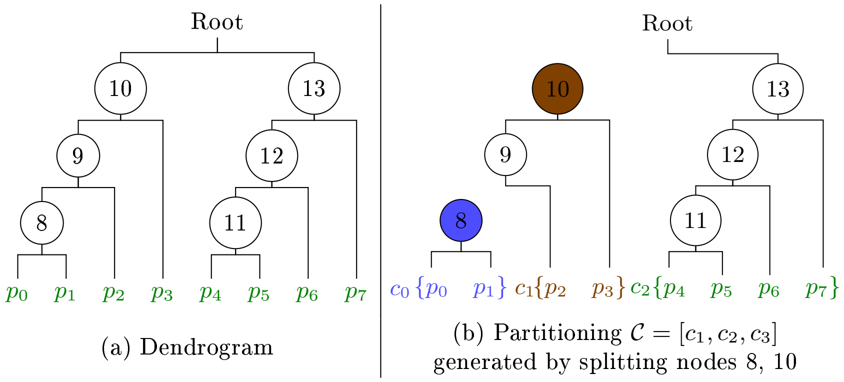
Problem statement. `InfoClus` aims to partition the dataset into clusters that are cohesive in the embedding and the high-dimensional space. The clustering is shown on the embedding; information on the high-dimensional space is provided through a few selected attributes. The partitioning and explanations are chosen such that they maximize the *explanation ratio* aiming to optimize the ratio of the amount of information provided over the effort required to interpret it.

3.2 InfoClus: finding cohesive and informative PwXs efficiently

`InfoClus` seeks to partition the embeddings into visually coherent clusters with high *explanation ratio*. To do so, existing clustering algorithms (e.g. hierarchical clustering or k-means) are used to generate candidate partitionings (ensuring visual coherence), among which the *PwX* with highest *explanation ratio* is retained. In the following, we focus on hierarchical clustering, which we advise using as a default candidate generator.

Hierarchical clustering strikes a suitable balance with a large number of possible partitionings, but since these are structured in a tree we can still optimize efficiently over the space of candidates[1]. We proceed to detail the procedure used to navigate the search space for partitionings implied by hierarchical clustering, and sketch algorithmic details.

Search for Partitionings. Hierarchical clustering constructs a hierarchy of clusters, which may be seen as organizing data in a tree structure (dendrogram). For instance, agglomerative hierarchical clustering builds a tree from the ground up: each observation initially defines its own cluster, and clusters are progressively greedily merged based on a chosen linkage. We constrain our search space to partitionings compatible with the hierarchical clustering. More specifically, rather than relying solely on partitionings formed by slicing the dendrogram at a specific level, we consider a compatible partitioning with r clusters as consisting of $r-1$ clusters obtained from cutting nodes at any level of the dendrogram, and the final cluster encompassing the remaining points. For instance, Fig. 3 illustrates a partitioning $\mathcal{C}$ compatible with a dendrogram in our toy example.


Root
10
13
9
12
8
11
p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7
Root
10
13
9
12
8
11
$c_0\{p_0$ $p_1\}$ $c_1\{p_2$ $p_3\}$ $c_2\{p_4$ p_5 p_6 $p_7\}$

(a) Dendrogram

(b) Partitioning $\mathcal{C} = [c_1, c_2, c_3]$ generated by splitting nodes 8, 10

Fig. 3: How hierarchical clustering works as a candidate partitioning generator.

Yet, the number of partitionings compatible with a dendrogram is too large for enumeration[2]. Accordingly, `InfoClus` resorts to a greedy strategy to iteratively find a sequence of *PwX*s with an increasing number of clusters and high *explanation ratio*s, where each *PwX* is obtained by splitting one node extra from the previous *PwX*. More precisely, at iteration k, `InfoClus` considers partitionings with $k+1$ clusters by enumerating all *PwX*s formed by cutting any subtree

[1] Its use will be compared to that of k-means in Section 4.1.

[2] There are 56 different partitionings in a dendrogram generated by a 8 points dataset.

of the *PwX* we chosen at $k-1$ iteration. The partitioning with highest *explanation ratio* is selected as the *PwX* at iteration k. Thus, K iterations of the greedy procedure yield a sequence of K *PwX*s of size 2, ..., $K+1$. The *PwX* in this sequence with highest *explanation ratio* is the one finally selected.

Example. Fig. 3a presents a dendrogram generated for the example dataset by hierarchical clustering. Given $\alpha = 1, \beta = 2$, we have 5 candidate partitionings with 2 clusters: those obtained by splitting subtrees with root of node 8, 9, 10, 11 or 12. After computing the best explanations for each of these partitionings, we find the associated *explanation ratio*s to be $0.18, 0.34, 0.67, 0.18$ and 0.34 respectively. The partitioning generated by splitting node 10 is the best one with $R_{1,2} = 0.67$; this corresponds to *PwX* $(\mathcal{C}', \mathcal{E}')$. The next iteration considers partitionings with 3 clusters that could be obtained by splitting one more node (node $8, 9, 11$ or 12) out of *PwX* $(\mathcal{C}', \mathcal{E}')$. For instance, splitting node 8 out of *PwX* $(\mathcal{C}', \mathcal{E}')$ would lead to a new partitioning $[c_1 : \{p_0, p_1\}, c_2 : \{p_2, p_3\}, c_3 : \{p_4 ... p_7\}]$.

Time budget. In the following, we call an **iteration** the process through which `InfoClus` tries different nodes to find a good partitioning with k clusters based on the partitioning with $k-1$ previously selected clusters. Iterations can be conducted as long as there are possible nodes to consider, but a partitioning of a dataset with almost as many clusters as points might not be desirable. Therefore, in the implementation of `InfoClus` we introduce a time budget t, which imposes `InfoClus` to terminate before time t.

Search for Explanations. The best explanations for a partitioning in a dataset with homogeneous attributes (e.g. only numeric or categorical) can be found by applying a greedy strategy in each cluster: iteratively adding attributes with highest *information content* to a cluster's explanations until the *explanation ratio* $R_{\alpha,\beta}$ decreases when a new attribute is added. In the code implementation, the number of attributes selected for each cluster can be controlled by hyper-parameters *minatt* and *maxatt*, which serve as lower and upper bounds respectively for the number of attributes used to explain each cluster.

Example. Given $\alpha = 1, \beta = 2$, consider computing the best explanation $\mathcal{E}$ for partitioning $\mathcal{C}' = [c_1 : \{p_0, p_1, p_2, p_3\}, c_2 : \{p_4, p_5, p_6, p_7\}]$ in the example dataset (Fig. 2). We first compute *information content* for both attributes x, y, and both clusters c_1 and c_2. We get $I_1^x, I_1^y, I_2^x, I_2^y = 2.88, 5.68, 5.68, 2.88$. Next, we assign a_2 to c_1 and a_1 to c_2 to make sure that at least one attribute is assigned to each cluster. At this stage, the *explanation ratio* $R_{1,2}$ is 0.67. Then we assign the next attribute (a_1) with highest I to its corresponding cluster (c_1). This results in $R_{1,2} = 0.38$. Since $0.38 < 0.67$, we stop searching, and serve $\mathcal{E}' : [e_1 = \{y\}, e_2 : \{x\}]$ as the best explanation for $\mathcal{C}'$.

Pseudo-code. Full pseudo-code for `InfoClus` is shown in Algorithm 1.

4 Experiments

In Section 4.1 we first present experiments on three datasets to evaluate the ability of `InfoClus` to generate useful insights, including a comparative assessment

Algorithm 1 InfoClus

```
Input: dataset $X$, its embedding $Y$, attributes $A$, $minAtt$, $maxAtt$, $\alpha, \beta$, and $t$
Output: a $PwX$ $(\mathcal{C}, \mathcal{E})$
Dendrogram = hierarchicalClustering($Y$)
Nodes = getSplittingNodes(Dendrogram)% Potential splits in dendrogram
best$R$, best$R^1 = 0$, best$\mathcal{C}$, best$\mathcal{C}^1$ = the full dataset
$k = 1$
while $t$ has not been used up & Nodes is not None do
    % iteration starts
    $k = k + 1$
    best$R^k = 0$, best$Node^k = None$
    for $Node$ in Nodes do
        $\mathcal{C}^k$ = split $Node$ out of best$\mathcal{C}^{k-1}$ to get partitioning
        $\mathcal{E}^k$ = find best explanation for $\mathcal{C}^k$ under bounds $minAtt$ and $maxAtt$
        compute $R_{\alpha,\beta}(\mathcal{C}^k, \mathcal{E}^k)$
        if best$R^k \leq R_{\alpha,\beta}(\mathcal{C}^k, \mathcal{E}^k)$ then
            best$R^k = R_{\alpha,\beta}(\mathcal{C}^k, \mathcal{E}^k)$, best$\mathcal{C}^k = \mathcal{C}^k$, best$Node^k = Node$
        if best$R \leq R_{\alpha,\beta}(\mathcal{C}^k, \mathcal{E}^k)$ then
            best$R$, best$\mathcal{C}$, best$\mathcal{E}$ = $R_{\alpha,\beta}(\mathcal{C}^k, \mathcal{E}^k)$, $\mathcal{C}^k$, $\mathcal{E}^k$
    Update Nodes by removing best$Node^k$ and its ancestors
return $PwX$ (best$\mathcal{C}$, best$\mathcal{E}$)
```

with respect to the state of art [15,21]. We then study the sensitivity of `InfoClus` to the hyper-parameters (Section 4.2), and its scalability (Section 4.3).

4.1 Case study

To evaluate the usefulness of `InfoClus`, we conduct case studies on three datasets covering different domains and attribute types: the `Cytometry`, `GSE` (German Socio-Economic) and `Mushroom` data; details of which will be introduced below. For the `Cytometry` data, we compare insights given by `InfoClus` to manual annotation by domain experts, to `InfoClus` with k-means, and with RVX [21] and VERA [15]. Embeddings studied in the case studies are generated with t-SNE.

Cytometry data [16]. The `Cytometry` dataset represents different cell types of mouse splenocytes, described by 9 columns/features recording the stain of different markers. We conduct the case study on a random sample of 2500 instances from the original dataset, which contains more than 300K instances[3].

Advantageously for this case study, the data has been annotated by domain experts through manual gating [16]. This enables us to compare insights derived by `InfoClus` with an expert procedure.

Manual gating, routinely used in the analysis of cytometry data [11,16], proceeds by iteratively plotting cells on two-dimensional scatter plots with respect

[3] As in [16], the analysis is conducted on a subsample for the sake of limiting t-SNE's runtime; the scalability of `InfoClus` with larger data is investigated in Section 4.3.

to a chosen subset of features, and selecting at each step a subset of cells on which to further focus in the next iteration [16]. The sequence of features studied at each step is determined by expert knowledge. Fig. 4a1, reproduced from [16], represents the results of this process in a t-SNE embedding of the `Cytometry` dataset. The manual gating process is illustrated by Fig. 4b1, in which arrows indicate the sequence in which the analysis is conducted.

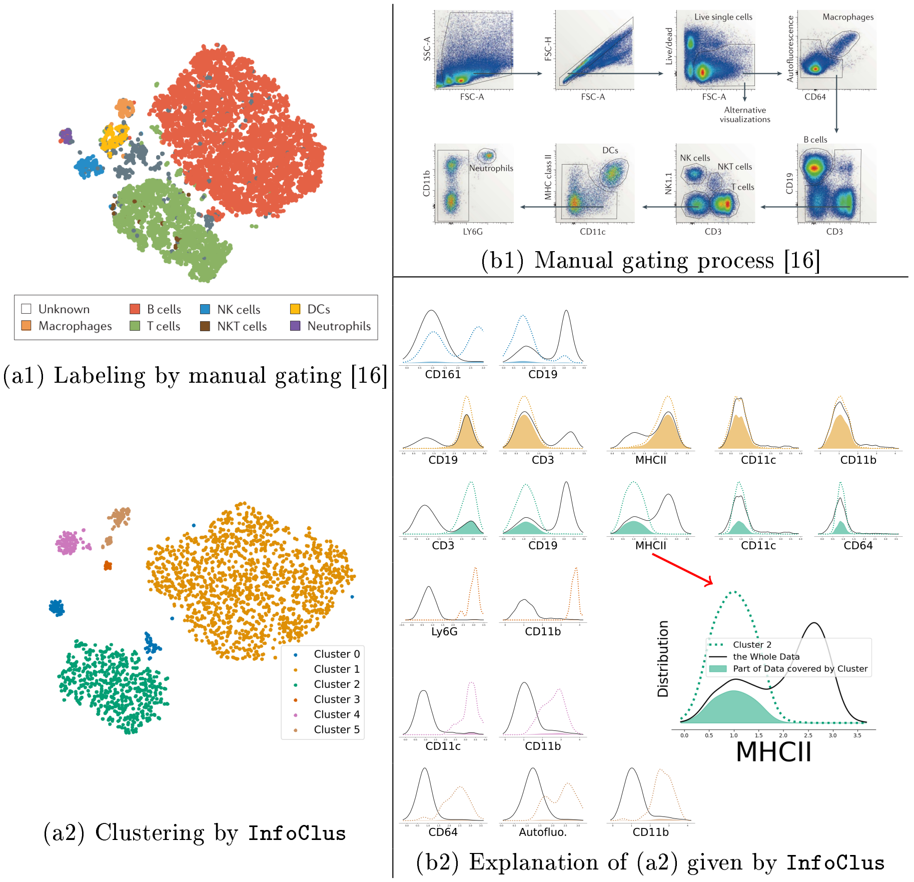
Qualitative analysis. We applied `InfoClus` on `Cytometry` with hyperparameters $\alpha = 1500$, $\beta = 1.5$, $minatt = 2$ and $maxatt = 5$. Given a 5 second limit for iterations (t in Algorithm 1), `InfoClus` conducted 9 iterations. Fig. 4a2 and Fig. 4b2 display the resulting 6 clusters, and their explanations (i.e., the distributions of features selected as explanations for each cluster).

Combining the partitioning and explanations provided by `InfoClus` (Fig. 4a2, b2), cluster 1 and 2 stand out—the former being characterized by high CD19 and low CD3, and the latter by high CD3 and low CD19. These findings align with those of experts, which also use CD19 and CD3 to identify B and T cells—as displayed in the 2nd row, 3rd and 4th sub-figures in Fig. 4b1. Similar conclusions apply to cluster 3 (orange in a1; characterized by higher Ly6G and CD11b; labeled by experts as Neutrophils); cluster 4 (purple; higher CD11c; labeled as DCs), and cluster 5 (Ochre; higher CD64 and Autofluo; labeled as Macrophages). Altogether, the clusters and explanations generated by `InfoClus` align with those of domain experts. `InfoClus` may even help in uncovering possible inconsistencies in labels generated by manual gating: for instance, points in a cluster in the top-left of Fig. 4a1 share similar values in high dimension, but are labeled as different cells—red (B cells) and purple (Neutrophils) labels.

Comparison We now compare `InfoClus` to other methods aimed at explaining embeddings: `InfoClus` with k-means, RVX [21], and VERA [15]. `InfoClus` with k-means proceeds by running k-means on a dataset for a range of K different values ($k = 3, 4, 5, \ldots, 32$ here), generating K partitionings. Each partitioning is then associated to a set of explanations with best PwX in terms of *explanation ratio*; the final partitioning selected is the one with highest PwX among the K candidates. Descriptions of RVX and VERA have been presented in Section 2.

Fig. 5 shows the partitionings and explanations provided by `InfoClus` with k-means, RVX, and VERA on the `Cytometry` dataset. We compare these to the `InfoClus` result of Fig. 4a1. Explanation ratios for `InfoClus` and `InfoClus` with k-means are 5.61 and 5.39 respectively (Fig. 5a). With k-means, clusters are roughly evenly sized and visually less in line with the structure of the embedding than those generated by `InfoClus`: the top-right block of points in the embeddings is split in three (blue, green and pink clusters), while the disconnected point sets in the top-left corner are considered a single cluster (red).

In Fig. 5b we find that RVX also shows clear cluster structure, even though it is not a clustering method. However, the largest cluster lacks any clear visual structure. Note also that by definition of the method, each point is associated to exactly one attribute and not more. This is probably why the B cells that



(a1) Labeling by manual gating [16]

(a2) Clustering by `InfoClus`

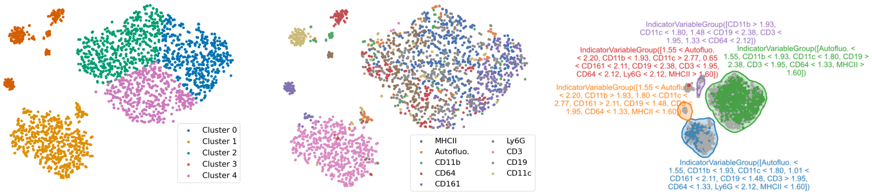
(b1) Manual gating process [16]

(b2) Explanation of (a2) given by `InfoClus`

Fig. 4: `Cytometry` dataset. Panels a1 and a2 show t-SNE embeddings colored by manual gating [16] and by `InfoClus` respectively (the embeddings are slightly different because the available data is not equivalent to that in the original paper and we use a different implementation of t-SNE). Panel b1 shows how manual gating proceeds to label cells in a2. Panel b2 displays the explanations selected by `InfoClus` for each cluster (the color coding maps explanations to their respective clusters). Each plot displays Kernel Density Estimates (KDE) of the distributions of the attribute in the cluster (dotted line) and on the full dataset (solid line). Filled color represents a KDE of the cluster scaled by how many points are covered by the cluster in the full data.

comprise the top-right cluster do not form a visually coherent cluster, they are not defined through any single attribute.

The clusters generated by VERA (Fig. 5c) are visually coherent, but the annotations involve almost all attributes in the data. The threshold-based ex-

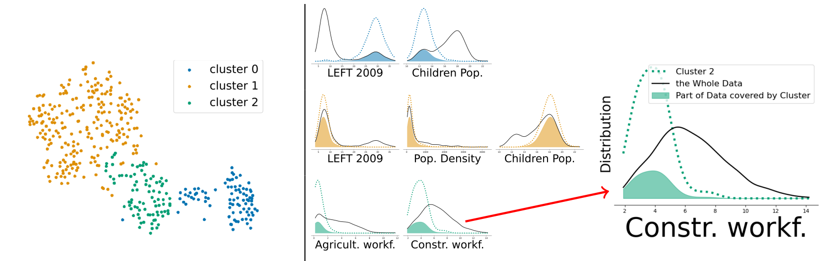


(a) `InfoClus` with k-means (b) RVX (c) VERA

Fig. 5: Results of other methods aimed at explaining embeddings on `Cytometry`

planations given by VERA also cause the coverage of the clusters to be low, i.e., the explanations cover only part of the points in the identified regions.

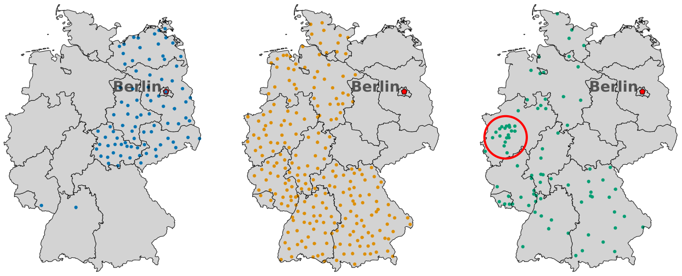
German Socio-Economics (`GSE`) data [4]. This dataset describes 412 administrative districts of Germany in terms of 31 socio-demographic attributes, such as distributions of population age and labor sector, and voting percentages, recorded at the 2009 elections. We set hyper-parameters on this dataset as follows: $\alpha = 50$, $\beta = 1.5$, $minatt = 2$ and $maxatt = 5$. Given a 5 second limit for iterations, `InfoClus` conducted 18 iterations and returned a partitioning with 3 clusters, presented alongside their explanations in Fig. 6.



(a) `InfoClus` partitioning (b) Explanation for the left partitioning

Fig. 6: `InfoClus` analysis in GSE

In Fig. 6, the voting ratio of the LEFT party in 2009 is selected as the most interesting attribute in both cluster 0 and 1, with high and low support respectively. Cluster 2 stands out for the low shares of its workforce working in agriculture and construction, hinting that cluster 2 might correspond to urban areas. We verify this insight using metadata of `GSE`: 86% of districts in cluster 2 are urban, compared of 28% of all districts in the data. Out of interest, we visualized the partitioning of districts on the German map (Fig. 7). Indeed, Cluster 0 and 1 are clearly geographically separate, and correspond to former East and West Germany, and cluster 2 corresponds in a large part to the Ruhr area, shown in the red circle in Fig. 7c. Note that this geographic coherence is

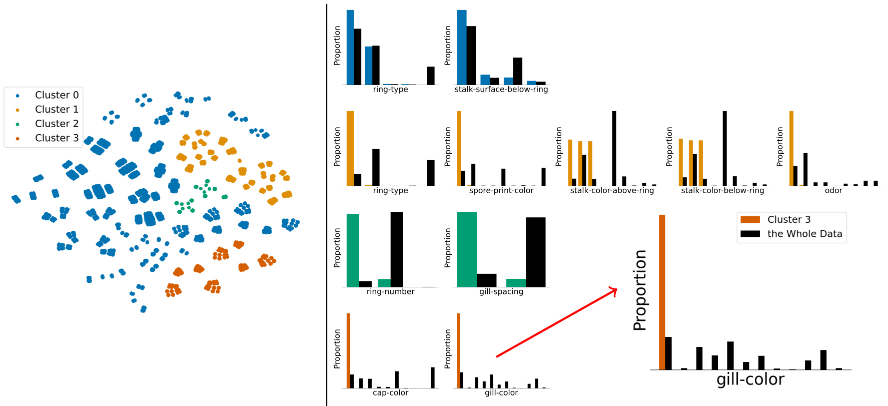


(a) Cluster 0 (b) Cluster 1 (c) Cluster 2

Fig. 7: Visualization of `InfoClus` clusters on the map of Germany.

striking as t-SNE and `InfoClus` do not know about the geographic location of the districts: these clusters emerged purely from demographic and voting statistics.

UCI Mushroom data [22]. `Mushroom` consists of 8124 instances characterized by 22 categorical attributes that describe 23 species of gilled mushrooms within the Agaricus and Lepiota family. We considered this data to investigate `InfoClus`' ability to interpret embeddings of categorical data. We set hyperparameters on this dataset as follows: $\alpha = 800$, $\beta = 1.5$, $minatt = 2$ and $maxatt = 5$. Given a limit of 30 seconds for iterations, `InfoClus` conducted 10 iterations and returned a partitioning with 4 clusters, shown in Fig. 8.



(a) `InfoClus` partitioning (b) Explanation for the left partitioning

Fig. 8: `InfoClus` analysis on Mushroom dataset

Four visually coherent clusters emerge, differing from the dataset as a whole with respect to several attributes: for instance, large rings and orange print colors for cluster 1, low ring number and close gill spacing for cluster 2, purple caps and red gill colors for cluster 3. Unfortunately, a lack of background on mushrooms prevents us from further interpreting these findings.

4.2 Hyper-parameter analysis

We now investigate the sensitivity of `InfoClus`' results to hyperparameter selection, using the `Cytometry` dataset. Since t, *minatt* and *maxatt* (respectively: imposed iteration time, minimum and maximum features per cluster) should be fixed based on users' preference, we focus on the role of hyper-parameters α and β. In the following experiment, we set $minatt = 2$, $maxatt = 5$, and $t = 5$.

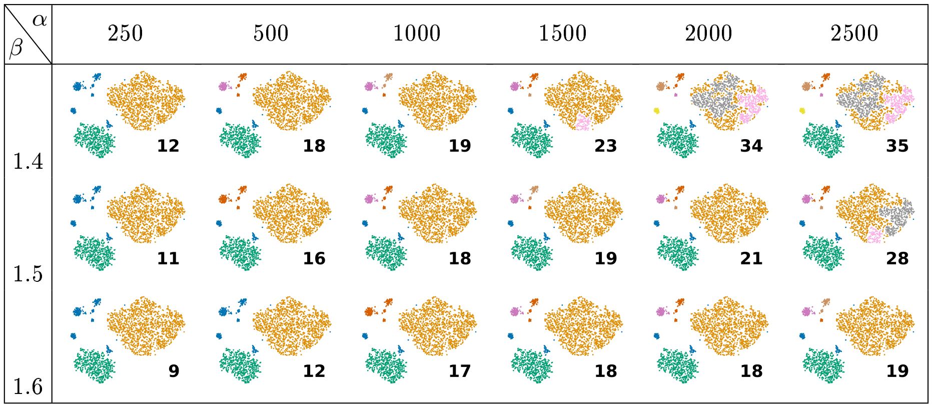


Fig. 9: Hyper-parameter analysis on `Cytometry`. partitionings generated by `InfoClus` under different α and β are shown, along with the numbers of attributes selected for all clusters (bottom right corner).

Fig. 9 displays the evolution of `InfoClus` clusters and the total number of explanatory attributes selected for different values of α and β (α varying from 250 to 2500, β from 1.4 to 1.6). The number of `InfoClus` clusters increases when α increases, and decreases when β increases. Similarly, the number of selected attributes increases with α increases, and decreases when β increases. This was to be expected given the definition of the *explanation ratio* (R) (Def. 4). α and β both intervene in R's denominator $\alpha + (\sum_{i=1}^{r} \sum_{j=1}^{|e_i|} |a_i^j|)^\beta$. Increasing β promotes sparser explanations. When α increases, the weight of $(\sum_{i=1}^{r} \sum_{j=1}^{|e_i|} |a_i^j|)^\beta$ decreases in the denominator, with similar effect to a decrease in β.

Nevertheless, partitionings does not change heavily when α and β vary. It is a good signal for users that even they have few knowledge about `InfoClus`, `InfoClus` might also return a reasonable result. We also suggest users to select α between $n/10$ and n, and β around 1.5.

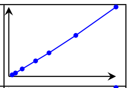
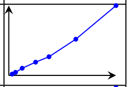
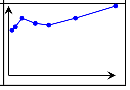
Sample size ($\times 10^3$)		2.5	5	10	20	30	50	80	Trend
Time (s)	Initialization rows 3-6 in Algo. 1	7.1	15.6	34.6	71.8	109	190	322	
	Avg. per iteration rows 11-18 in Algo. 1	0.6	1.4	3.3	6.1	8.6	16.7	32.2	
Time ($\times 10^{-4}$s)	Avg. per partitioning rows 12-18 in Algo. 1	2.6	2.8	3.3	3.0	2.9	3.3	4.0	

Table 1: Scalability of `InfoClus` on samples from `Cytometry` with various size

4.3 Scalability Analysis

We next turn to the analysis of `InfoClus`' scalability. `InfoClus`' runtime can be divided into two components: initialization and searching for a good *PwX*. The initialization proceeds: by i) computing embeddings, if none are user-provided; ii) computing a hierarchical clustering of the embeddings; iii) computing statistics used downstream for computing *explanation ratio*s (means and variances of candidate clusters), see Appendix **??** for details. The search for the optimal *PwX* then proceeds using these components.

As users may impose an upper bound t on total runtime, measuring the overall time spent on the search phase is meaningless. We instead list the time spent for initialization and describe the search phase by the time spent per iteration and per partitioning. The time spent per iteration corresponds to the average time spent to find the best k-cluster partitioning starting from a partitioning with $k-1$ clusters. The time spent per partitioning is the time used to select the top explanations of a partitioning with respect to the *explanation ratio* R.

Table 1 reports runtimes of `InfoClus` components, computed on samples of increasing size drawn from the original `Cytometry` dataset. The experiments were conducted on a machine equipped with a 13th Gen Intel(R) Core(th) i7-1365U (5.20 GHz) and 16.0 GB of RAM.

First, initialization is a rather costly process although it is only run once: 5 minutes are required to initialize a dataset with size of 80,000. Second, we turn to the average times per iteration and partitioning in the *PwX* search phase. These two quantities can be formally related: the cost per iteration is at most the product of the data size n with the cost per partitioning[4]. The time needed per partitioning seems to fluctuate as data size n varies, with no obvious increase or decrease trend. The absolute time for each partitioning is quite small with an order of magnitude of 10^{-4} seconds. Altogether, these results indicate that `InfoClus` can credibly scale to problems of at least moderate size.

[4] Partitionings are generated by splitting notes from the dendrogram generated by the hierarchical clustering. This dendrogram has at most $n-1$ nodes, including its root.

5 Discussion

In this paper, we introduced InfoClus, which finds a partitioning of an embedding that is cohesive in both the high-dimension and low-dimension space and also returns explanations in the form of attributes that stand out for each cluster. By constraining the solution with a hierarchical clustering, using greedy optimization with locally optimal cuts, we obtain an efficient and effective algorithm. By means of two hyperparameters, users can control how fine-grained (i.e., complex) the output is. We have found from the empirical results that the solutions can be helpful and that have distinct advantages over existing approaches.

The proposed method and experiments have several limitations that could be addressed in future work. We do not know the computational complexity to find optimal PwXs compatible with hierarchical clustering, it may be possible to identify better optimization strategies than greedy. Also, the search strategy can lead to disconnected clusters because they can be nested. To prevent this, a different approach would be necessary. Finally, it would be interesting to consider how to unify the *information content* for discrete and numeric variables, enabling the use of InfoClus on mixed data.

Acknowledgments. The research leading to these results has received funding from the Special Research Fund (BOF) of Ghent University (BOF20/IBF/117), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project no. G0F9816N, 3G042220, G073924N). Funded by the European Union (ERC, VIGILIA, 101142229). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. For the purpose of Open Access the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. EH is supported by a doctoral scholarship from the FWO (project number: 11J2322N)

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Artelt, A., Schulz, A., Hammer, B.: "Why here and not there?" – diverse contrasting explanations of dimensionality reduction. https://arxiv.org/abs/2206.07391
2. Bibal, A., Clarinval, A., Dumas, B., Frénay, B.: IXVC: An interactive pipeline for explaining visual clusters in dimensionality reduction visualizations with decision trees. Array p. 100080 (2021)
3. de Bodt, C., Mulders, D., López-Sánchez, D., Verleysen, M., Lee, J.A.: Class-aware t-SNE: cat-SNE. In: Proc. of ESANN (2019)
4. Boley, M., Mampaey, M., Kang, B., Tokmakov, P., Wrobel, S.: One click mining: interactive local pattern discovery through implicit preference and performance learning. In: Proc. of ACM SIGKDD IDEA Workshop. pp. 27–35 (2013)
5. Cavallo, M., Ç. Demiralp: A visual interaction framework for dimensionality reduction based data exploration. In: Proc. of CHI (2018)

6. Chatzimparmpas, A., Martins, R.M., Kerren, A.: t-viSNE: Interactive assessment and interpretation of t-SNE projections. IEEE TVCG **26**(8), 2696–2714 (2020)
7. De Bie, T.: An information theoretic framework for data mining. In: Proc. of ACM SIGKDD. pp. 564–572 (2011)
8. Espadoto, M., et al.: UnProjection: Leveraging inverse-projections for visual analytics of high-dimensional data. IEEE TVCG **29**, 1559–1572 (2023)
9. Faust, R., Glickenstein, D., Scheidegger, C.: DimReader: Axis lines that explain non-linear projections. IEEE TVCG **25**(1), 481–490 (2019)
10. Kobak, D., Berens, P.: The art of using t-SNE for single-cell transcriptomics. Nature Communications **10**, 5416 (2019)
11. Liu, P., et al.: Comprehensive evaluation and practical guideline of gating methods for high-dimensional cytometry data: manual gating, unsupervised clustering, and auto-gating. Briefings in Bioinformatics **26**(1), bbae633 (2025)
12. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**(86), 2579–2605 (2008)
13. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. https://arxiv.org/abs/1802.03426 (2018)
14. Montambault, B., Appleby, G., Rogers, J., Brumar, C.D., Li, M., Chang, R.: DimBridge: Interactive explanation of visual patterns in dimensionality reductions with predicate logic. IEEE TVCG **31**(1), 207–217 (2024)
15. Policar, P.G., Zupan, B.: VERA: Generating visual explanations of two-dimensional embeddings via region annotation. arXiv:2406.04808 (2024)
16. Saeys, Y., Van Gassen, S., Lambrecht, B.: Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nat Rev Immunol **16**, 449–462 (2016)
17. Salmanian, P., Chatzimparmpas, A., Karaca, A.C., Martins, R.M.: DimVis: Interpreting visual clusters in dimensionality reduction with explainable boosting machine. arXiv:2402.06885 (2024)
18. Sedlmair, M., Brehmer, M., Ingram, S., Munzner, T.: Dimensionality reduction in the wild: Gaps and guidance. Tech. Rep. 2012-03, UBC Computer Science (2012)
19. Sohns, J.T., Schmitt, M., Jirasek, F., Hasse, H., Leitte, H.: Attribute-based explanation of non-linear embeddings of high-dimensional data. IEEE TVCG **28**(1), 540–550 (2022)
20. Stahnke, J., Dörk, M., Müller, B., Thom, A.: Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. IEEE TVCG **22**(1), 629–638 (2015)
21. Thijssen, J., Tian, Z., Telea, A.: Scaling up the explanation of multidimensional projections. In: EuroVis Workshop on Visual Analytics (2023)
22. UCI Machine Learning Repository: Mushroom (1981). `https://doi.org/https://doi.org/10.24432/C5959T`, last accessed 12 March 2025
23. Xia, J., Huang, L., Lin, W., Zhao, X., Wu, J., Chen, Y.: Interactive visual cluster analysis by contrastive dimensionality reduction. IEEE TVCG **29**(1), 734–744 (2024)