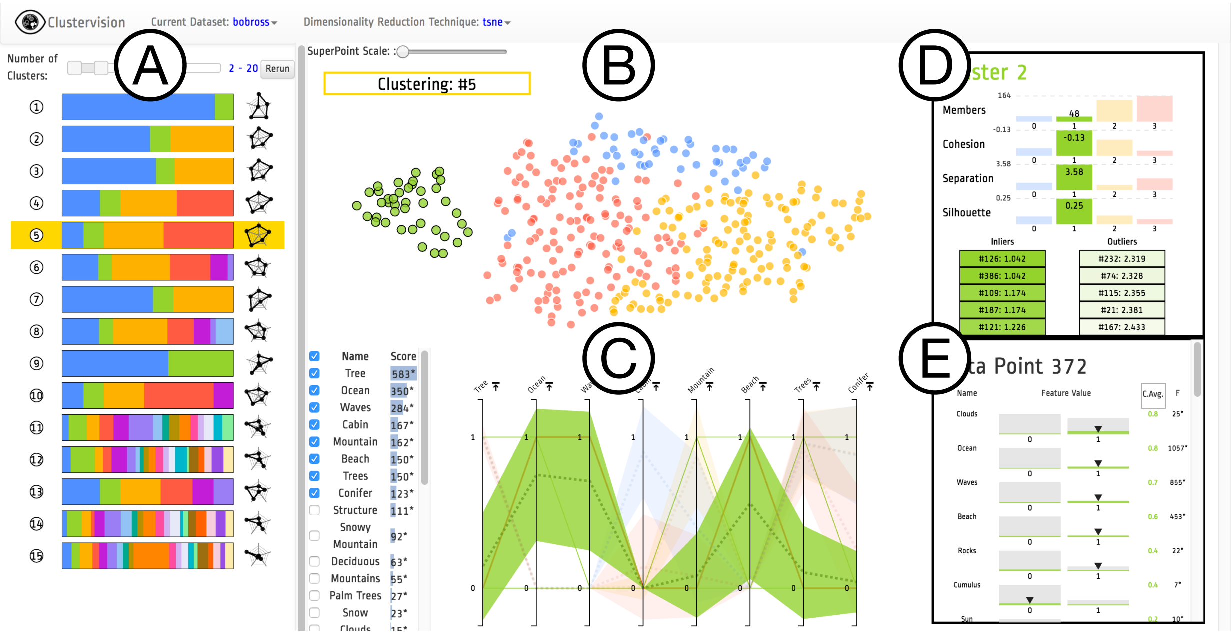
Clustervision: Visual Supervision of Unsupervised Clustering

Bum Chul Kwon, Ben Eysenbach, Janu Verma,
Kenney Ng, Christopher deFilippi, Walter F. Stewart, and Adam Perer



Current Dataset: bobross

Dimensionality Reduction Technique: tsne

Number of Clusters: 2 - 20

SuperPoint Scale:

Clustering: #5

	Name	Score
☑	Tree	583*
☑	Ocean	350*
☑	Waves	284*
☑	Cabin	167*
☑	Mountain	162*
☑	Beach	150*
☑	Trees	150*
☑	Conifer	123*
☐	Structure	111*
☐	Snowy Mountain	92*
☐	Deciduous	63*
☐	Mountains	55*
☐	Palm Trees	27*
☐	Snow	23*
☐	Clouds	15*

ster 2

	0	1	2	3
Members (164)		48		
Cohesion (-0.13)		-0.13		
Separation (3.58)		3.58		
Silhouette (0.25)		0.25		

Inliers	Outliers
#126: 1.042	#232: 2.319
#386: 1.042	#74: 2.328
#109: 1.174	#115: 2.355
#187: 1.174	#21: 2.381
#121: 1.226	#167: 2.433

ta Point 372

Name	Feature Value	C.Avg.	F
Clouds	0 1	0.8	25*
Ocean	0 1	0.8	1057*
Waves	0 1	0.7	855*
Beach	0 1	0.6	453*
Rocks	0 1	0.4	22*
Cumulus	0 1	0.4	7*
Sun		0.2	10*

Fig. 1. An overview of *Clustervision* on a dataset describing 400 paintings by the "Joy of Painting" artist Bob Ross. (A) *Ranked List of Clustering Results* shows 15 different clustering results that are sorted by the aggregated quality measures; (B) *Projection* shows a selected clustering result (highlighted in yellow in (A)) on a projection of data points colored according to corresponding clusters; (C) *Parallel Trends* show the trends of feature values of data points within corresponding clusters in areas across parallel coordinates. Cluster 1 (Green Color) is highlighted; (D) *Cluster Detail* shows quality measures of a selected individual cluster (Cluster 1); (E) *Data Point* shows the feature value distribution of the selected cluster as well as the selected data point (Data Point 372 within Cluster 2).

Abstract—Clustering, the process of grouping together similar items into distinct partitions, is a common type of unsupervised machine learning that can be useful for summarizing and aggregating complex multi-dimensional data. However, data can be clustered in many ways, and there exist a large body of algorithms designed to reveal different patterns. While having access to a wide variety of algorithms is helpful, in practice, it is quite difficult for data scientists to choose and parameterize algorithms to get the clustering results relevant for their dataset and analytical tasks. To alleviate this problem, we built *Clustervision*, a visual analytics tool that helps ensure data scientists find the right clustering among the large amount of techniques and parameters available. Our system clusters data using a variety of clustering techniques and parameters and then ranks clustering results utilizing five quality metrics. In addition, users can guide the system to produce more relevant results by providing task-relevant constraints on the data. Our visual user interface allows users to find high quality clustering results, explore the clusters using several coordinated visualization techniques, and select the cluster result that best suits their task. We demonstrate this novel approach using a case study with a team of researchers in the medical domain and showcase that our system empowers users to choose an effective representation of their complex data.

Index Terms—Unsupervised Clustering, Visual Analytics, Quality Metrics, Interactive Visual Clustering

1 Introduction

Clustering algorithms are a common type of unsupervised machine learning that can be useful for summarizing and aggregating complex multi-dimensional data to make it more interpretable. The goal of clustering is to group together similar items into distinct clusters, so items within a single cluster are similar to each other and different from items outside the cluster. Data can be clustered in many ways, and there is a rich history of techniques designed to achieve clustering results. For instance, algorithms like *k*-means attempt to find cluster centers that are representative of regions in the data. Other techniques like agglomerative clustering start by declaring each item its own cluster and then merge similar clusters into a hierarchy. Other advanced techniques include DBSCAN, which attempt to find dense regions of data in the

- *Bum Chul Kwon, Janu Verma, Kenney Ng, and Adam Perer are with IBM T.J. Watson Research Center in Yorktown Heights, NY, USA. E-mail: {bumchul.kwon|jverma|kenney.ng|adam.perer}@us.ibm.com*
- *Ben Eysenbach is with Massachusetts Institute of Technology in Cambridge, MA, USA. E-mail: bce@mit.edu*
- *Christopher deFilippi is with Inova Heart and Vascular Institute in Fairfax, VA, USA. E-mail: christopher.defilippi@inova.org*
- *Walter F. Stewart is with Sutter Health Research in Walnut Creek, California, USA. E-mail: stewarwf@sutterhealth.org*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

feature space, or Spectral Clustering which reduces data to a low-dimensional embedding and then clusters data. However, given a particular dataset and analytical task, there are no systematic procedures for knowing which algorithm will provide the best cluster. Among the wide variety of algorithms and parameters, how do you choose which to use?

Clustering is often an exploratory problem. Even if one has enough CPUs to try all clustering techniques and parameters, it would still be unclear which results to show users. Furthermore, looking at the same dataset, different users might want to learn different aspects of datasets. For example, when clustering electronic health records, cardiologists might want to cluster patients by their cardiovascular symptoms, and coaches might want to cluster patients by features relevant to their skills of their sport. We need an interactive system for clustering to help users gain new insights into datasets with confidence.

Therefore, we propose *Clustervision*, a visual analytics system that meets this criteria by computing all reasonable clusterings for users, but instead of presenting all options, it provides high quality and diverse clusterings. Quality is determined by evaluating the clustering results using a variety of quality scoring metrics, which emphasize different aspects of good clusters; we combine these metrics to provide diverse recommendations. However, the goal is not to simply show users the clustering result with the highest score according to some metric, but rather to provide insight into the data and to provoke new questions. Users can guide the system to produce more desirable results by expressing constraints on the data relevant to their analytical tasks.

We also provide a case study that demonstrate the effectiveness of *Clustervision* with a team of data scientists, clinicians, and clinical researchers on a longitudinal database of electronic medical records. The research team is interested in finding clusters of similar patients to extract meaningful groups of patients with heart failure. The analysis described in the case study illustrates how the design of *Clustervision* forced scientists to think about their data in new ways and ask new questions about it.

Concretely, our contributions include:

- A design and implementation of an interactive visual analytics system, *Clustervision*, for exploring relevant unsupervised clustering results. Our tool includes:
  - a clustering back-end that runs a variety of clustering techniques and parameters, and provides rankings of high quality results from a diverse set of quality metrics.
  - a visual user interface that allows users to select recommended clustering results, explore the clusters using a variety of visualization techniques, and select the cluster result that best suits their analysis.
- A case study of data scientists using *Clustervision* to find clusters of patients with heart failure from electronic health records.

2 Related Work

This section reviews prior studies that propose various clustering techniques and visual interactive clustering methods.

2.1 Clustering Techniques and Approaches

There exist a large variety of algorithms for clustering [46], and many of these algorithms can be classified into the following five categories:

- **Centroid-based methods:** e.g., *k*-means, Fuzzy c-mean [2]. These algorithms require a priori knowledge of number of clusters, and a choice of metric.
- **Connectivity-based methods:** e.g., Hierarchical and Agglomerative methods [2]. These algorithms use a linkage criterion and distance metric to split or join clusters.
- **Density based methods:** e.g., DBSCAN [18], OPTICS [3]. These algorithms require parameters to quantify the density of the clusters and how to partition density.
- **Low Dimensional Embeddings:** e.g., Spectral Clustering [41]. These algorithms require a specific number of low dimensions to be projected on, and number of clusters.
- **Probabilistic clustering methods:** e.g., Gaussian Mixture Models [36], Latent Dirichlet Allocation [9]. These algorithms use probability distributions to determine which cluster points belong and which hyperparameters to use.

Each of these classes of algorithms have somewhat different strengths [32]. For example, centroid-based methods support a representation of clusters using the cluster means. Density-based methods support the detection of outliers that are not assigned to any cluster. Connectivity-based methods provide a hierarchical representation of possible groupings which can be inspected with dendrograms. Spectral clustering is particularly useful when the clusters are not completely described by their centroids. Probabilistic clustering methods may represent the data more faithfully by using decisions from the model, but are often less interpretable to users. Complementary to these approaches, interactive clustering, where users provide feedback to the algorithm, is also an active area of research [4, 5, 7].

As there are many clustering algorithms and user constraints, the optimal choice often depends on the dataset and task. The difficulty in choosing appropriate values for the parameters also makes it difficult to optimally utilize a clustering method. In order to assess the quality of a clustering, many quality metrics have been proposed, including:

- **Calinski-Harabaz index:** The Calinski-Harabaz index of a clustering is defined as the ratio of the between-cluster variance and the within-cluster variance [21]. Well-defined clusters have higher between-cluster variance and lower within-cluster variance.
- **Silhouette Coefficient:** The Silhouette Coefficient [37] is a measure of how similar a point is to its own cluster compared to other clusters, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.
- **Davies-Bouldin index:** This metric is similar to the Calinski-Harabaz index and is defined as the average over all clusters the ratio of within-cluster dispersion and the pairwise between-cluster dispersion [16].
- **Gap Statistic:** The Gap Statistic [44] measures the quality by considering clusterings of random permutations of the data and comparing these to a null reference distribution with no clustering structure.
- **$\mathbf{S_{Dbw}}$:** The $\mathbf{S_{Dbw}}$ Validity Index [20] attempts to measure quality by taking into consideration cluster compactness, separation, and the density of the clusters.

The effectiveness of these metrics in gauging the quality of the clustering is also difficult to determine due to the lack of ground truth. To understand clustering metrics, Liu et al [25] studied 11 quality metrics and investigated their validation properties in five different aspects: monotonicity, noise, density, subclusters and skewed distributions.

As there is no systematic approach for finding the best clustering result, an alternative is to summarize results from multiple clustering runs. For instance, Gionis et al. [19] proposes *clustering aggregation*, which aims to find a clustering that agrees with other clusterings by running different algorithms and different parameter values. A related approach in clustering community is called *meta clustering*, where many different clusterings of the data is performed and then users can choose the clusterings based on their requirements. This problem was formulated by Caruana et al. [14] where they proposed methods to generate diverse clusterings of the data and then (meta-)cluster this set of data clusterings. Phillips et al. [34] proposed a framework to generate diverse, high-quality clusterings by sampling high-quality clusterings and choosing *k* representatives. *Subspace clustering* aims to find clusters in different subspaces of datasets by integrating feature relevance evaluation and clustering. There are many algorithms to find such optimal partitions of data by identifying relevant dimensional subspaces [33]. Similarly, *consensus clustering* attempts to provide consensus between multiple runs of clusterings, which can be outputs of different parameters or different clustering techniques, to determine the number and assess the stability of the groupings [31].

While clustering summarization approaches are promising, the results may be hard to interpret. *Clustervision* builds on these approaches by using diverse metrics to measure quality, supporting user interaction, and making results more interpretable with visualization to help guide users towards an appropriate clustering result.

2.2 Visualization Systems for Cluster Analysis

There is a rich history of visual analytics systems that employ clustering as a part of high dimensional data analysis. Hierarchical Clustering Explorer [39] allows users to investigate an overview of a clustering result and to inspect and compare details of clusters by using coordinated displays. VISTA [15] enables users to visually view clusters of a clustering result on a 2D projection, then to re-label data points, and verify user-adjusted results using internal quality metric scores like RMSSTD (Root Mean Square Standard Deviation), RS (R-Square) and S_{Dbw}. Dicon [13] visualizes multidimensional clusters' quality as well as attribute-based information through icon-based visualization and embedded statistical information. Unlike *Clustervision*, these systems do not support comparison between multiple clustering results.

Some applications allow users to provide feedback on clustering results so that the next run applies their inputs. desJardins et al. [17] proposed a technique to iteratively run and visualize clustering with constraints made by users. User input is made by moving objects initially displayed using spring-based embedding on a 2D projection. iVisClustering allows users to adjust cluster hierarchies and to re-label individual data items (i.e., documents) into another cluster [23]. Cluster Sculptor also allows users to update cluster labels on a 2D projection while iterating t-SNE steps [12]. Boudjeloud-Assala et al. propose an interactive visual clustering system that allows users to define seeds (i.e., center) and limits of clusters for steering the clustering process [11]. Clusterix is a system that allows users to add or remove features for future clustering runs [28]. While these systems help steer the user toward better clustering results, the user must define how to make the clustering better rather than receiving recommendations from the system, unlike *Clustervision*.

On the other hand, some visual analytics techniques allow users to generate and compare multiple clustering results with respect to their quality, as well as attribute-based information. Turkay et al. [45] propose a visual analytic framework that users can form clustering by automated algorithms or manual formation and evaluate them visually by using cluster tendency scores as well as a parallel cluster view. XCluSim allows users to interactively generate and compare multiple clustering results with multiple coordinated views [26]. In these systems, views and computational techniques are combined to help users interactively reach a stable or satisfying clustering result. However, no single quality metric can guarantee users' diverse analysis goals and requirements. Even with multiple quality metrics, users may want to explore more diverse sets of clustering results and drill down into interesting results.

Clustervision differentiates itself from the aforementioned work by contributing a comprehensive visual analytics system that lets users rank and compare multiple clustering results based on quality metrics, provides meaningful feature-based summaries of clusters using visualizations and univariate statistics, and allows users to apply their domain expertise to constrain and steer clustering analysis.

3 Design Goals

The initial design goals of *Clustervision* were derived from prior work and refined with iterative development of prototypes and interviews with data scientists. In addition, we were inspired by the Visual Parameter Space Analysis (vPSA) conceptual framework proposed by Sedlmair et al. [38]. Using the terminology of vPSA, *Clustervision*'s data flow utilizes a *sampling* data flow by systematically sampling multiple clustering algorithms and parameters to generate a variety of possible clustering results. Users can browse the clustering results using a *global-to-local* navigation strategy by beginning with an overview of the highest quality results. *Clustervision* was also designed to support various analysis tasks, including *optimization* to find a satisfying clustering result guided by quality metrics, *partitioning* to show the diverse clustering results possible due to different parameters, and *sensitivity* by allowing users to constrain parameters to find relevant clusterings. With these tasks in mind, our concrete list of design goals include:

1. **Compare clustering results w.r.t. quality, technique, parameter:** *Clustervision* should allow users to compare clustering results with respect to their quality, clusters, clustering technique, and parameters. Using *Clustervision*, users should be able to visually interpret the results to assess their relevance to reaching insights.
2. **Compare clusters within a clustering result w.r.t. features, quality, point:** *Clustervision* should allow users to pick a clustering result and to visually explore clusters with respect to their features, quality, and data points within clusters, to ensure the clusters represent their data faithfully.
3. **Compare a data point to its cluster:** *Clustervision* should allow users to see details of data points and to assess their similarities and differences from the cluster with respect to the data attributes.
4. **Understand why data points are clustered together or apart:** *Clustervision* should allow users to help understand what features of the dataset are responsible for the grouping of the data points.
5. **Retrieve new clustering results recommended by *Clustervision* based on users' input:** Using *Clustervision*, users should be able to steer clustering results towards their analysis goals. For instance, *Clustervision* should enable users to find the size and type of clusters they are seeking, as well as specifying constraints for clustering while users analyze data.

Clustervision targets multi-dimensional data composed of up to a hundred semantically meaningful features, as this is a common upper bound for most visual parameter space analysis tools surveyed [38]. Furthermore, the survey illustrates the novelty of our design, as no other tools are described as primarily supporting a *sampling* data flow with *global-to-local* navigation, and *optimization*, *partitioning*, *sensitivity* analysis tasks like *Clustervision*.

4 System

In order to support interactive exploration of clustering results, we propose *Clustervision*, a web-based interactive visual analytics system. Although the tool's design was motivated by challenges with clinical data, the tool is also able to cluster multi-dimensional data from any domain. For example, the tool has also been used to explore clusters of handwritten digits, university rankings, as well as classic data sets from the UCI Machine Learning Repository [24].

4.1 Running Example: The Joy of Clustering

We demonstrate the workflow and system features by using a dataset of all of the 403 paintings produced on the PBS show "The Joy of Painting". This television show was hosted by Bob Ross, famous for painting "happy trees" and "fluffy clouds" and each episode resulted in the completion of new work of art. Over the course of the 403 episodes of the show, a variety of diverse landscapes were painted featuring trees, oceans, mountains or man-made structures. Walt Hickey, the chief culture writer for the website FiveThirtyEight, recently conducted a statistical analysis of the work of Bob Ross[1] and manually coded each of the episodes using 67 features (e.g., trees, water, mountains, and weather elements). Hickey was interested in finding clusters of similar paintings for his featured article, but chose to use a single clustering technique (k-means) and a single parameter (k=10). Hickey remarked that while some of the clusters "were the kinds of clear clusterings we were hoping to find", others were "groupings are not supremely helpful in defining what Ross painted". We use this dissatisfaction by Hickey to motivate our discussion of how *Clustervision* could potentially be used to arrive at more satisfactory clusterings.

4.2 Workflow

In order to support the workflow of data scientists, the UI of *Clustervision* is organized in the following ways. Figure 1(a) shows the *Ranked List of Clustering Results* on the left, which lets users compare multiple clustering results. In the middle, the *Projection* (Figure 1(b)), *Ranked Features* and *Parallel Trends* (Figure 1(c)) views help users compare clusters within a clustering result using multiple high-dimensional visualizations. The *Cluster Detail* (Figure 1(d)) and and *Data Point* (Figure 1(e)) help users understand and compare data points and their clusters.

[1] https://fivethirtyeight.com/features/a-statistical-analysis-of-the-work-of-bob-ross/

Furthermore, *Ranked Features* and *Data Point* views also provide details on why data points are clustered together and apart. Users can use each of these views to pivot to clustering results more relevant to them, by supporting searching for clusterings that meet their clustering constraints.

The following sections shows the design and function of each view.

4.2.1 *Ranked List of Clustering Results*

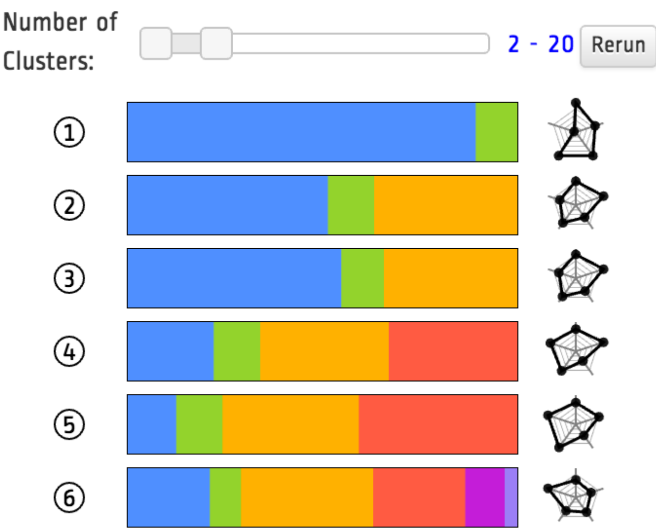


Fig. 2. The *Ranked List of Clustering Results* view features a ranked list of clustering results. Each row features a clustering summary glyph, where each colored stripe represents a color whose width is proportional to the number of data points in that cluster. Each cluster has a unique color that is consistently used across all views in the UI. On the right is a radar chart, where each quality metric value is visualized along an axis and all the five quality metrics are connected to form a polygon. When *Clustering Comparison*, multiple polygons are overlaid for visual comparison.

After a dataset is loaded into the tool, *Clustervision* computes and evaluates all possible combinations of clustering techniques and parameters. These calculations are offloaded to our cluster analytics server, which is multi-threaded and can farm out these calculations to multiple cores. Using a default configuration, *Clustervision* will use three clustering techniques (k-means, Spectral Clustering, and Agglomerative Clustering) and 19 parameter configurations (k=2-20), resulting in 58 clustering results. The system can also optionally include more clustering techniques and parameters, including DBSCAN [18] and Gaussian Mixture Models, but this optional configuration is not used for describing the system in this paper.

All of the clustering results are then analyzed using 5 quality metrics: Calinski-Harabaz, Silhouette, Davies-Bouldin, S_{Dbw}, and Gap Statistic. As each of these quality metrics aim to compute quality using different properties of the clusters (e.g., variance, within-cluster distance, between-cluster distance, density), we chose not to rely on a single metric but instead a variety of diverse metrics. Furthermore, although S_{Dbw} [25] was shown by Liu et al. to perform best on synthetic data, there is still open debate on the most effective quality metrics, so our system favors a consensus approach. By default, the top 3 highest ranking results from each metric are presented to the user, resulting in the top 15 results in total for the user to consider. In order to ensure the results aren't too similar, an item will only be considered as a top result if it is at least 5% different from another top result (that is, less than 95% of the data points should belong to different clusters for the result to be considered distinct).

These results are presented in the *Ranked List of Clustering Results* view as a ranked list of clustering results. Figure 2 shows an example of the top 6 clustering results. Each row is a clustering result, which has a numeric ranked index (e.g., 1-6), a clustering summary glyph, and a quality summary radar chart. The clustering summary glyph looks visually similar to a set of horizontal colored stripes, where each colored stripe represents a color whose width is proportional to the number of data points (e.g., paintings) in that cluster. Each cluster has a unique color that is consistently used across all views in the UI using a repeating 20-color palette. As the total number of data points is consistent across all clustering results, users can quickly check and compare the number of clusters and the distribution of data points across clusters using the view. To minimize the number of points that change color when the user switches from one clustering result to another, color assignment is formulated as a minimum cost perfect matching problem, using the Hungarian algorithm [22] to keep colors consistent for similar clusters across clustering results.

On the right, the radar chart consists of a sequence of five spokes, with each spoke representing one of the quality metrics. The length of each spoke from the center is proportional to the normalized score of the quality metric, and a line is drawn to connect the quality metrics as a polygon. Moving the mouse over a spoke will reveal the name of the quality metric responsible for the score.

Ranked List of Clustering Results allows users to interactively redefine and request clustering results that they want to view. Users can move the mouse over each clustering summary glyph, which displays additional information about the clustering result, including the number of data points, the clustering algorithm, quality metric scores, and the quality metric that was responsible for this particular clustering result to appear in the top results.

User can also adjust a range slider to focus on clustering sizes relevant to their analysis. For example, if users wished to summarize Bob Ross's painting in a small number of groups, they could select smaller ranges on the slider and focus their attention on high quality results with less clusters of paintings.

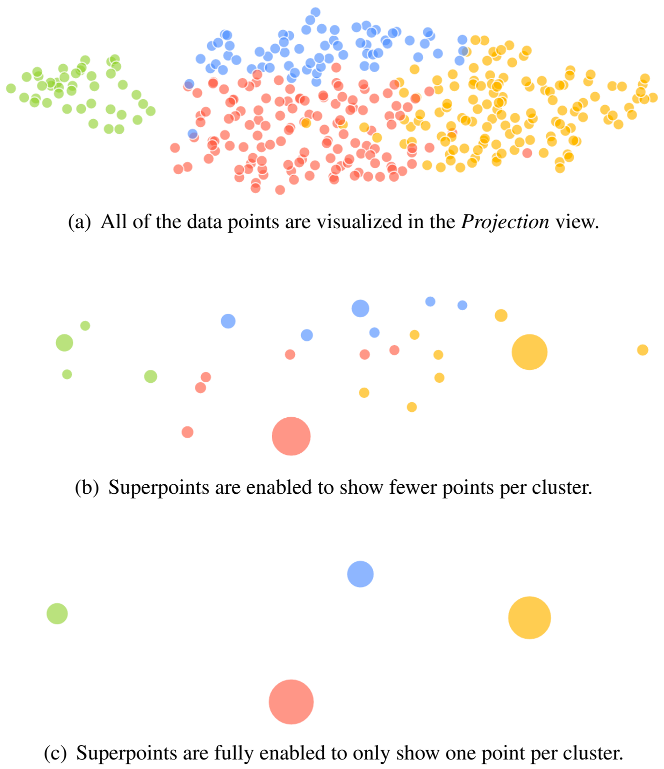
Figure 2 shows the top clustering results for the Bob Ross dataset, and it illustrates the variety of results. Some clusterings have as few as two clusters of paintings, like Clustering 1, which has a large blue cluster and a small green cluster. Other clusterings, like Clustering 6, have six clusters of diverse sizes. None of these highly ranked clusterings match Hickey's chosen technique of k-means, with k=10, suggesting it may be worthwhile to explore other options.

4.2.2 *Projection*

In order to understand if a particular clustering result is relevant to the analytical task, users often need to see their data points in context of the cluster groupings. The *Projection* view encodes data points as circular elements in a two dimensional space, resembling a scatterplot, as shown in Figure 3(a). However, instead of plotting the data on only two dimensions of the data, *Clustervision* uses dimensionality reduction techniques to synthesize all of the dimensions of the data into two dimensions. Unlike scatterplots, this results in axes that do not have a clear meaning, so it may provide difficulties for inexperienced users in interpreting the meaning of data point positions and the axes of the projection [30]. Nonetheless, this technique was chosen as it is applicable to any high-dimensional data from any domain, and can provide a consistent way to represent this data. The main use of the *Projection* view is to have a consistent and stable representation across all clustering results, as the positions of the data points remain stable across all clustering results. Though the position of the data points gives clues to the distance and separation between clusters, users can find more evidence about the underlying properties of the clusters from the other views.

When users select a clustering result from the *Ranked List of Clustering Results* view, the data points in the *Projection* view are colored to match its cluster. By default, *Clustervision* projects data using the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [27] which is currently a popular method for exploring high-dimensional data. However, if users are unhappy with the projection, they can pivot to other popular techniques including principal component analysis (PCA), spectral projection, and multidimensional scaling (MDS) by choosing an alternative type in the title bar.

The *Projection* view serves as one way to explore both individual



(a) All of the data points are visualized in the *Projection* view.

(b) Superpoints are enabled to show fewer points per cluster.

(c) Superpoints are fully enabled to only show one point per cluster.

Fig. 3. The *Projection* view encodes data points as circular elements in a two dimensional space using dimensionality reduction techniques to synthesize all of the dimensions of the data into two dimensions. The *Projection* view provides consistent and stable representation across all clustering results. Users can enable superpoints to reduce the visual clutter with full control over the number of superpoints.

data points and clusters. Most importantly, it allows users to use other views to get more details about the selected data points and clusters. Users can move the mouse over an individual data point to see details on demand in the *Data Point* view, or select a cluster for analysis in the *Cluster Detail* view. User can also view the feature values of the data points in different clusters in the *Parallel Trends* view.

Figure 3(a) shows a t-SNE projection of the 403 paintings in the Bob Ross dataset. The projection view shows an island of green points on the left, whereas a bigger island of data points on the right is divided into three clusters (red, blue, and yellow). This suggests that the green paintings may be very distinct paintings from others in the collection. In order to investigate this, the paintings and cluster can be selected to get more information in the *Data Point* and *Cluster Detail* views.

As users may feel overwhelmed by having all data points visible, users can reduce the visual clutter by using the *Superpoints* option in the *Projection* view, which adopts the idea of coresets [1] and Splatterplots [29]. In this mode, similar points are represented by a superpoint, which is a representative of their neighbors. These are visually encoded to be larger and proportional to the number of neighbors they represent, whereas the neighboring points that have a represented are removed from the view. *Superpoints* are computed using hierarchical clustering on each cluster, where representatives are found by finding the point which is closest to all other points in the same cluster. Users can control the number of superpoints using a slider, as shown in Figure 3.

4.2.3 Feature-based views: *Ranked Features* and *Parallel Trends*

In order to help summarize the clustering results, the *Ranked Features* and *Parallel Trends* views are coordinated with the projection view and shows information about the features of the selected clustering result.

One of the challenges associated with unsupervised clustering is that even after clusters are defined by a technique, it is difficult to summarize why the cluster groupings were made. In an attempt to retrieve the features responsible for the separation, we utilize univariate statistics to compute whether there is a statistically significant relationship between each feature and each cluster. We consider this a classification task, where each cluster is a class, and compute the analysis of variance (ANOVA) for the each feature.

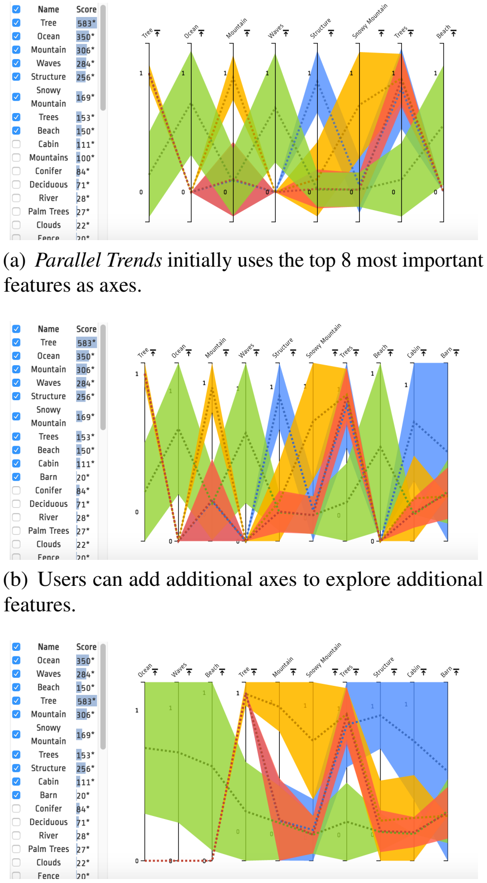
The resulting scores, based on the ANOVA F-Value, allow us to rank each feature in order of importance, as well as retrieve an associated p-value to ensure the relationship is statistically significant. This approach is similar to using such univariate statistics for feature selection for determining the most informative features, but instead of using it to remove non-informative features from a model, we use the resulting scores to rank the importance of features. These important features are displayed as a ranked list in the *Ranked Features* view, where each feature name is augmented with a numeric importance score and a corresponding bar chart, as shown in Figure 4.

While this test is univariate and only considers each feature separately, this nonetheless provides clues to users which features may be most responsible for the separation amongst clusters. An additional caveat is that features selected by an F-Value only indicates that the feature is important among some of the clusters, but may not be important for all clusters. While post-hoc tests could be used to decide which clusters the feature is responsible for, choosing a proper post-hoc test depends on the variances of features across clusters. Instead, we opted to pair these importance scores with the *Parallel Trends* view to visualize the trends of each cluster across these important features.

The *Parallel Trends* is similar to parallel coordinates, but in order to simplify the complexity of many lines, initially the view only shows the trends of each cluster. As in parallel coordinates, *Parallel Trends* has vertical axes that represents each feature of the data points. However, instead of drawing a line crossing the axes for each data point as in parallel coordinates, *Parallel Trends* draws an area path per cluster. The intervals cross each axis, where the vertical ends represent standard deviation or 95% confidence intervals for the corresponding features. Then, a dotted line is drawn on top of the area path per cluster to show the mean values for each cluster for the corresponding data feature. To see details of a cluster, users can click on an area path to show individual lines that represent corresponding data points within the cluster as shown in Figure 4. This implementation also allows users to sort axes, switch axes, and filter on specific feature values on each axis, which are interaction techniques common to parallel coordinates.

For example, in the selected Bob Ross clustering shown in Figure 4(a), the top features most responsible for the cluster grouping are the presence of trees, mountains, and oceans in paintings, which is consistent with the features that Hickey manually used to summarize his meaningful clusters (e.g., clusters of “ocean scenes”, “trees and at least one mountain”, and “trees but no mountains”). This ranked list in conjunction with the *Parallel Trends* views help show how these features correlate with the clusters. The Green cluster has uniquely high values in Ocean, Waves, and Beach, giving a clear indication that this cluster represents the ocean-oriented paintings of Ross. This cluster is demonstrably different from the Yellow cluster (which has high values of tree, mountain, snowy mountains, and trees), the Blue cluster (with Structures), and the Red cluster (with tree and trees). While only the top 8 features are shown, other features can be added by selecting them. For example, after analyzing this cluster in the *Data Point* view, it became clear that many of the paintings in the Blue cluster appear to have cabins and barns. By adding these features to the *Parallel Trends* view, it is clear how the red cluster dominates these features (Figure 4(b)). This becomes even more evident when sorting the axes by their relevance to the cluster (Figure 4(c)).

Like parallel coordinates, *Parallel Trends* may suffer from scalability issues if there are many features. For example, if there are many features and thus many axes, it may be difficult to view the trends due to limited screen real estate. To resolve this issue, the *Parallel Trends* view is coordinated with the *Ranked Features* table, and only the most important features are shown initially. Users can add new features as new axes to the *Parallel Trends* by selecting the checkbox, as shown in Figure 4. Users can also remove features by unselecting them.



(a) *Parallel Trends* initially uses the top 8 most important features as axes.

(b) Users can add additional axes to explore additional features.

(c) Users can also re-order axes to make trends of clusters more clear.

Fig. 4. The *Parallel Trends* view is similar to parallel coordinates, but in order to simplify the complexity of many lines, the view focuses on showing the trends of each cluster. *Parallel Trends* has vertical axes that represents each feature of the data points. However, instead of drawing a line crossing the axes for each data point as in parallel coordinates, *Parallel Trends* draws an area path per cluster. The intervals cross each axis, where the vertical ends represent standard deviation or 95% confidence intervals for the corresponding features.

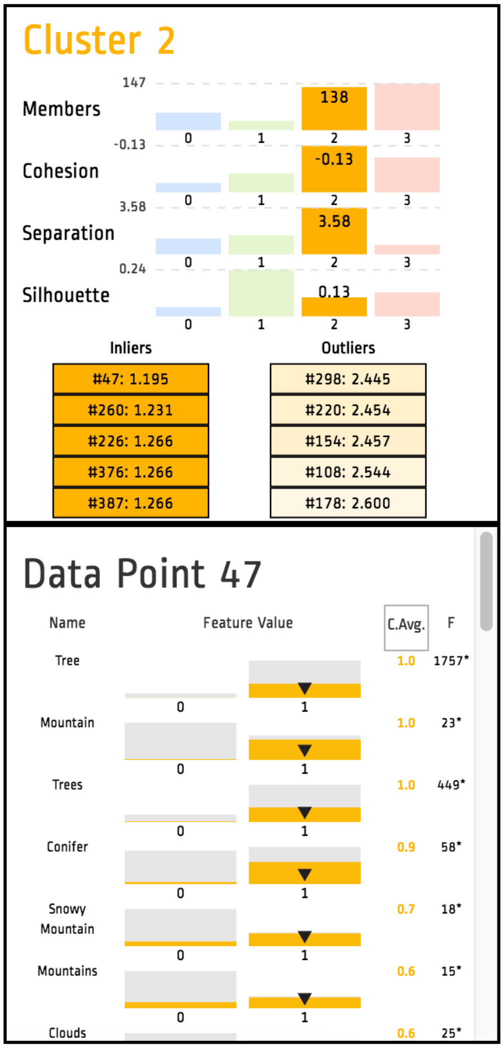


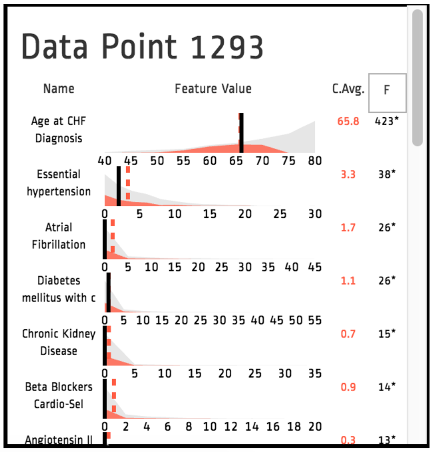
Fig. 5. The *Cluster Detail* view (top) presents a summary of the clusters using statistics and prototypes. The *Data Point* view (bottom) provides details about the actual values of a data points and provides context by presenting the distribution of values alongside each value.

4.2.4 *Cluster Detail* and *Data Point*

The *Cluster Detail* view appears when users select a particular cluster from the *Projection* or *Parallel Trends* views. This view is designed to present a summary of the clusters using statistics and prototypes. For the selected cluster, the number of data points that are members of the cluster is shown as a labeled bar that is the same color of the cluster. This number is put in context with all of the other cluster sizes by showing translucent bars representing each cluster to form a bar chart. Similar bar charts are shown for statistics summarizing the cluster, such as cohesion, separation, and silhouette scores, as shown at the top of Figure 5. Cohesion measures how closely related are the data points in a cluster, defined as the intra-cluster sum of squares [2]. Separation quantifies how distinct a cluster is from other clusters, and is defined as the inter-cluster sum of squares [2]. Silhouette is the mean of all of silhouette scores for the cluster (defined above). In addition to these statistical summaries, the *Cluster Detail* view also shows members of the cluster that are typical or atypical for the cluster based on the distance metric. On the left, the top 5 "inliers" are shown, which are the five data points closest to the center of the cluster. On the right, the top 5 "outliers" are shown, which are the data points farthest from the cluster's center. The description of inliers and outliers show the euclidean distance of the corresponding data points from the centroid of the cluster. By clicking on any of these points, the data point will highlight in the *Projection* and *Parallel Trends* views, and also show more details in the *Data Point* view.

The *Data Point* view appears when users select or mouseover a data point in the *Projection* or *Parallel Trends* views. The *Data Point* provides details about the actual values of a data points features. However, this view also puts them in the context of other data points by presenting the distribution of values alongside each value. The value distributions are shown using a kernel density plot, which has been shown to be an effective visual technique for communicating how a cluster relates to the whole dataset [43]. In order to demonstrate continuous values, not present in the Bob Ross dataset, Figure 6 shows a data point selected from a red cluster in a medical dataset described in Section 5. Each density plot shows the data point's cluster distribution (area in red) as well as the distribution of all datapoints (area in gray). Vertical marks represent the mean values of the chosen cluster (striped vertical mark in red) and the currently selected data point (black) for continuous feature values. For binary variables and categorical feature values with less than five levels, such as the data in the Bob Ross dataset, a histogram is shown rather than density plot, with triangle marks to show the selected data point as seen in Figure 5. User can quickly observe the attribute values of each cluster compared with the attribute values of all.

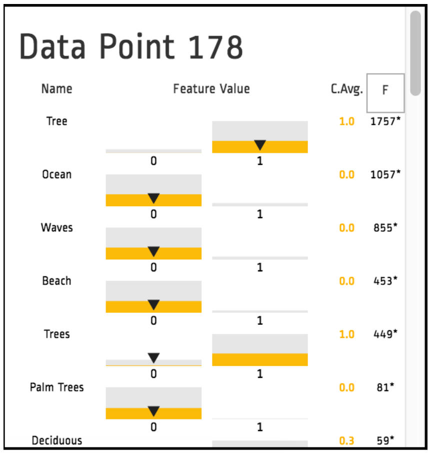
Users can sort features by their name, value, cluster average value, and importance. The importance calculation is similar to the technique described above in the *Ranked Features* view. However, here the technique considers assigning the selected cluster as one class, and



Data Point 1293

Name	C.Avg.	F
Age at CHF Diagnosis	65.8	423*
Essential hypertension	3.3	38*
Atrial Fibrillation	1.7	26*
Diabetes mellitus with c	1.1	26*
Chronic Kidney Disease	0.7	15*
Beta Blockers Cardio-Sel	0.9	14*
Angiotensin II	0.3	13*

Fig. 6. Kernel density plots are used in the *Data Point* view when features have continuous values. This view illustrates data from the Sutter Health case study, described later in Section 5.



Data Point 178

Name	C.Avg.	F
Tree	1.0	1757*
Ocean	0.0	1057*
Waves	0.0	855*
Beach	0.0	453*
Trees	1.0	449*
Palm Trees	0.0	81*
Deciduous	0.3	59*

Fig. 7. This figure illustrates the same *Data Point* view as Figure 5, but instead sorted by feature importance score, abbreviated as F, to reveal features that are both common and missing from this cluster.

all other clusters as a second class. By computing an ANOVA using these cluster-centric classes, it is possible to determine which features are responsible for why the selected cluster is different from all other clusters. This option presents the most important features at the top of this view, making it easy to compare between data points and clusters by mouse-overing regions of the interest in the *Projection* view.

In Figure 5, the yellow cluster is selected. While having less data points than the green and red clusters, it nonetheless features the highest cohesion and separation scores. The top "inlier" was selected in Figure 5, which shows that this representative painting has tree, mountain, trees, conifer, and snowy mountains. Sorting by feature score (Figure 7), this panel illustrates how this clustering is also defined by the absence of oceans, waves, beach and palm tree elements.

4.2.5 Clustering Constraints

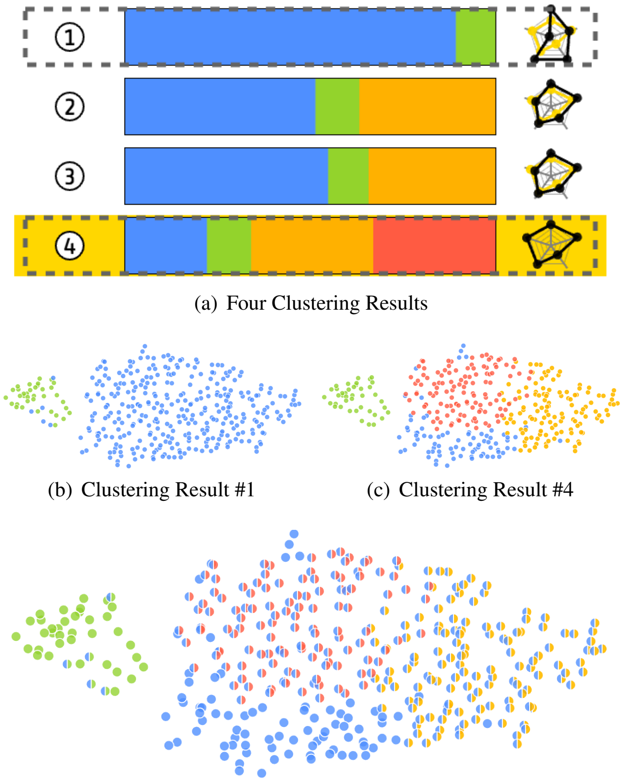
Users can also interactively request new results by setting up constraints with respect to specific data points. Constrained clustering is to filter the nearest clustering result that satisfies users' constraints (e.g., 'must-link' for a set of data points to be grouped together within a cluster and 'cannot-link' for a set to be in separate clusters) [6]. Users can select multiple data points and tell the system that they need to be either in the same cluster or in separate clusters. Then, the system filters clustering results based on the requirements set by the user. The user can create constraints by right-clicking on data points to prompt a menu and by choosing them to be either same or separate clusters in the new clustering results.

For example, after a deeper exploration the Bob Ross dataset, the *Parallel Trends* view made it clear there were paintings with lakes in the blue, yellow, and red clusters. If one wanted to see if a clustering result exists where these lake paintings might make up their own cluster, users could select a lake painting from each cluster and declare a constraint where they need to be a part of the same cluster. *Clustervision* would then search all clustering results and update the *Ranked List of Clustering Results* with results that match the constraint.

4.2.6 Comparing Clustering Results

After examining multiple clustering results, users may wish to compare them to understand them better. *Clustervision* supports *Clustering Comparison* by allowing users to select multiple clustering results in the *Ranked List of Clustering Results* view for comparison. The *Projection* view shows an overview of differences between clustering results. Instead of each data point having a single color according to its cluster, data points in the *Clustering Comparison* view are represented as a circle divided into multiple slices, with each slice colored by each result selected.

For instance, Figure 8(b) and 8(c) show two clustering results. As shown in Figure 8(d), *Clustering Comparison* highlights data items that were clustered with blue in Figure 8(b) but clustered with red and yellow in Figure 8(c). The projection view highlights such items by dividing a circle into two halves, the left half and the right half: the left half showing the cluster color for a clustering result and the right half showing the cluster color for the other clustering result. When multiple clustering results are selected for comparison, the selected clustering (highlighted in yellow) has its quality metric scores provided for context in each of the radar charts, as shown in 8(a). In this example, Clustering 4 has a stronger S_{Dbw} (upper left spoke) score, whereas Clustering 1 has a stronger Davies-Bouldin index (bottom right spoke).



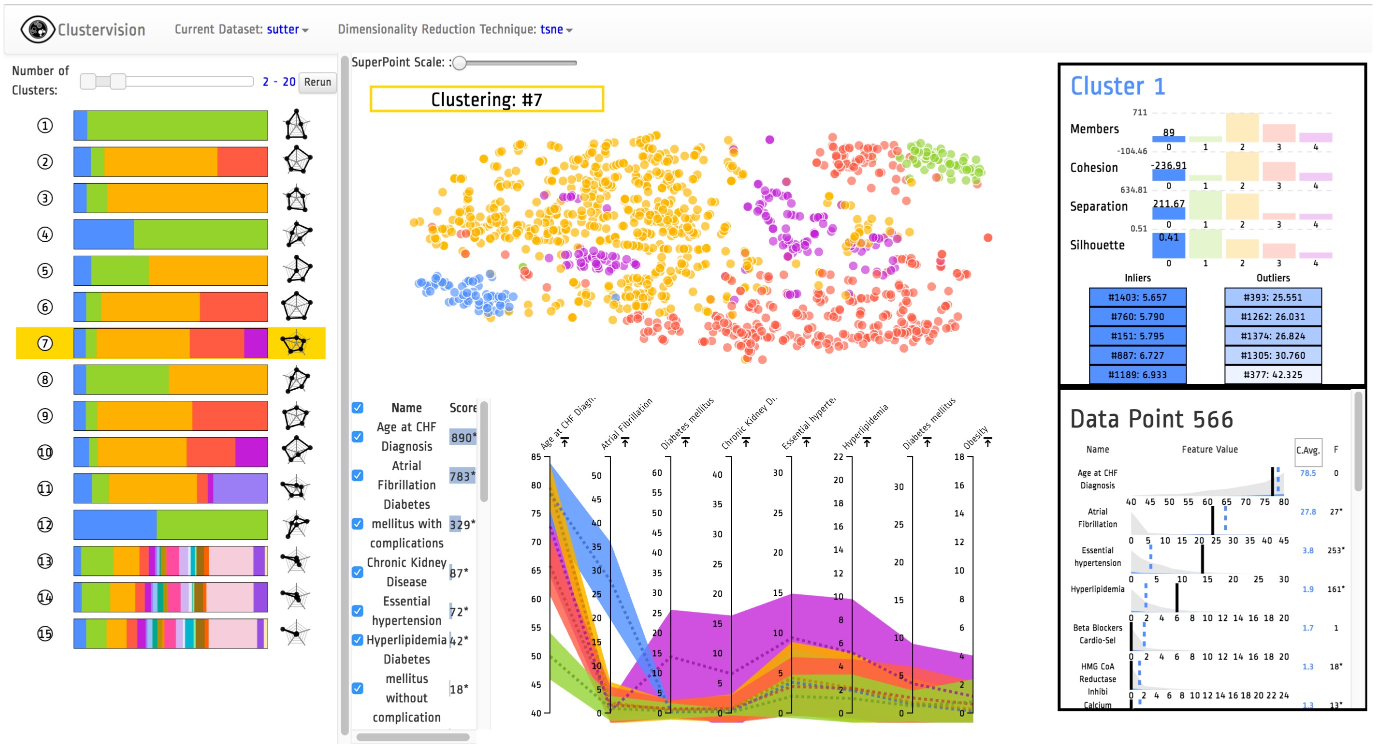
(a) Four Clustering Results

(b) Clustering Result #1

(c) Clustering Result #4

(d) The Difference between Clustering Results #1 and #4

Fig. 8. *Clustering Comparison* shows the difference (d) between two clusterings (b) and (c) in circles with slices of different colors. When multiple clustering results are selected, the selected clustering (highlighted in yellow) has its quality metric scores provided for context in the radar charts, as shown in (a).



Current Dataset: sutter

Dimensionality Reduction Technique: tsne

Number of Clusters: 2 - 20

SuperPoint Scale:

Clustering: #7

Name	Score
Age at CHF Diagnosis	890*
Atrial Fibrillation	783*
Diabetes mellitus with complications	329*
Chronic Kidney Disease	87*
Essential hypertension	72*
Hyperlipidemia	42*
Diabetes mellitus without complication	18*

Cluster 1

Inliers	Outliers
#1403: 5.657	#393: 25.551
#760: 5.790	#1262: 26.031
#151: 5.795	#1374: 26.824
#887: 6.727	#1305: 30.760
#1189: 6.933	#377: 42.325

Data Point 566

Fig. 9. As a part of a case study, clinical researchers were able to use *Clustervision* to identify meaningful clusters of patients with heart disease.

5 Case Study: Finding Clusters of Similar Patients

There is a growing belief in the visualization community that traditional evaluation metrics (e.g., measuring task time completion or number of errors) are often insufficient to evaluate visualization systems [8,35,42]. Instead, we demonstrate the effectiveness of *Clustervision* using a case study as a team of five data scientists and two clinicians interested in using unsupervised clustering techniques on a longitudinal database of electronic medical records. The research team is interested in finding clusters of similar patients to extract meaningful groups of patients with heart failure using a database of approximately 1,500 patients from Sutter Health, a healthcare provider in Northern California.

There are many diseases in which patients may be diagnosed as having the same disease, but will respond to treatments differently. For example, heart failure is often described as a heterogeneous disease, which makes it difficult to find treatments to improve outcomes consistently among patients [10]. Researchers believe that if they could classify patients into groups of similar individuals, they could impact these distinct groups with more predictable, group-specific treatments.

A recent study by Shah et al. managed to use unsupervised clustering techniques to classify patients with HFpEF (a cardiovascular syndrome known as heart failure with preserved ejection fraction) [40]. The researchers imported clinical variables, physical characteristics, laboratory data, and echocardiographic parameters of 397 patients into a hierarchical clustering package in R and tried using varying parameters of k (which defines the number of clusters in the output, which many clustering algorithms require as input). After trying all values of k between 1 and 8, they measured their clusters using a quality metric, Bayesian information criterion, and determined the clustering that resulted in 3 groups received the highest score. We refer to this grouping as the Northwestern clustering result. After examining these groups more closely, they believe that these 3 groups represent 3 archetypes of HFpEF, respectively, which are: (1) a group of younger patients with a lower number of comorbidities, (2) obese patients with diabetes and hypertension, and (3) older patients with atrial fibrillation and chronic kidney disease. This work claims to be the first study that applies unsupervised clustering to resolve heterogeneity among HFpEF patients using observational data.

However, these researchers opted to use a single clustering algorithm with only 8 different parameter configurations and 1 quality metric. While the results the researchers derived appear to be clinically meaningful, there is an open question if any additional insights could have been reached had any other clustering techniques or parameter configurations been explored.

5.1 Goal: Analysis Beyond the Northwestern Results

While this motivating study [40] used data from patients collected after they were diagnosed with HFpEF, our case study team was interested in going beyond this to determine if data from patients before their diagnosis of HFpEF could be used to cluster patients. Furthermore, rather than running a prospective observational study, they were hoping to utilize retrospective data already collected in electronic health records. Identifying meaningful groups of patients with data proceeding the diagnose could make it possible for patients to start early treatments to hopefully prevent the disease from occurring in the first place. The research team utilized a database of patients diagnosed with HFpEF, but only extracted records that occurred during the two years prior to diagnosis. While certain features, such as physical characteristics, laboratory data, and echocardiographic parameters, were not available in the electronic health records, the researchers managed to extract the co-morbidities and medications that were used in the Northwestern clustering to describe the differences between their cohorts. This data was assembled into a table where each row is a patient and each column is a comorbidity or medication. Each cell in the table is a count of the number of times the patient was diagnosed with the comorbidity or medication in the two years leading up to their diagnosis. In total, there were 1474 patients, each with 23 features.

5.2 Gaining an Overview of Diverse Clustering Results

As an initial baseline, the researchers were interested in using unsupervised clustering techniques on this table and determine if any results mimicked the clusters of Northwestern. Since the researchers were using pre-diagnosis data that consist of different types of features, it was unclear if any similar patterns would emerge. Figure 9 shows a screenshot of the interface with the data loaded. On the left side, there are a variety of clustering results that emerge with high rankings based on the quality metrics. Some have as few as 2 clusters, and other as many as 20. The lack of agreement about the number of clusters, even among high quality results, initially surprised the researchers. However, the researchers remarked the visualization made it clear that their selection of a clustering algorithm and parameter would have an important impact on their analysis.

The researchers initially focused on the results with fewer clusters, as the Northwestern analysis resulted in only 3 groups of patients. However, these clusterings (Results 3 and 5 in Figure 9) were not of equal sizes like the Northwestern clustering. Both of these results had one small cluster, alongside two bigger clusters. Furthermore, when the users selected the results, the important features that appeared in the *Ranked Features* view, calculated using feature selection techniques, did not map to the discriminating features mentioned in the Northwestern study. The researchers remarked that had they only tried looking at results with 3 clusters, they would have been unsatisfied with the results and may have concluded that it was not possible to replicate the results using pre-diagnosis data.

Thankfully, the overview of *Clustervision* made it clear there were other high quality results with more clusters. This led them to wonder that perhaps there would be additional clusters of patients not present in the Northwestern study. The researchers quickly scanned the ranked list of results and found other results with a larger number of clusters but also contain roughly three clusters of similar size (which mimic the cluster sizes of the Northwestern study). Clustering result 7 (highlighted) fit this description with 5 total clusters, but the 3 smaller clusters appeared to be of roughly the same size based on the width of their vertical stripes.

5.3 Finding New Clusters

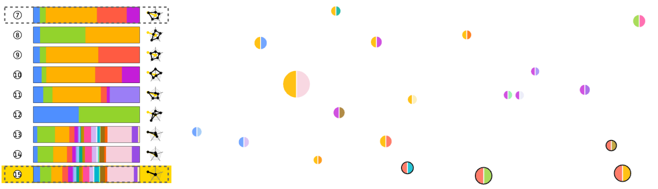
After selecting the result, the *Ranked Features* view showed the top features responsible for these diverging clusters involved Age, Atrial Fibrillation, Diabetes, Chronic Kidney Disease, and Hypertension. Remarkably, these comorbidities overlapped with many of the comorbidities used by the authors in the Northwestern clustering to distinguish the patient groups. The researchers felt like they were back on track and gained confidence that unsupervised clustering might still be an effective technique.

After examining *Parallel Trends* in Figure 9 (middle bottom), the blue cluster (N=89) appears to feature the oldest population and have a high count of Atrial Fibrillation diagnoses, which resembles Northwestern Group 3. The green cluster (N=82) appears to have the youngest population and also the least amount of diagnoses as its trend interval hovers close to zero for all of the top dimensions, which resembles Northwestern Group 1. The purple cluster (N=178) involves a population aged between these two groups, and has high prominence of Diabetes, Hypertension, Obesity, which resembles Northwestern Group 2. The only key difference between these two groupings is that in the Northwestern clustering, Group 3 had the highest prominence of Chronic Kidney Disease, whereas this occurs in the purple group in our analysis. Nonetheless, this exploration that led to groupings consistent with Northwestern clustering results was a promising finding.

While the smaller clusters map well to the existing Northwestern clusters, an open question remains about the two larger clusters. The younger red cluster (N=427, average age 65) and the older yellow cluster (N=752, average age 80) have trend lines that hover close to 0 for most medications and co-morbidities. Is it because these patients have little data because the cluster is only using examining pre-diagnosis data? Or is it because these clusters themselves feature many heterogeneous groupings that need to be examined in more detail?

5.4 Comparing Clustering Results

Motivated by the latter question, the researchers decided to explore a clustering result that breaks down the red cluster into multiple groupings. The researchers enabled the *Clustering Comparison* view to compare the above result with Clustering Result 15 which had 20 total clusters. After also enabling *Superpoints*, this view made it clear that the red cluster split into four smaller clusters that appear to be distinguishable based on treatments. These four small clusters are selected in Figure 10 with a black outline. Clicking on each superpoint allowed the researchers to see the summaries of each clustering. This result features a gold cluster (N=153) with higher counts of Statins, Ace Inhibitors, Beta Blockers, and Calcium-Channel Blockers, a brown cluster (N=38) with higher counts of ACE Inhibitors and Statins only, a green cluster (N=156) with higher counts of Thiazides and Thiazide-like Diuretics, and a teal cluster (N=64) with higher counts of Calcium Channel Blockers and Loop Diuretics. This exploration led to the insight that these subgroups of patients may have been treated differently before their diagnosis and likely represent different patient phenotypes.



(a) Clustering Results (b) Superpoints comparing Clusterings 7 and 15

Fig. 10. Researchers used *Clustering Comparison* and *Superpoints* to break down the large red cluster into meaningful subgroups.

The researchers concluded that the interactive features of *Clustervision* empowered them to do analyses they might otherwise not have considered. By having access to an overview of high quality clustering results, they considered additional clustering algorithms they were previously unfamiliar with, as well as additional parameters. The researchers remarked that since *Clustervision* automatically ranked clusterings with different parameters, they were unconstrained to the parameters used by the Northwestern study, and likely made novel discoveries about their dataset that might not have been unearthed using their traditional analysis techniques. The researchers are excited about these discoveries and hope to validate these findings in an upcoming clinical publication.

6 Conclusion and Discussion

In this paper, we demonstrated how the design and implementation of an interactive visual analytics system, *Clustervision*, can help data scientists find good and meaningful clusterings of their data. *Clustervision* accomplishes this by integrating clustering techniques and quality metrics with coordinated visualizations that allow users to interactively explore and analyze clustering results at various levels. Finally, we presented a case study, which involved a team of data scientists using *Clustervision* to find meaningful clusters of patients with a subtype of heart failure. Their use of the tool led to improved groupings of patients, which they plan to publish in an upcoming medical journal.

Our work opens up many interesting paths towards users' full comprehension of clustering. However, there are still many challenges to further support the needs of users. Users could benefit from more concretely having access to stability metrics that measure how often a set of data points are grouped together across multiple clustering results. Stability can be a clue to users of how accurate a grouping may be. Furthermore, it might be possible to give users more control over interactively defining and validating distance functions so that users can steer clustering results with respect to different feature subspaces of relevance. Finally, the team of data scientists would like to extend the work for interactive segmentation of not just static features but also temporal data, which is often a challenging problem in healthcare. As these future directions illustrate, there is great promise for the use of advanced clustering tools in many domains. We believe *Clustervision* is a first step in that direction to supporting exploration of high quality and diverse clustering results to help users find clustering results they may have otherwise missed.

References

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and Computational Geometry*, pp. 1–30. University Press, 2005.

[2] C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. Chapman and Hall/CRC, 2013.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the ACM*

SIGMOD International Conference on Management of Data, pp. 49–60. ACM, New York, NY, USA, 1999.

[4] P. Awasthi. Local algorithms for interactive clustering, 2014.

[5] M.-F. Balcan and A. Blum. Clustering with interactive feedback. In *Algorithmic Learning Theory*, pp. 316–328. Springer, 2008.

[6] S. Basu, A. Banerjee, and R. J. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 333–344. Society for Industrial and Applied Mathematics, 2004.

[7] S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 1 ed., 2008.

[8] E. Bertini, H. Lam, and A. Perer. Summaries: a special issue on evaluation for information visualization. *Information Visualization*, 10(3), 2011.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[10] B. A. Borlaug and M. M. Redfield. Diastolic and systolic heart failure are distinct phenotypes within the heart failure spectrumresponse to borlaug and redfield. *Circulation*, 123(18):2006–2014, 2011.

[11] L. Boudjeloud-Assala, P. Pinheiro, A. Blansch, T. Tamisier, and B. Otjacques. Interactive and iterative visual clustering. *Information Visualization*, 15(3):181–197, 2016.

[12] P. Bruneau, P. Pinheiro, B. Broeksema, and B. Otjacques. Cluster sculptor, an interactive visual clustering system. *Neurocomputing*, 150:627–644, 2015.

[13] N. Cao, D. Gotz, J. Sun, and H. Qu. DICON: Interactive Visual Analysis of Multidimensional Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2581–2590, 2011.

[14] R. Caruana, M. Elhawary, and N. Nguyen. Meta clustering. In *In Proceedings IEEE International Conference on Data Mining*, 2006.

[15] K. Chen and L. Liu. VISTA: Validating and Refining Clusters Via Visualization. *Information Visualization*, 3(4):257–270, 2004.

[16] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, Feb. 1979.

[17] M. desJardins, J. MacGlashan, and J. Ferraioli. Interactive Visual Clustering. In *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 361–364. ACM, New York, NY, USA, 2007.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. AAAI Press, 1996.

[19] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[20] M. Halkidi and M. Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 187–194. IEEE Computer Society, Washington, DC, USA, 2001.

[21] M. Kozak. "a dendrite method for cluster analysis" by caliski and harabasz: A classical work that is far too often incorrectly cited. *Communications in Statistics - Theory and Methods*, 41(12):2279–2280, 2012.

[22] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[23] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[24] M. Lichman and K. Bache. UCI machine learning repository, 2013.

[25] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 911–916. IEEE Computer Society, Washington, DC, USA, 2010.

[26] S. L'Yi, B. Ko, D. Shin, Y.-J. Cho, J. Lee, B. Kim, and J. Seo. XCluSim: a visual analytics tool for interactively comparing multiple clustering results of bioinformatics data. *BMC Bioinformatics*, 16(11):S5, 2015.

[27] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[28] E. Maguire, I. Koutsakis, and G. Louppe. Clusterix: a visual analytics approach to clustering. In *Symposium on Visualization in Data Science at IEEE VIS*, 2016.

[29] A. Mayorga and M. Gleicher. Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE transactions on visualization and computer graphics*, 19(9):1526–1538, 2013.

[30] S. I. T. M. Michael Sedlmair, Matt Brehmer. Dimensionality reduction in the wild: Gaps and guidance. In *UBC Computer Science Technical Report TR-2012-03*, 2012.

[31] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1):91–118, 2003.

[32] A. C. Muller and S. Guido. *Introduction to machine learning with Python*. O'Reilly Media, 2017.

[33] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *Acm Sigkdd Explorations Newsletter*, 6(1):90–105, 2004.

[34] J. M. Phillips, P. Raman, and S. Venkatasubramanian. Generating a diverse set of high-quality clusterings. In *Proceedings of the 2Nd International Conference on Discovering, Summarizing and Using Multiple Clusterings - Volume 772*, pp. 80–91. CEUR-WS.org, Aachen, Germany, Germany, 2011.

[35] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 109–116. ACM, 2004.

[36] S. Richardson and P. J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.

[37] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[38] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[39] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, 2002.

[40] S. J. Shah, D. H. Katz, S. Selvaraj, M. A. Burke, C. W. Yancy, M. Gheorghiade, R. O. Bonow, C.-C. Huang, and R. C. Deo. Phenomapping for novel classification of heart failure with preserved ejection fraction. *Circulation*, 131(3):269–279, 2015.

[41] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[42] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the BELIV Workshop*, pp. 1–7, 2006.

[43] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):629–638, 2016.

[44] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2000.

[45] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Integrating Cluster Formation and Cluster Evaluation in Interactive Visual Analysis. In *Proceedings of the Spring Conference on Computer Graphics*, pp. 77–86. ACM, New York, NY, USA, 2013.

[46] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005.