Abstract—Data clustering is a common unsupervised learning method frequently used in exploratory data analysis. However, identifying relevant structures in unlabeled, high-dimensional data is nontrivial, requiring iterative experimentation with clustering parameters as well as data features and instances. The space of possible clusterings for a typical dataset is vast, and navigating in this vast space is also challenging. The absence of ground-truth labels makes it impossible to define an optimal solution, thus requiring user judgment to establish what can be considered a satisfiable clustering result. Data scientists need adequate interactive tools to effectively explore and navigate the large space of clusterings so as to improve the effectiveness of exploratory clustering analysis.

We introduce Clustrophile 2, a new interactive tool for guided clustering analysis. Clustrophile 2 guides users in clustering-based exploratory analysis, adapts user feedback to improve user guidance, facilitates the interpretation of clusters, and helps quickly reason about differences between clusterings. To this end, Clustrophile 2 contributes a novel feature, the clustering tour, to help users choose clustering parameters and assess the quality of different clustering results in relation to current analysis goals and user expectations.

We evaluate Clustrophile 2 through a user study with 12 data scientists, who used our tool to explore and interpret sub-cohorts in a dataset of Parkinsons disease patients. Results suggest that Clustrophile 2 improves the speed and effectiveness of exploratory clustering analysis for both experts and non-experts.

Index Terms—Clustering tour, Guided data analysis, Exploratory data analysis, Interactive clustering analysis, Interpretability, Explainability, Visual data exploration recommendation, Dimensionality reduction, What-if analysis, Clustrophile, Unsupervised learning.

1 INTRODUCTION

The success of exploratory data analysis (EDA) depends on the discovery of patterned relations and structures among data instances and attributes. Clustering is a popular unsupervised learning method [15] used by analysts during EDA to discover structures in data. A cluster is a collection of samples that are considered similar to each other and dissimilar from those belonging to other clusters. By automatically dividing data into subsets based on a measure of similarity, clustering algorithms provide a powerful means to explore structures and variations in data.

However, there is currently no single systematic way of performing exploratory clustering analysis: data scientists iteratively combine clustering algorithms with different data-transformation techniques such as data preprocessing, feature selection and dimensionality reduction, and experiment with a large number of parameters. This is an iterative process of trial and error based on recurring formulation and validation of assumptions on the data. Data scientists make multitude decisions in determining what constitutes a cluster, including which clustering algorithm and similarity measure to use, which samples and features (dimensions) to include, and what granularity (e.g., number of clusters) to seek.

The space of clusterings determined by different choices algorithms,
parameters, and data samples and attributes is vast. The sheer size of this exploration space is the first challenge in exploratory clustering analysis. Data scientists need tools that facilitate iterative, rapid exploration of the space of data clusterings. The second important challenge is how to navigate efficiently in this large space rather than mere ad hoc wandering. Therefore, clustering tools would benefit from incorporating techniques that guide users, imposing a structure over the space of clustering that leads to efficient navigation. Although clustering is unsupervised by definition, we consider here the common case of complete absence of labels for validation (sometimes referred to as “fully unsupervised clustering”). If formal validation is not possible, how do we estimate the meaningfulness of the outcome of a clustering algorithm? Using the concepts of cluster compactness (closeness of data points within the same cluster) and separation (how far a cluster is from others), different internal validation measures have been introduced to estimate the “goodness” of a clustering and compare it to other clustering results. Though widely used, these metrics fail to incorporate the context of the analysis and the user’s goals and her prior knowledge and expectations, which often have significant role in determining the meaningfulness of a clustering result. Each internal validation metric takes into account slightly different characteristics of the data and should be carefully chosen based on the clustering task. There is indeed no absolute best criterion that, independently of the data and of the final aim of the clustering, can establish the best result for the user’s needs.

We introduce Clustrophile 2, a new interactive tool for guided clustering analysis. Clustrophile 2 guides users in clustering-based exploratory analysis, adapts user feedback to improve user guidance, facilitates the interpretation of clusters, and helps reason quickly about differences between clusterings. To this end, Clustrophile 2 contributes a novel feature, the Clustering Tour, to help users choose clustering parameters and reason about the quality of different clustering results in line with user analysis goals and expectations. We evaluate Clustrophile 2 through a user study with 12 data scientists of varying skill sets who used our tool to explore and interpret sub-cohorts in a dataset of Parkinson’s disease patients. We find that the Clustering Tour enables data scientists to discover algorithms and parameters that they are unfamiliar with or hesitant to use. Similarly, the Clustering Tour helps data scientists avoid prematurely fixing on a particular set of data attributes or algorithmic parameters during exploratory analysis.

In addition to the Parkinson disease dataset used for the evaluation study, we use the OECD Better Life (OECD for short) dataset [30] to demonstrate the use of our tool in figures throughout the paper. The OECD dataset consists of eight numerical socioeconomic development indices of 34 OECD member countries. Below we first summarize related work and then discuss our design criteria for Clustrophile 2. We then present the user interface of Clustrophile 2 along with the integrated visualizations and interactions, operationalizing the design criteria presented. Then we introduce the clustering tour and the underlying graph model. Next we present a qualitative user study conducted with 12 data scientists, followed by an in-depth discussion of the study results. We conclude by summarizing our contributions and reflecting on lessons learned.

2 RELATED WORK

Clustrophile 2 draws from prior work on interactive systems for visual clustering analysis and guided exploratory data analysis.

2.1 Tools for Visual Clustering Analysis

Prior research applies visualization to improve user understanding of clustering results across domains. Using coordinated visualizations with drill-down/capabilities is a typical approach in earlier interactive tools. The Hierarchical Clustering Explorer (HCE) [40] is an early and comprehensive example of interactive visualization tools for exploring clusterings. HCE supports the exploration of hierarchical clusterings of gene expression datasets through dendrograms (hierarchical clustering tree) stacked up with heatmap visualizations.

Earlier research has also introduced clustering comparison techniques in interactive systems [6, 21, 24, 33, 40]. DICON [6] encodes statistical properties of clustering instances as icons and embeds them in the plane on the basis of similarity using multidimensional scaling. Pilhofer et al. [33] propose a method for reordering categorical variables to align with each other, thus augmenting the visual comparison of clusterings. Others have proposed similar visual encoding techniques for comparing different clusterings of data dimensions with applications to gene expression datasets in mind. To that end, HCE [40], CComVis [53], Matchmaker [21], StratomeX [22] and XCluSim [24] all encode changes across clusterings of dimensions by essentially tracing them with bundled lines or ribbons.

Researchers have also proposed tools that incorporate user feedback into clustering. Matchmaker [21] builds on techniques from [40] with the ability to modify clusterings by grouping data dimensions. ClusterSculptor [29] and Cluster Sculptor [5] enable users to supervise clustering processes. Schreck et al. [39] propose using user feedback to bootstrap the similarity evaluation in data space (trajectories, in this case) and then apply the clustering algorithm. FURBY [43] lets users refine or improve fuzzy clusterings by choosing a threshold, transforming fuzzy clusters into discrete ones.

ClustVis [27] uses both PCA and clustering heatmaps but in isolation without interaction or coordination. Clustrophile 7 coordinates heatmap visualizations of discrete clusterings with scatterplot visualizations of dimensionality reductions. It also enables ANOVA-based significance testing and visual correlation analysis along with what-if analysis through new interactions on dimensionality reduction scatterplots. Akin to Clustrophile, ClusterVision [20] incorporates significance testing and couples clustering visualizations with dimensionality reduction scatterplots. Clustrophile 2 extends Clustrophile with 1) a richer set of clustering algorithms, 2) new features to guide users in clustering analysis, including the Clustering Tour, and 3) broader and deeper support for visual and statistical analysis and validation of clustering instances and clusters.

2.2 Guiding Users in Exploratory Data Analysis

Prior research in data analysis propose various tools and techniques to guide users in exploring low-dimensional projections of data. For example, PRIM-9 (Picturing, Rotation, Isolation, and Masking—in up to 9 dimensions) [10] enables the user to interactively rotate the multivariate data and view a two-dimensional projection of the data continuously updated. Motivated by user behavior in the PRIM-9 system, Friedman and Tukey [11] propose first the projection index, a measure for quantifying the “usefulness” of a given projection plane (or line), and then an optimization method, projection pursuit, to find projection direction that maximizes the projection index value. The proposed index considers projections that result in large spread with high local density to be useful (e.g., highly separated clusters). In a complementary approach, Asimov introduces the grand tour, a method for viewing multidimensional data via orthogonal projections onto a sequence of two-dimensional planes [2]. Asimov considers a set of criteria such as density, continuity, and uniformity to select a sequence of projection planes from all possible projection planes. Hullman et al. [16] study how to generate visualization sequences for narrative visualization, using a directed graph for modeling the sequence space. Similar to Hullman et al., Clustrophile 2 also models the visual exploration space as a graph. However, Clustrophile 2 uses an undirected graph model and focuses on modeling clustering state space. GraphScape [18] extends Hullman et al. with an actionable model implemented using the Vega-Lite grammar and a new transition cost function. Akin to GraphScape, Clustrophile 2’s modeling of the state space and state transitions is actionable. While GraphScape purely considers chart specifications without taking data or user preferences into consideration, Clustrophile 2 transitions consider both data, clustering parameters, and user preferences.

Visualization recommender systems also model the space of visual exploration and evaluate various measures over the space to decide what to present the user. For instance, Rank-by-Feature [41], AutoVis [50], Voyager [51], SeeDB [47], and Foresight [8] use statistical features and perceptual effectiveness to structure the presentation of possible visualizations of data. Clustrophile 2 also provides methods for enumeration and ranking of visual exploration. However, while recommendation systems typically focus on suggesting individual items based on attributes, Clustrophile 2 uses the Clustering Tour to focus on clusterings and their visualizations, complementing existing recommender systems. SOMFlow [38] enables iterative clustering together with self-organizing maps (SOMs) to analyze time series data. To guide users, SOMFlow also uses clustering quality metrics. Clustrophile 2 goes beyond the use of quality metrics, considering user feedback, clustering parameters, data features along with interpretable explanations in guiding users.
3 Design Criteria

We identify a set of high-level design criteria to be considered in developing systems for interactive clustering analysis. These criteria are based both on research literature and on the regular feedback we received from data scientists during the development of Clustrophile 2.

D1: Show variation within clusters Clustering is useful for grouping data points based on similarity, thus enabling users to discover salient structures. The output of clustering algorithms generally consists in a finite set of labels (classes, clusters) to which each data point belongs. In fuzzy clustering, the output is the probability of belonging to one of those classes. However, in both cases the user receives little or no information about the differences among data points in the same cluster. Clustrophile 2 combines various visualizations of the data—table, scatterplots, matrix diagrams, distribution plots—to facilitate the exploration of data points at different levels of granularity. In particular, scatterplots represent dimensionally reduced data and thus provide a continuous spatial view of similarities among high-dimensional data points.

D2: Allow quick iteration over parameters The outcome of a clustering task is highly dependent on a set of parameters: some of them may be chosen based on the type of data or the application domain, others are often unknown a priori and require iterative experimentation to refine. Clustrophile 2 enables users to interactively update and apply clustering and projection algorithms and parameters at any point while staying in the context of their analysis session.

D3: Represent clustering instances compactly It is important for users to be able to examine different clustering instances fluidly and independently without visual clutter or cognitive overload. We define “clustering instance” to be the outcome of a clustering algorithm according to specific parameters. The Clustrophile 2 interface uses the “Clustering View” element as the atomic component representing a clustering instance. Clustering View pairs a projection scatterplot and a clustering heatmap, using two complementary visualizations. A compact, self-descriptive representation is also useful for visually comparing different clustering instances. Clustrophile 2 lets users work simultaneously on multiple Clustering Views, which can be freely organized by users across the interface and help them keep track of how different choices of features, algorithms and distance measures affect clustering results.

D4: Facilitate interpretable naming How to attach meaning to the “learned” structures in clustering is an important yet challenging problem. It is essential to facilitate the meaningful naming and description of clusters and clustering instances. For each cluster computed, Clustrophile 2 designates the cluster centroid as the cluster representative and assigns its identifier as the cluster name. Clustrophile 2 lets the user freely rename the cluster according to her understanding of the data.

D5: Support analysis of large datasets The ability to interactively explore and analyze large datasets is important for analysts in many domains and has been a major request of our collaborators. Clustrophile 2 adopts caching, precomputation, sampling and feature selection, which can be considered a form of importance sampling, to support analysis with larger datasets. Addressing computational scalability also helps mitigate the visual scalability issues. Clustrophile 2 also supports common interaction techniques such as panning & zooming and visual grouping with smooth convex-hull patches to reduce visual clutter. Finally, Clustrophile 2 lets the user select a set of “preferred features” that will have higher rendering priority in situations in which it would be hard to represent many data features, as in the matrix plot (heatmap) of the Clustering View.

D6: Support reasoning about clusters and clustering instances Users often would like to know what features (dimensions) of the data points are important in determining a given clustering instance, or how different choices of features or distance measures might affect the clustering, or whether it is a “valid” or “good” clustering. Users also would like to understand the characteristics of data points within a given cluster that distinguish them from other clusters and how these data points came to be in the cluster. Clustrophile 2 dynamically chooses a combination of metrics based on the data and user preference in supporting clustering analysis. It also includes automated metric suggestions, visual explanations (e.g., decision-tree based cluster visualization), quantitative indicators (e.g., stability and confidence scores), textual descriptions and hyperlinks to help user better interpret results and make informed decisions, eschewing the blind use of clustering parameters and validation methods.

D7: Promote multiscale exploration Clustering requires users to iteratively generate hypotheses and try to validate them. Consequently, it is fundamental for interpretability and trust to arm the user with the ability to reach the same solution in multiple ways, thus enabling data exploration at multiple scales. Clustrophile 2 integrates views to compare and analyze single clustering results, drill down the perspective of single clusters, identify potential subclusters and even inspect single points with respect to their feature distribution. Clustrophile 2 lets users view statistical summaries for each cluster and perform “isolation” [11], which enables splitting clusters characterized by mild features into more significant subclusters. Dynamic filtering and selection of single data points are also implemented and coupled with statistical analysis to identify and eventually remove outliers and skewed distributions in the data.

D8: Keep a stateful representation of the current analysis One of the primary motivations for data analysts in using interactive tools is to increase their productivity or save time. The iterative nature of a clustering analysis continuously forces users to try out different parameters and features, perform a set of computations and decide which of the many directions to take nextmaking the analysis session extremely hard to reproduce. Clustrophile 2 enters each operation performed by users in a log, enabling them to undo/redo single operations and also to review the workflow of their analysis.

D9: Guide users in clustering analysis Due to the number of possible combinations, iterative experimentation on different clustering parameters can be non-trivial or time consuming, and becomes even more challenging in a high-dimensional dataset. Furthermore, most users do not know in detail the advantages and disadvantages of clustering or projection methods, sometimes choosing them blindly and simply trying all possible parameter combinations. It is thus important that the system provide assistance to the user in navigating complex clustering spaces, while incorporating the user’s feedback in the process. Clustrophile 2 provides textual explanations with suggestions on when it could be worth using certain parameters with references (hyperlinks) to existing literature. Clustrophile 2 also provides automated suggestions based on the dataset currently analyzed, on previous computations and on user preferences. Clustrophile 2 introduces a novel feature, the Clustering Tour. The Clustering Tour recommends a sequence of clusterings based on clustering configuration choices, data features, and user feedback. It samples the clustering space promoting coverage in the absence of user feedback. When the user “likes” a recommended clustering, Clustrophile 2 recommends “nearby” clusterings.

4 User Interface and Interactions

In this section we briefly describe the main components of the Clustrophile 2 interface and interactions. Clustrophile 2 has been developed iteratively according to the design considerations introduced in the previous section. We refer back to the relevant design criteria to motivate our design choices. Clustrophile 2 interface consists of a main, central view Fig. 1, two collapsible sidebars (left Fig. 1 and right Fig. 1) and multiple modal windows displayed with relevant interactions. The left sidebar (or Navigation Panel) contains a button menu to...
import datasets from comma-separated-values (CSV) files, load data from previous analyses and export the results (i.e. clusters, chart images) of the current session. Clustrophile 2 supports saving the current state of the analysis (D8) for follow-up analysis and sharing it with contributors who are also listed in the Navigation Panel. The right sidebar (hidden by default) records a log of the operations and parameter changes made by the user (Fig. 1), enabling him easily to revert the analysis to a previous state (D8). A convenient list of the top pairwise feature correlations in the dataset is also displayed (Fig. 1), facilitating a quick overview of statistical dependencies. The main view is subdivided into an upper region containing the Data Table (Fig. 1) and a lower region that displays one or more clustering views. Clustrophile 2 in fact enables data scientists to work simultaneously on multiple clustering instances (D3), but at the same time links the coordinated Data Table view to only one instance at a time. The currently selected clustering instance is generally the one the user last interacted with, and its corresponding Clustering View is marked with a blue header. The selected instance and its cluster names are also made available in the Navigation Panel (Fig. 1).

4.1 Visualization Views

Clustering View A Clustering View (Fig. 2) represents a single clustering instance and has the goal of both visualizing the identified clusters and characterizing them based on their distinctive features. In our user interface, the Clustering View also lets the user dynamically change projection and clustering parameters for an associated clustering instance, and keeps them always visible for easier comparison with other Clustering Views.

The minimal set of visualizations we choose to summarize a clustering instance consists of a scatterplot (Fig. 2, left) and a heatmap (Fig. 2, right). The scatterplot provides a two-dimensional projection of the data obtained using dimensionality reduction and encodes clustering assignments through color. Since clustering algorithms divide data into discrete groups based on similarity, projections are a natural way to represent different degrees of variation within and between groups as distance between data points (D1). Each cluster of points can also be identified by a colored convex hull, simplifying the visualization in case of a larger number of data points (D5).

The heatmap aims instead at providing a qualitative and quantitative description of each cluster based on the aggregate feature values of the data points it contains. Each column of the matrix represents a cluster, rows represent data features (dimensions). The color of each cell encodes the average value of cluster members for a specific feature with respect to the feature distribution. For instance, in the heatmap in Fig. 2 the dark red cells indicate that the Red cluster is characterized by very high WORKINGHOURS, whereas the dark blue cells indicate that the same cluster has very low EDUCATIONALATTAINMENT and STUDENTSKILLS (i.e., red means higher values, blue lower values). This way, each cluster can be quickly described (D6) by observing the heatmap vertically (e.g. intense colors indicate the key features identifying a cluster, mild colors indicate average values); similarly, clusters can be compared by looking horizontally at the matrix diagram (e.g., from the second row of the heatmap, it is easy to see that the green cluster is the one with highest STUDENTSKILLS). By hovering on each cell, the user can inspect the average feature value of each cluster and the p-value associated to the current selection feature algorithm (which encodes the relevance of a feature). Clusters are ordered from largest to smallest and display their member number and color right beneath each column. Since with high-dimensional datasets (D5) the number of rows would become too large, we display only the top ten relevant features, which are chosen automatically by a feature selection algorithm (more on this later) or manually selected by the user.

Users can select one or more data points or clusters from both the scatterplot and the heatmap. When a selection is performed, it is reflected both with visualizations and the Data Table. The right sidebar further lets users re-cluster and re-project only the selected points, an operation particularly useful for finding subclusters (D7). From the Clustering View, users can dynamically change the parameters associated to the associated clustering instance. Currently supported clustering methods include K-means, Agglomerative (Hierarchical), Spectral [42], DBSCAN [9], Birch [52], Cure [13] and CLIQUE [1] algorithms that, as applicable, can be combined with ten different clustering metrics and three elements of linkage strategies. Sixty algorithms are also available: PCA [45], MDS [19], CMDs [46], t-SNE [25], Isomap [44] and LLE [37]. Users can also define custom projection and clustering algorithms and metrics. We note that by default Clustrophile 2 applies dimensionality reduction and clustering in high-dimensional space, and then visualizes the results using, respectively, a scatterplot and a heatmap.

The user can control the number of displayed clusters through a slider located underneath the heatmap (Fig. 2). Different numbers of clusters are automatically precomputed by Clustrophile 2 based on user settings, so that the user can quickly change the number of clusters without waiting for further computations (D5). Another parameter that can be chosen from the clustering view is the sampling rate of the data; this is useful for doing clustering in the presence of larger datasets (D5).

Data Table While the Clustering View provides a high-level summary of a clustering instance, it is fundamental for data scientists to be able to drill down in the data and inspect individual data samples. The Data Table view gives the user the raw data, supporting statistical analysis, automatic outlier detection, selection and filtering. These features in particular make it possible to reason about how single data points and feature distributions affect the current clustering, and help the user decide which dimensions should be considered or excluded by the clustering algorithm.

The Data Table (Figure 1a) contains a dynamic table visualization of the current dataset in which each column represents a feature (dimension) and each row represents a data sample. The Data Table displays the data and cluster assignments associated only to the currently selected Clustering View. For each row, a vertical, colored bar encodes the cluster of membership of the associated data sample (Figure 1b), whereas a set of green or red bars respectively identify particularly high or low feature values (“outliers”) with respect to each feature distribution (Figure 1b). Clicking on the buttons next to each feature name orders rows by cluster or by column and displays basic statistics on a particular feature in a pop-up window (Figure 3d). In particular, Clustrophile 2 can compare the statistical values computed on the currently selected rows and those of the whole dataset (Figure 3g), plus a histogram plot of the feature distribution. A list of the features that correlate most to the selected feature is also given, allowing quick discovery of data trends. The search functionality (Figure 3a) lets users select data samples using arbitrary keyword search on feature names and values. Users can also filter the table using expressions in a mini-language (Figure 3b). For example, typing age > 40 & weight < 180 dynamically selects data points across visualizations in which the fields age and weight satisfy the entered constraint. When some rows are selected, the corresponding points of the scatterplot and cluster columns in the heatmap in the current Clustering View are highlighted.

Cluster Details While the Data Table works well for inspecting single data points and feature distributions across the dataset, the Cluster Details modal (Fig. 4) aims at a deeper characterization of a specific cluster. The modal can be opened by double-clicking on any cluster in the user interface and contains statistical information about the members of the cluster—such as most relevant features, top pairwise feature correlations and outliers. The user can use this view to assign a custom name to a cluster or to display the histogram for each feature distribution.
with theoretical advantages/drawbacks and use cases for each method.

Fig. 4: Cluster Details modal window. Double-clicking on a cluster lets the user access cluster-specific statistical information. The modal window shows b) an automatically generated description for the cluster, to which the user can a) associated a custom name. The modal provides overall c) relevance and correlation analysis and d) detailed aggregate statistics and a histogram density estimation of the currently selected attribute for the data points in the cluster. This modal view also includes e) an interactive data table containing only the members of the cluster.

4.2 Raising Awareness About Choosing Parameters

Given the high number of parameter combinations that may influence a clustering outcome, it is important to guide users towards a reasonable choice of parameters in the context of the current analysis. From each Clustering View, the user can access a “Help me decide” panel containing a tab dedicated to the choice of each parameter.

Feature selection The choice of which features of the original dataset to feed to the clustering algorithm can strongly influence both the performance and the quality of clustering results. To help the user understand if and which data dimensions should be included in the analysis, Clustrophile 2 provides a list of the most relevant data dimensions according to several feature-selection algorithms useful in removing features with low variance or high pairwise correlation (e.g. variance threshold, feature agglomeration, random projection) and in filtering out noisy or uninfluential features (e.g. univariate feature selection based on chi-squared or ANOVA F-value, recursive feature elimination [14]).

Clustrophile 2 also introduces a hierarchical clustering of the dataset’s features (Fig. 5b), displaying through a scatterplot and a dendrogram how data dimensions can be grouped together based on similarity through feature agglomeration. After applying any feature selection strategy, the user can choose to apply the suggested selection or choose manually which features to consider in the clustering.

Sampling In presence of larger datasets (more than 10,000 data samples) Clustrophile 2 suggests that the user perform clustering only after sampling the original data, in order to speed up the computation. Since the analysis is unsupervised, the user can change only the percentage of random sampling.

Clustering algorithm, metric and linkage For each possible choice of clustering parameters, Clustrophile 2 provides a textual description with theoretical advantages/drawbacks and use cases for each method (Fig. 5a). For instance, users can learn that K-means is not suited in the presence of uneven cluster sizes and non-flat geometries, or that Cityblock affinity can outperform Euclidean distances in case of sparse data. In the case of clustering metrics and linkage criteria, Clustrophile 2 can suggest to the user which parameters to use by testing them asynchronously and picking the one that generates the best cluster separation. Hyperlinks to related literature are also included.

Number of clusters Clustering algorithms do not generally output a unique number of clusters, since this is generally a user-defined parameter. By generalizing the idea of the “elbow plot” for the K-means cost function, Clustrophile 2 precomputes a range of clustering solutions, each with a different number of clusters in a user-defined range, and compares them in a line chart (Fig. 5c). In particular, the horizontal axis corresponds to the maximum number of clusters and the vertical axis represents the value of one of the internal validation measures. Based on the metric formulation, the optimal number of clusters is given by the maximum, minimum or elbow value of the line chart [23].

When applicable, Clustrophile 2 complements the line chart with a clustering-algorithm-specific plot (e.g., a dendrogram for hierarchical clustering). A silhouette plot [36] is also included (Fig. 5c, right), providing more detailed information about which clusters were merged and which data points are critical to determining the optimal cluster number.

Projection Although the dimensionality-reduction method used to visualize the scatterplot does not influence the clustering results, it may visually bias how a user perceives the quality of a given clustering instance. To handle this, Clustrophile 2 provides a description and set of references for each projection method in addition to an automated suggestion. By precomputing each projection, our tool applies the same internal evaluation metrics used for clustering to the dimensionally reduced data and suggests a projection algorithm that optimizes cluster compactness and scatterplot separation.

4.3 Guiding Users Towards a Better Clustering

Once clustering parameters are chosen, the next step is assessing the quality of clustering outcome. In the panel “Is this a good clustering?”, Clustrophile 2 aims at helping the user reason on the absolute and relative “satisfactoriness” of the results.

Quantitative validation Since no ground truth labels are available, internal validation measures can only assess the goodness of a clustering instance. Instead of adopting only one metric, Clustrophile 2 acknowledges the pros and cons of each measure and tries to help the user choose the measure that better fits the data and requirements. Using Liu et al.’s work [23], we associate the performance of each validation metric to a set of five conditions: presence of skewed distributions, subclusters and different cluster densities; robustness of the algorithm to noise; and monotonicity of the measure’s cost function. While the first three can be automatically inferred from the data, the last two are dictated by user preferences. For instance, using Silhouette in presence of subclusters or Calinski-Harabasz with noisy data could lead the user to a non-optimal cluster choice. On top of briefly describing each measure, Clustrophile 2 filters them dynamically to show only those that match the user’s interest and displays how their value changes based on the number of clusters (Fig. 5e).

Interpretability We believe that, in addition to a quantitative evaluation, a qualitative analysis of the clustering results is fundamental in understanding the user goal has been reached. Even if the user is exploring the data freely, it is important to give an interpretation to each cluster in relation to its features. To this end, we apply decision trees [4], in combination with cluster average feature values, as an approximate and generalizable solution for cluster interpretability. Once a cluster results is obtained, we use its clustering assignments as ground-truth labels to train a decision-tree classifier, whose decision rules are then displayed in a tree diagram (Fig. 5d). By interactively exploring paths of the decision tree, the user can reason about the main features used to distinguish data points in different clusters. By combining the information presented in the Clustering View’s heatmap, Clustrophile 2 also provides a textual description of each identified cluster.

Uncertainty Most clustering algorithms output discrete classes of objects: either a data point belongs to cluster A or it belongs to cluster B. However, as can be seen in the Clustering View’s scatterplot, the position of some “outlier” data points may suggest a forced clustering assignment. Small differences in clustering parameters can easily cause uncertain data points to change their cluster assignment, unbalancing the size of the clusters involved and sometimes hiding some of their distinctive features. For this reason we believe that being aware of low-confidence clustering assignments is important, and we propose a dedicated panel where these critical points are displayed through a variation of Clustering View’s scatterplot (Fig. 5f). When fuzzy clustering confidence values are not available, we use the distribution of per-point silhouette scores to determine which data points are uncertain.
4.4 Clustering Tour

By iteratively changing all clustering parameters, a user can dynamically explore the space of possible clustering solutions until a satisfactory outcome is found. However, even with guidance in parameter choice, the space of possible parameter combinations and clustering solutions is too large to explore manually in its entirety. There are certain parameter choices that largely affect the clustering outcome, whereas other parameters have minimal influence on the results. Overall, it would be useful to let users explore first the parameter choices that determine solutions very different from each other, metaphorically making large leaps in the space of possible clusterings, in order to get a quick tour of the data. If the user likes a solution and wants to refine it, then other parameter choices can be made to explore solutions similar to the selected one. With this concept in mind, we introduce a Clustering Tour feature to help the user quickly explore the space of possible clustering outcomes.

The interface shown in Fig. 6 contains (a) a list of previously explored solutions, (b) a scatterplot and a heatmap representing the current solution, (c) a set of buttons for the user to give feedback, and (d) a choice of modalities with which the user can constrain how parameters are updated.

To determine which clustering solutions to suggest to the user, our algorithm considers three fundamental measures. Let’s imagine the space of possible clustering solutions as a fully connected graph where each node represents a clustering outcome, (3) a set of buttons for the user to give feedback, and (4) a choice of modalities with which the user can constrain how parameters are updated.

The distance between two nodes in the graph is given by a linear combination of internal clustering validation measures. The distance between two nodes is also associated to their difference in parameters ($\Delta l$), and their difference in scores ($\Delta s$). We consider the current choice of parameters of the associated Clustering View to be the entry node of the branch and go back, restarting the neighborhood of the node is explored, applying smaller parameter changes and optimizing cluster separation ($\Delta s$).
the number of clusters in the current solution produces on average a $\Delta$ higher than changing the clustering metric, cluster number will have a higher weight in the determination of a possible $\Delta$. Once this first step is computed, a set of $\Delta$ is chosen giving priority to changing first parameters with higher weight. Based on our tests, changes in input features are those that determine the biggest $\Delta$ and thus those that are explored first. In the absence of user constraints, input features are determined by cyclically applying a subscore to each data dimension through the feature selection methods described in Section 4.2 and selecting those with highest relevance and/or lowest pairwise correlation. At the same time, we randomly exclude from the analysis features of very high relevance to prevent single features from biasing the clustering result. The remaining parameters are explored in a similar way by applying sampling and randomization. Once a clustering suggestion is computed, we perform a subset of the dimensionality reduction methods in Section 4.1 and apply clustering validation measures on their output to choose the one that best visualizes the separation among clusters. The result is then shown to the user, who can continue generating solutions that are very different from one another by pressing the “Generate new solution button” to explore different perspectives on the data. If the user is relatively satisfied with the current solution, he presses the “I like it” button to explore the neighborhood of the current node in the graph. In this situation high-weight parameters (often features and cluster number) tend to remain fixed, and lower-weight parameters (e.g., typically the clustering metric) are changed to produce slight variations in the clustering outcome. Only at this stage, when the semantics of the identified clusters has become stable, are validation measures used to decide which solution to suggest to the user. If the user presses the “Very bad :(" button, the Clustering Tour goes back to the previous node of the graph and explores a different set of parameters with higher weight in the determination of a possible $\Delta$. Once this first step is computed, a set of $\Delta$ is chosen giving priority to changing first parameters with higher weight. Based on our tests, changes in input features are those that determine the biggest $\Delta$ and thus those that are explored first. In the absence of user constraints, input features are determined by cyclically applying a subscore to each data dimension through the feature selection methods described in Section 4.2 and selecting those with highest relevance and/or lowest pairwise correlation. At the same time, we randomly exclude from the analysis features of very high relevance to prevent single features from biasing the clustering result. The remaining parameters are explored in a similar way by applying sampling and randomization. Once a clustering suggestion is computed, we perform a subset of the dimensionality reduction methods in Section 4.1 and apply clustering validation measures on their output to choose the one that best visualizes the separation among clusters. The result is then shown to the user, who can continue generating solutions that are very different from one another by pressing the “Generate new solution button” to explore different perspectives on the data. If the user is relatively satisfied with the current solution, he presses the “I like it” button to explore the neighborhood of the current node in the graph. In this situation high-weight parameters (often features and cluster number) tend to remain fixed, and lower-weight parameters (e.g., typically the clustering metric) are changed to produce slight variations in the clustering outcome. Only at this stage, when the semantics of the identified clusters has become stable, are validation measures used to decide which solution to suggest to the user. If the user presses the “Very bad :(" button, the Clustering Tour goes back to the previous node of the graph and explores a different direction (i.e. tries to generate a solution with high $\Delta$ from the dislik ed solution). At any point of the Clustering Tour, the user can constrain the exploration to a subset of the available parameters, deciding which ones should be fixed or changed and which ones should be decided by the algorithm. When the user is satisfied, he can decide to apply the identified parameters to the associated Clustering View.

We illustrate in Fig. 8 a sample execution of the Clustering Tour on the OECD dataset, showing the results generated by the algorithm based on user feedback.

### 4.5 System Architecture

Clustrophile 2 is deployed as cloud-based web application based on a client-server architecture model. Users get access to the tool through their web browsers, by connecting to a website hosted on a dedicated server, which also holds a MongoDB [28] database containing information about users and their datasets. In particular, the system enables users to create accounts and upload their own datasets, save the state of their analyses and share it with other users. Clustrophile 2's web interface is implemented in Javascript with the help of D3 [3] and React [35], whereas PEG.js [32] was used to compile the mini-language used for filtering. A separate analytics server on the same physical machine carries out the computations required by Clustrophile 2, exchanging input and output values with the user interface in JSON format. The analytics server, based on Flask and running Python, makes heavy use of NumPy [49], scikit-learn [31] and PyClustering [34] libraries. To minimize the amount of information exchanged between the client application and the analytics server, the latter creates a session for each user and keeps their datasets in memory (stateful server), so that front-end operations do not require sending the raw data to the back end. Secondly, we reduce the number of computational requests generated by the user through a caching system in the client application, leveraging the fact that clustering analysis is iterative by nature.

### 5 User Study

We conduct a study with twelve data scientists using Clustrophile 2 to answer an open analysis question about a real-world dataset. Our user study has two goals: 1) understanding how data scientists use the interactions, visualizations, and user-guidance features of our tool based on their level of expertise and prior knowledge of the data domain, 2) studying the overall workflows adopted by data scientists to arrive to a solution they consider satisfactory in an open-ended analysis task about a real-world dataset, where there is no guarantee of finding a solution.

#### Data

We chose a real-world dataset about subjects with Parkinson’s disease, in which there is not trivial solution to the clustering problem. The dataset has 8652 rows and 37 features, obtained after preprocessing a subset of the data made publicly available by the Parkinsons Progression Markers Initiative (PPMI). The data contains records about human subjects associated to the Unified Parkinson’s Disease Rating Scale (UPDRS), which consists of a set of measures that describe the progression of a person’s Parkinson’s disease. The measures in the scale are evaluated by interview and clinical observation of human subjects by a clinician, and include symptoms such as rigidity of upper and lower limbs, leg agility, gait, spontaneity of movement, finger and toe tapping, tremor (see [12] for the full list of measures of the UPDRS scale). While most features were UPDRS values ranging from 0 to...
Table 1: Results of our user study. Twelve participants, subdivided by data analyst archetype and by domain expertise, were asked to answer the question “Can you identify phenotypes in Parkinson’s disease?” by analyzing a real-world dataset containing patient data. The table reports the clustering parameters, features adopted and subsets considered in each analysis. Participants were also asked to choose a single clustering instance and assign an interpretable name to the identified clusters.

<table>
<thead>
<tr>
<th>Id</th>
<th>Archetype</th>
<th>Domain</th>
<th>Time</th>
<th>Isolation based on</th>
<th>Features</th>
<th>Feat. Selection</th>
<th>Algorithm</th>
<th>Metric</th>
<th>Projection</th>
<th>Clusters</th>
<th>Cluster Names</th>
<th>Validation</th>
<th>Tour</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hacker</td>
<td>No</td>
<td>60m</td>
<td>-</td>
<td>Custom</td>
<td>-</td>
<td>Agglomerative (Complete)</td>
<td>Euclidian</td>
<td>Isomap</td>
<td>5</td>
<td>Mild, Tremor dominant, Rigidity, Hand &amp; foot mobility, Posture &amp; gait issues</td>
<td>Silhouette</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Hacker</td>
<td>No</td>
<td>51m</td>
<td>Disease severity</td>
<td>Custom</td>
<td>-</td>
<td>Agglomerative (Average)</td>
<td>Cityblock</td>
<td>MDS</td>
<td>3</td>
<td>Tremor at rest, Hand Tasks, Rigidity, expression, Posture: and gait, Limbs agility</td>
<td>Silhouette</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Hacker</td>
<td>Yes</td>
<td>55m</td>
<td>-</td>
<td>Custom</td>
<td>Kmeans</td>
<td>4</td>
<td>Euclidian</td>
<td>sSNE</td>
<td>4</td>
<td>Bilateral, Unilateral left, Unilateral right, Axial</td>
<td>Silhouette</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Hacker</td>
<td>Yes</td>
<td>46m</td>
<td>Drug use,</td>
<td>Custom</td>
<td>CLIQUE</td>
<td>3</td>
<td>Euclidian</td>
<td>LLE</td>
<td>3</td>
<td>Bradykinesia, Tremor, Dyskinesia</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Scripter</td>
<td>No</td>
<td>39m</td>
<td>Afflicted side &amp; disease severity</td>
<td>ANOVA</td>
<td>K-means</td>
<td>Euclidian</td>
<td>PCA</td>
<td>4</td>
<td>Mild, Hand movements, Tremor</td>
<td>-</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Scripter</td>
<td>No</td>
<td>40m</td>
<td>Drug use,</td>
<td>PCA</td>
<td>Birch</td>
<td>Euclidian</td>
<td>MDS</td>
<td>3</td>
<td>Mild, Rapid, Tremor, Posture &amp; Gait, Extreme</td>
<td>-</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Scripter</td>
<td>Yes</td>
<td>62m</td>
<td>Drug use, Outlier removal, Random samples</td>
<td>PCA</td>
<td>Kmeans</td>
<td>Euclidian</td>
<td>PCA</td>
<td>4</td>
<td>Bradykinetic, Tremoring, Gait movements</td>
<td>-</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Scripter</td>
<td>Yes</td>
<td>24m</td>
<td>Drug use</td>
<td>Custom</td>
<td>Kmeans</td>
<td>3</td>
<td>Euclidian</td>
<td>CMDMS</td>
<td>5</td>
<td>Not impaired, Left, Right, Task driven impairment</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>User</td>
<td>No</td>
<td>34m</td>
<td>-</td>
<td>Custom</td>
<td>Kmeans</td>
<td>5</td>
<td>Euclidian</td>
<td>t-SNE</td>
<td>4 Mild, Limbs agility, Rigidity and posture, Advanced non-tremor, Advanced tremor</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>User</td>
<td>No</td>
<td>25m</td>
<td>-</td>
<td>Custom</td>
<td>Kmeans</td>
<td>3</td>
<td>Euclidian</td>
<td>PCA</td>
<td>4 Healthy, Better functioning on left</td>
<td>Silhouette</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>User</td>
<td>Yes</td>
<td>28m</td>
<td>Afflicted side</td>
<td>Custom</td>
<td>Kmeans</td>
<td>3</td>
<td>Euclidian</td>
<td>PCA</td>
<td>5 Mild, Bradykinesia, Tremor</td>
<td>-</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>User</td>
<td>Yes</td>
<td>27m</td>
<td>-</td>
<td>Custom</td>
<td>Kmeans</td>
<td>3</td>
<td>Euclidian</td>
<td>PCA</td>
<td>4 Low Symptotic, Smoothness L1/L2, Tremor at rest, Smoothness L2V2, Medication use</td>
<td>-</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 10: Parameters changed during the analysis grouped by category and user archetype. The bar chart shows that choosing input features (enabling/disabling data dimensions) was the most performed activity, while clustering method and metric were changed less often. The cluster number parameter was also highly adjusted, especially by the Application User archetype in the context of Agglomerative clustering. It is interesting to note that about 27% of the total parameters were repeated by the same user, demonstrating the highly iterative nature of exploratory data analysis and the need for caching results. Despite not choosing a clustering parameter, we report under the name “Statistics” the number of times each user accessed feature distribution information from the Data Table.

4, few others indicated the overall progression of the disease (Hoen & Yahr stage), the use of medication (ON/OFF_DOSE) and the number of hours passed from when the subject took the drug (PD_MED_USE). Task Participants were asked to complete a single task: “Identify the different phenotypes characterizing Parkinson’s disease in the given dataset.” We define “phenotypes” as the observable behaviors of a subject due to the interaction of the disease with the environment. We asked our participants to identify one clustering instance that they are satisfied with, assign a name and a description to each of its clusters, and finally explain verbally the significance of their obtained results.

Participants We recruited twelve participants with a minimum experience of two years working as data scientists. They all had at least a master’s degree in science or engineering. To recruit our participants, we first interviewed 16 candidates and then selected twelve by matching the candidates with the three analyst archetypes [17], hackers, scripters, and application users, based on their expertise and domain knowledge. We ensured that we had four participants for each of the three analyst types. Note that hackers have solid scripting and programming skills in different languages such as C++, Java, and Python and are capable of developing their own tools for data analysis, scripters are more familiar with scripting (e.g., using R, Matlab, etc.) than programming and generally have a robust background in Mathematics or Statistics, and application users conduct their analysis using spreadsheet applications such as Microsoft Excel or other off-the-shelf data analysis tools such as SAS and SPSS. For each of these archetypes, we also made sure that we had two participants with domain expertise in Parkinson’s disease or neuroscience, and two participants with no prior knowledge about the domain of the data—for a total of six domain experts and six novices of neuroscience.

Procedure The study took place in the experimenters office, where one participant at a time used Clamp 2.0 on the experimenters laptop. Participants were first briefed about the study and then given a tutorial about Clamp 2.0 for about fifteen minutes, using the OECD dataset as sample data. After the tutorial, participants were introduced to the test dataset, and the experimenter explained the medical terminology found in feature names (e.g., “pronation-supination left hand”). Regardless of their knowledge of Parkinson’s disease, all participants were given the same task of identifying groups of patients with different phenotypes in the dataset. Participants were given two hours to provide a “solution,” with which they were satisfied. Using Clamp 2.0’s logging feature, we timestamped and recorded each operation performed by participants. During the analysis session, participants were asked to think aloud and comment on the reasons behind their choices, which we recorded using an audio recorder. Participants could conclude the session and stop the timer whenever they felt they had obtained a satisfactory result. At the end of the analysis session, participants were asked to verbally describe the clusters in their clustering solution, based on the insights (derived from their analysis). They also completed a post-questionnaire, where they were asked to answer the following questions: Q1) “Are you satisfied with the results or insights you obtained?” Q2) “Would you be able to obtain a better result with another tool or your own coding skills?” Q3) “Did naming clusters help you reason on their significance?” and Q4) “Did Clamp 2.0 help you in deciding the clustering parameters?”

5.1 Results We summarize results of our study in Table 1, reporting for each user their archetype and their eventual expertise in the We find that clusters identified by different participants are in line with recent work on phenotypes in Parkinson’s disease [12, 26]. We below discuss the results of our user study in depth, where we use a classification to refer to participant number i. When meaningful, we also report how each archetype had a different type of interaction with the tool or a different approach to the analysis.

User Archetypes and Domain Knowledge Our study confirmed indeed the relationship between the analyst types and attitude in performing data analysis [17]. For instance, Fig. 9 shows that, on average, hackers seem to invest more time in the analysis and to try out more parameter combinations than other data analysts. Similarly, expertise in the neuroscience domain suggests shorter analysis time, possibly due to better knowledge of the data features. In Fig. 10 we break down the interactive parameters changed by participants during their analysis session into sub-categories (e.g., how many times they enabled/disabled a feature, how many times they changed the number of clusters). We find that different archetypes tended to use differently the features available in Clamp 2.0. Even the type of algorithms and methods used seem to be correlated to analyst archetypes, as shown in Table 1. Based on the participants’ answers to Q1 and Q2, we demonstrate that Clamp 2.0 proved to support the analysis of all types of data analysis.

Analysis Flow For all participants, the analysis started with a default Clustering View automatically applying PCA and Agglomerative clustering to the data. The first action performed by five out of twelve users was to select the features of interest in the data, using either the Data table or the Help me decide panel. Most domain experts removed the non-UPDRS features directly from the Data table, whereas participants without prior knowledge often identified them through the
feature selection tab of the Help me decide panel. Five other participants preferred instead to try out first different clustering algorithms and number of clusters, observing the changes in the scatterplot and in the heatmap. These users generally later noticed the high influence of the non-UPDR features such as PD_MED_DOSE and HOEN & YAIR by primarily help of the heatmap visualization. Consequently, they proceeded in a fashion similar to the domain experts, excluding these features from their subsequent analysis. Finally, three out of twelve users (of which two belonged in the application user archetype) preferred to start their analysis with the Clustering Tour. In most cases, the analysis continued with an iterative and cyclic modification of clustering parameters and selected features, until participants realized that they could only find clustering outcomes based on affected side or severity of the disease. These clusters were easily interpreted from the heatmap visualization, which would show an horizontal gradient for increasing severity and an alternate pattern in rows corresponding to left and right related features.

5.2 Discussion

The Importance of Feature Selection

The first insight identifiable from the final clustering outcomes in Table 1 and from the histogram in Fig. 10 is the relevance of feature selection in clustering analysis. More than any other parameter, the choice of the features to feed to the clustering algorithm was able to lead users towards a satisfactory result, and at the same time was the part of the analysis in which participants spent most of their time on. In particular, participants used the feature distribution information available from the Data Table in combination with the statistical analysis methods available in the “Help me decide” panel. Whereas domain experts were often able to spot uninteresting features from their name (e.g., non-UPDRS features such as ON_OFF_STATE and PD_MED_USE) and directly remove them from the data table, participants with no prior knowledge about the domain made heavy use of principal component analysis (PCA) and univariate feature selection (e.g., ANOVA) to test the relevance of data dimensions. This allowed the scripters in particular to quickly spot features that were contributing the most to the clustering outcome, and eventually remove them from the analysis. The hacker archetype often complemented these findings by inspecting the distribution values (e.g., variance) and pairwise correlations of each feature from the Data Table. The application users seemed instead to prefer identifying relevant features and correlations from the horizontal color distribution of cells in the heatmap, expressing a more qualitative approach. After removing a first set of features, participants generally applied different clustering parameters until they realized a second round of feature selection was needed. Here the most used method was feature agglomeration, with which participants tried to agglomerate features based on correlation or semantics (e.g., removing features with high pairwise correlation, keeping only one feature out of four for tremor, keeping the feature with the highest variance for each left-right pair).

Clustering Tour: Exploring Different Perspectives

While most participants were eager to adopt well-tested clustering parameters and algorithms, metrics, leaving their comfort zone. “I only pushed a button and it already gave me insights I would probably not have found by trying out parameter combinations myself”, commented P9, P10, belonging to the hacker archetype, stated “I generally hate automated features that allow script kiddies to do result shopping [i.e. blindly use system-generated results]. However, Clustrophile 2 gives me the possibility to decide myself if solutions are reasonable and useful for the particular dataset, and at the Clustering Tour was initially used in unconstrained form, whereas the remaining users decided set first the desired number of clusters and a custom set of input features. The average number of solutions generated before a participant expressed a positive feedback is 3.7, followed by an average of 2.3 iteration in the solution neighborhood. In particular, the Clustering Tour proved to be useful to remove non-relevant features and selected features, until participants realized that they could only find clustering outcomes based on affected side or severity of the disease. These clusters were easily interpreted from the heatmap visualization, which would show an horizontal gradient for increasing severity and an alternate pattern in rows corresponding to left and right related features.

(In)Effectiveness of Validation Measures

The results of our user study show how validation measures do not perform well in presence of specific goal-oriented questions that go beyond pure exploration of the data. While most participants did not consider at all the use of validation measures, four participants made use of the “Help me decide” and of the “Is this a good clustering?” panels to try to compare measures among different number of clusters and across clustering instances. However, theoretical best cluster separation often suggested to consider two or three clusters, less than what we would generally expect while searching for phenotypes. In most cases, changing clustering parameters according to validation measures generally produced clustering outcomes with a different and possibly less interesting semantic meaning. P2 commented “I think it makes more sense to see first if the clusters are interesting from the heatmap, and then simply check if they are not too overlapping in the scatterplot”. We believe validation measures are effective in comparing clustering results only when the latter are not too semantically different from each other (i.e. low ΔI). In particular, we use validation metrics to filter and rank the solutions automatically generated by our Clustering Tour after the user has expressed positive feedback (i.e. when most influential parameters have been fixed). Separately, we use them to select the most suited projection to visualize a clustering result. Cluster Naming and Interpretability

According to the answers to Q3, seven out of twelve participants stated that having to verbally describe and name clusters through the tool interface significantly helped them better reason about the clustering instance they found. “I personally never did it, but giving a name to a cluster forces you to go beyond the superficial instinct of finding well separated groups. It often makes you realize you are not searching for the right things”, commented P3. Ten participants named their final clusters only by interpreting the colors of each column in the heatmap, whereas two of them complemented this information with the automated cluster descriptions in the “Is this a good clustering?” panel. This proves how the heatmap visualization can be powerful and self-descriptive, yet simple, to semantically summarize clustering results. Probably in relation to this and to the fact cluster members changed often during the analysis, only one participant used the cluster naming functionality before being required to provide the final clustering solution. Naming clusters automatically based on the centroid also did not generalize to this dataset, where data points were named based on the subject’s numerical identifier. The automatically generated textual descriptions for clusters that we introduced in this paper are not fit for systematically assigning short, meaningful names to clusters. However, a possible solution could be to generate cluster identifiers semi-automatically by incorporating user feedback on which are his main features of interest and how they are semantically related to each other.

6 Conclusion

We present Clustrophile 2, a new interactive tool that guides users in exploratory clustering analysis, adapts user feedback to improve user guidance, facilitates the interpretation of clusters, and helps quickly reason about differences between clusterings. Clustrophile 2 introduces the Clustering Tour to assist users in efficiently navigating the large space of possible clustering instances. We evaluate Clustrophile 2 through a user study with 12 data scientists in exploring clusters in a dataset of Parkinson’s disease patients.

Our work here confirms that clustering analysis is a nontrivial process, which requires iterative experimentation over different clustering parameters and algorithms as well as data attributes and instances. We find that, despite different users exhibit different attitudes towards exploratory analysis, feature selection is where they spend most of their effort. We also find precomputation to be essential for supporting interactive analysis. Fig. 10 shows in fact that dynamically changing the number of clusters was a frequent interaction performed by participants. Given the number of repeated operations, caching also proved to be essential. While the relevance of user assumptions and prior knowledge on the data further confirmed that clustering cannot be automated without incorporating these concerns, participants showed a tendency to stick to well-known parameter combinations or blindly reusing previous parameter settings in trial and error. This way, the system can come into play and assist the user in making parameters more explainable or in comparing alternative choices through the support of statistical analysis. The Clustering Tour we introduce here demonstrates how to nudge users to think outside the box in exploratory clustering analysis, avoiding premature fixation to certain attributes or algorithmic and parametric choices. Finally, the ability to compare clusterings through validation metrics does not necessarily improve their
interpretable. To this end, we urge the development of clustering tools that facilitate explainability, while keeping in mind that usefulness of a clustering outcome mostly depends on the underlying data and user task.

References


