Visual Exploration of Relationships and Structure in Low-Dimensional Embeddings

Klaus Eckelt, Andreas Hinterreiter, Patrick Adelberger, Conny Walchshofer, Vaishali Dhanoa, Christina Humer, Moritz Heckmann, Christian Steinparz, and Marc Streit

Abstract—In this work, we propose an interactive visual approach for the exploration and formation of structural relationships in embeddings of high-dimensional data. These structural relationships, such as item sequences, associations of items with groups, and hierarchies between groups of items, are defining properties of many real-world datasets. Nevertheless, most existing methods for the visual exploration of embeddings treat these structures as second-class citizens or do not take them into account at all. In our proposed analysis workflow, users explore enriched scatterplots of the embedding, in which relationships between items and/or groups are visually highlighted. The original high-dimensional data for single items, groups of items, or differences between connected items and groups are accessible through additional summary visualizations. We carefully tailored these summary and difference visualizations to the various data types and semantic contexts. During their exploratory analysis, users can externalize their insights by setting up additional groups and relationships between items and/or groups. We demonstrate the utility and potential impact of our approach by means of two use cases and multiple examples from various domains.

Index Terms—Dimensionality reduction, projection, visual analytics, layout enrichment, aggregation, comparison.

1 Introduction

Multivariate datasets are ubiquitous. The challenge of making high-dimensional data accessible for visualizations in a two-dimensional space is typically addressed by dimensionality reduction (DR). A plethora of powerful visualization and interaction methods have been proposed for interpreting and exploring scatterplots of dimensionally reduced data, also referred to as *embeddings* [1]. However, most existing approaches do not take a defining characteristic of many real-world datasets into account: *structural relationships* between items and groups of items. Item-to-item relationships, for instance, can result from an inherent (temporal) ordering of data items. Item-to-group associations can be based on shared categorical values or user-defined group labels. Group-to-group relationships are defining properties in hierarchical datasets.

To effectively analyze such structures in scatterplots, users need to be able to relate visual patterns to the underlying structure and high-dimensional data. This is complicated by a general drawback of DR techniques—embedding the data in a space with reduced degrees of freedom naturally introduces distortions. Even data analysts and machine learning engineers—who may know the underlying principles—have difficulties in analyzing these distorted spaces [2], [3]. Nonato and Aupetit [2] pointed out that as a result of these complications, many analytic tasks cannot be performed with the embedding scatterplots alone. Layout enrichment, such as coloring items by an attribute, is necessary to visually convey information about the original high-dimensional data and enable the most common tasks when working with embeddings: analyzing point clusters, representing them as groups, and mapping high-dimensional data to the embedding [2], [3].

In this paper, we propose an interactive visual exploration workflow of embedding scatterplots that treats structural relationships as first-class citizens. We represent the structures directly within an enriched scatterplot layout. Additional interaction methods and summary visualizations let users relate the structures to the underlying high-dimensional data. Users can define groups in datasets on the fly, compare groups, and introduce new relationships between them. Summary visualizations show the high-dimensional data for the groups, while difference visualizations show how groups are different from each other. We designed our approach to be independent of both the DR technique and the application domain.

Our **primary contribution** is an interactive visual exploration approach for scatterplots of low-dimensionally embedded, multivariate data, augmented with structural information about the dataset and an implementation thereof. As **secondary contributions**, we (*i*) discuss important design considerations for summary visualizations that grant users access to the high-dimensional data for items, groups of items, and differences between groups; (*ii*) elaborate on how users can explore an embedding using these summary and difference visualizations to find similarities and differences between items and groups and to form new structures and relationships; and (*iii*) provide use cases from various domains that demonstrate the utility of our approach.

The paper is structured as follows. In Section 2, we introduce our terminology for possible structures in datasets and discuss related analysis tasks. In Section 3, we summarize related visual exploration approaches and discuss important literature that inspired our design for the summary visualizations. The design of the various visual components

- *All authors are with the Johannes Kepler University Linz. E-mail: firstname.lastname@jku.at*
- *Vaishali Dhanoa is also with Pro2Future GmbH. E-mail: vaishali.dhanoa@pro2future.at.*

Manuscript received MONTH DAY, 2021.

is described in Section 4, followed by a discussion of the workflow in Section 5. Section 6 briefly describes how we implemented our approach as part of the *Projection Space Explorer* web application. We present two use cases in Section 7 and discuss remaining challenges for future work in Section 8.

2 Data & Tasks

In this section, we discuss the different types of structures and relationships that we considered for the design of our analysis workflow. From each of these structures, low-level analysis tasks can be derived.

We call the individual entities in a dataset *items* and their properties *attributes* [4]. For this work, we consider numerical and categorical attributes. Embeddings are calculated from a set of high-dimensional data items based on a subset of the available attributes, which we term *projected attributes*. The remaining *meta*-attributes are not reflected in the embedding positions, but might still be relevant for the interpretation and exploration, typically through a form of layout enrichment.

The high-dimensional data items can be structured in different ways. In the simplest case, the items form a *flat* set with no structural relationship between them (see “Flat” in Figure 1). Tasks related to such flat sets—e.g., identifying clusters or outliers—can be performed in non-enriched scatterplots. If one of the attributes of the data items is strictly ordered, individual items can be connected to form *sequences* (see “Sequence” in Figure 1). Multivariate time series are the prime examples of datasets that exhibit such sequential *item-to-item relationships*. Sequentially connected items in scatterplots are often referred to as *paths* or *trajectories* and are known as *Time Curves* in the context of multidimensional projection. Bach et al. [5] describe the possible patterns emerging from Time Curves, and how these patterns relate to different analysis tasks.

We use the term *groups* to refer to collections of items with some shared property. Groups can be defined in a variety of ways (see “Groups” in Figure 1). They can be based on the values of a categorical attribute, in which case they are typically represented in scatterplots with categorical color coding. Similarly, groups can be defined by numeric value ranges. Users can also set up groups themselves based on selections made either directly in the embedded scatterplot or in a different representation of the data (e.g., a tabular view).

Clusters are special types of groups defined via the spatial proximity of items (either in the embedding space or in the high-dimensional data space), typically by means of some density- or neighborhood-based clustering algorithm.

If a single split of the dataset into groups is based on attributes or the result of a typical clustering algorithm, each item will be associated with at most one group. However, more generally, items can belong to multiple groups. Such an independent splitting into groups can give rise to multi-partitioning and hierarchical group structures (see “Groups—Selection in Embedding” and “Related Groups—Hierarchy” in Figure 1). Exploratory data analysis often requires the creation of hierarchical relationships on the fly dependent on a user’s needs. In this scenario, the order of the levels in the hierarchy can be arranged in any way. *Group-to-group relationships* can also be sequential (see “Related Groups—Sequence” in Figure 1). This is the case, for instance, when a chain of clusters forms a trajectory bundle [6]. The most important tasks related to groups are the exploration of *item-to-group associations* (in both directions) and the estimation of data distributions within and across groups.

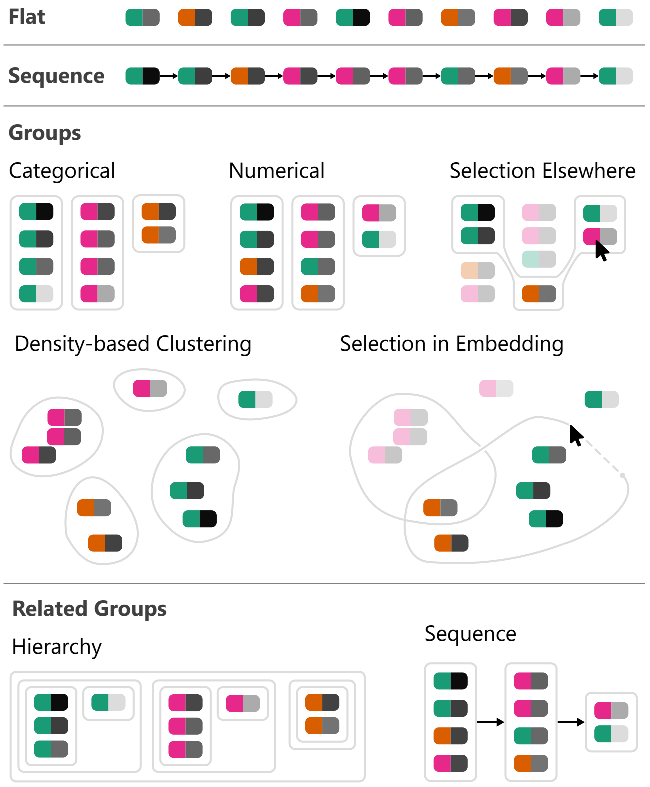


Fig. 1. Different types of structure in datasets considered for the design of our visual exploration workflow. The two halves of each “pill” represent the values of an items’ categorical (left) and numerical attribute (right).

As outlined in Section 4, we address the item-to-item, item-to-group, and group-to-group exploration tasks with layout enrichment, summary and difference visualizations, and by letting users freely connect groups in the embedding.

3 Related Work

In this section, we first discuss works related to the interactive exploration of embedding spaces. We then briefly describe the role of supervised, hierarchical DR techniques and network embeddings in the context of our work. Finally, our use of visual summaries for representing groups of items motivates a brief discussion of relevant aggregation/ summary visualization literature along with some related applications.

3.1 Exploration of Embedding Spaces

In their 2019 survey paper on multidimensional projections in visual analytics, Nonato and Aupetit [2] discuss the need to enrich embedding scatterplots to let users work around distortions introduced by the projection. Layout enrichment—combined with carefully chosen interaction techniques—lets users perform exploration tasks and relate visual clusters to "true" high-dimensional ones. This combined approach has been used for the interactive exploration of embeddings in a number of works.

An early example of enabling such an interactive analysis of embeddings is the Projection Explorer by Paulovich et al. [7], which features various labeling and encoding options. ProxiLens by Heulot et al. [8] allows within-cluster and between-cluster analysis by automatically moving false low-dimensional neighbors to the border of an interactive lens. Liao et al. [9] use abstract glyphs in combination with a tabular view to enable a cluster-focused analysis of multivariate scatterplots. The t-viSNE technique by Chatzimparmpas et al. [10] allows users to explore and interpret *t*-SNE scatterplots in a dashboard that shows information related to the preservation of neighborhoods or pairwise distances. Xia et al. [11] introduce a technique for finding clusters and outliers in embeddings based on successive projections that maximize a marked pattern's saliency. Other tools focus on the interactive comparison of embeddings, based on dissimilarity matrices [12], linking clusters between embeddings [13], or visualizations of neighborhood overlap [14].

To find similarities and differences between groups of a dataset, Fujiwara et al. [15] describe a technique to interactively adjust embeddings by moving or scaling group representations and showing the impact of attributes on the embeddings' axes, carrying on from previous work on interactive embeddings [16], [17]. Ma and Maciejewski [18] describe the analysis of class separations in embeddings through locally linear segments, which connects the work to other recent efforts to explain non-linear embeddings [19], [20], [21]. The Latent Space Cartography technique by Liu et al. [22] lets users explore embeddings of an autoencoder's latent space in multiple coordinated views with enriched scatterplots. Users can define groups through selection or meta-data and create pairwise relationships between them.

Perhaps most closely related to our work are the Probing Projections technique by Stahnke et al. [23] and the Non-Linear Embeddings Surveyor by Sohns et al. [24]. Probing Projections supports a variety of tasks related to cluster and distortion analysis based on four types of layout enrichment: (*i*) histograms show the value distribution of numeric high-dimensional attributes; (*ii*) glyphs indicate whether distances in the vicinity of points are exaggerated or reduced; (*iii*) a value heatmap lets users gauge the relation between directions in the embedding space and high-dimensional attribute values; and (*iv*) overlaid dendrograms indicate clustering hierarchies of items in the high-dimensional space. In Probing Projections, groups of points can be defined by automatic clustering or based on user selections, and small multiples of the groups are created on the fly in the side panel (similar to our summary and difference visualizations described in Section 4.3). In contrast, the Non-Linear Embeddings Surveyor creates groups by binning the data. Each group is represented by a colored non-convex hull, which may get split into multiple smaller areas if the point clouds are far apart. Small multiples show histograms of the individual attributes and how the binned data points are distributed in the embedding.

In most of these works, interactions and layout enrichment are used to let users analyze the distortions and the embeddings themselves, rather than focus on the data that is embedded (and some approaches are limited to specific DR techniques). More importantly, only simple item-to-group associations (such as those derived from categorical attributes or straightforward clustering) are represented visually. In contrast, our approach allows analysis of different types of relationships (see Section 2) and is agnostic to both the DR technique and the application domain. As such, it is an extension of our previous work ProjectionPathExplorer [6], which focused on the analysis of collections of Time Curves [5] (i.e., a combination of categorical grouping with sequential item-to-item relationships).

3.2 Hierarchical Embeddings

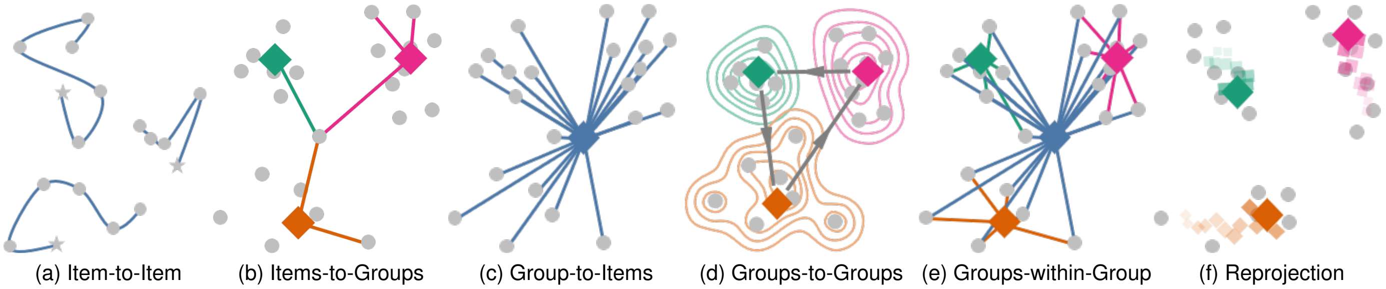
Analysis of the hierarchical properties of datasets in the context of DR cannot only be supported through a post-hoc interaction with existing scatterplots, as described above. It is also possible to consider the hierarchical information already during the calculation of the embedding. Notable techniques include Hierarchical SNE [25], Tree-SNE [26], and Haisu [27]. These approaches are members of the broader family of supervised DR techniques [28].

While these supervised, hierarchical DR techniques are certainly related to our approach, we see them as orthogonal to the interaction- and enrichment-focused analysis approach, which is at the core of our contribution. In fact, since our approach is agnostic to the type of embedding used, the results of supervised DR techniques can be analyzed with our technique. However, as discussed by Höllt et al. [29], additional Focus + Context exploration techniques may be required for an effective analysis of such hierarchical embeddings for large datasets.

3.3 Network Embeddings

Network embeddings map nodes to low-dimensional representations while preserving the network structure and properties in the low-dimensional space [30]. Yan et al. [1] showed that all common DR techniques can be phrased in this way if an intrinsic graph is created that represents certain aspects of the high dimensional dataset. Exploring and analyzing network embeddings has been the subject of several previous works, paying attention to the special requirements of network embeddings. In EmbeddingVis [31], users are offered multiple embeddings for comparison and to examine which properties the embeddings value. Comparison of embeddings is also the focus of the work by Heimerl et al. [14] and Boggust et al. [32], in which they compare two embeddings and their local neighborhoods. GNNVIS by Jin et al. [33] lets users analyze graph neural networks and their prediction results with multiple views that summarize node-level metrics, structure, and data.

Similar to the hierarchical embeddings discussed above, network embedding methods are related to our approach and provide an opportunity to analyze already structured data with our technique. In contrast to the aforementioned works, however, we focus on exploring and comparing the data that is embedded and let users form the structure based on their insights. A subsequent analysis of the structured data is, of course, possible with the above-mentioned tools.



(a) Item-to-Item (b) Items-to-Groups (c) Group-to-Items (d) Groups-to-Groups (e) Groups-within-Group (f) Reprojection

Fig. 2. Supported visual encodings to explore and identify relationships. (a) Related items are connected to form trajectories. (b) An item's association to groups is indicated by connecting it to the group centroids (◆, ◆, ◆). (c) Likewise, items associated with a selected group are indicated through lines from the group centroid (◆). (d) To encode related groups, directed edges are drawn between their centroids (◆ → ◆ → ◆); here, group contours are shown additionally. (e) Hierarchical structures lead to tree-like representations (◆ is connected to items of ◆, ◆, and ◆). (f) Changing centroid positions during and after reprojection are indicated by a fading trail of centroid marks (◆◆◆, ◆◆◆, and ◆◆◆).

3.4 Summary Visualizations

Our technique makes use of visual summaries of the high-dimensional data of groups of items in the embedding, which facilitate the mapping between low and high dimensional data [2], [3]. There are several related examples in the literature of combining summary visualizations with embeddings. In DICON [34], Voronoi-diagrams are used as glyphs to summarize clusters. Visualnostics [35], introduced by Lehmann et al., are pictograms that guide users through projections of high-dimensional data. Joia et al. [36] place textual representations of the most important attributes inside the concave hulls of clusters. Liao et al. [9] use radial glyphs as abstract visual summaries for clusters in multivariate scatterplots. Marcílio et al. [37] propose the use of star plots as visual summaries for the analysis of feature spaces. Jo et al. [38] provide a grammar to encode categorical data in scatterplots with multiple scalable designs. The Probing Projections technique [23], as explained above, uses histograms to summarize the high-dimensional value distributions of clusters. Some of these works also improve readability of cluttered scatterplots [9], [34], [36], [38].

We introduce the concept of *summary visualizations* to generalize such specific visual summaries. In Section 4.3, we discuss how the design of these summary visualizations depends on the data types, the supported tasks as well as the semantic context. This discussion draws heavily from previous work on visual aggregates by Elmqvist and Fekete [39], and from a more recent paper on the design factors for summary visualizations by Sarikaya et al. [40].

4 Visualization of Relationships and Structure in Embeddings

In this work, we enrich the embedding scatterplot with meta-data, structures, and relationships and provide summary and difference visualizations so that users can relate the data.

Based on the types of structure identified in Section 2, we have designed representations for groups and relationships between items and/or groups (see Figure 2). We use multiple layers to display the items, groups, and their relationships to avoid occlusion of the data in focus and improve readability: (*i*) unselected items and their relationships are at the lowest level, followed by their selected equivalents; (*ii*) groups, their relationships, and contours follow on top of the items in the same manner since items belong to groups and there are fewer groups than items; and (*iii*) the top layer is used to visualize selections.

In the following, we explain and justify the visual encoding choices. The workflow and interaction methods are described in Section 5.

4.1 Representing Items and Their Relationships

Items are represented as marks in the scatterplot. The position of these items can either be based on predefined coordinates (e.g., if users inspect a dataset, for which a DR was already applied externally) or based on the results of a DR algorithm. Color (hue and opacity), shape, and size channels can be used to encode additional data. The shape encoding only supports categorical data, while opacity and size require numerical data. The color scheme of the hue encoding automatically adapts to the attribute type.

We encode sequential item-to-item relationships by connecting item marks with Catmull–Rom splines (see Figure 2a). We have seen in previous work that the use of curves over straight lines improves readability [6]. Catmull–Rom splines do not require any control points to be defined [41], [42]. The color channel of the resulting trajectories can encode an attribute. In Figure 6, the states of the individual chess games are connected in order to follow the course of the game. The shape channel of the item marks can also be set to encode the initial, intermediate, and final states along a trajectory.

Apart from more advanced options for the different visual channels, the encodings described so far are mostly identical to those discussed in our previous work focusing on collections of trajectories [6]. The following subsections pertain to new features introduced for working with other types of structures, such as item-to-group associations and group-to-group relationships.

4.2 Representing Groups and Their Relationships

The base encoding for a group of items is the group's *centroid*. The centroid is represented by a diamond mark (◆) and differs in shape and size from the item marks in the embedding, see Figure 2. While we do not provide means to encode data in the group marks, their size scales with the item marks minimum size. Additionally, we use color to differentiate between selected (blue ◆) and unselected (gray ◆) groups. The position of the centroid is determined by averaging the low-dimensional coordinates of the embedded items. If the data is reprojected, group trails can be displayed that show how the group centroids change their position in the embedding (as illustrated in Figure 2f). Group trails show centroid positions for the last 50 iteration steps with decreasing opacity.

We also considered medoids, representative items of a group, and centroids based on the high-dimensional data as alternative group representations. However, we decided against these alternatives for two reasons. First, the high-dimensional centroids would require adapted DR techniques that provide an out-of-sample extension, to display interactively defined groups. Second, we observed that embedding the medoid or high-dimensional centroid together with the items causes it to almost always lie within a visible cluster. For groups with multimodal point distributions in the embedding space, this led to unintuitive results, especially in hierarchies where parent and child nodes coincided, while the low-dimensional centroids are well separated.

The group centroids are the foundation of our encoding for item-to-group and group-to-item associations. Depending on the analysis task and the complexity of the groupings, we propose different options for encoding item-to-group and group-to-item associations. By default, these associations are shown as lines connecting a centroid's group and an item, following the Gestalt grouping principle of connectedness (see Figures 2b and 2c). We opted for this encoding because it can be used bidirectionally and also works in cases, where items can belong to multiple groups. Additionally, it leads to intuitive visual representations of simple hierarchies (see Figure 2e), even without explicitly encoding group-to-group relationships.

Users can optionally switch to a contour representation of the group-to-item associations, as shown in Figure 2d, which is based on the grouping principle of common regions. The contours are calculated using kernel density estimation, with a default, global kernel bandwidth set to one-tenth of the total range spanned by the embedded points. For each group, contours for 10 evenly spaced density thresholds are drawn. Initially, we experimented with concave and convex hulls instead of contours, but we found that contours enable users to better judge spatial distributions of groups [43]. Additionally, contours are less prone to be distorted by outliers.

We encode group-to-group relationships as directed edges between group centroids. We have decided to use straight lines with small arrows to represent group-to-group relationships to clearly distinguish the curves of item-to-item relationships. Figure 2d shows a simple network of groups with three edges.

4.3 Summary and Difference Visualizations

We differentiate between two visualizations through which users can access the high-dimensional data for items or groups. *Summary* visualizations show a single item's or a single group's characteristics in a compact manner. *Difference* visualizations reveal (dis-)similarities between two items or two groups.

Inspired by enRoute [44] and Probing Projections [23], we position the summary visualizations in a juxtaposed view, as other positioning choices, such as integrated, superimposed, overloaded, or nested views, would suffer from occlusion problems [45]. Additionally, the placement in a separate view requires no additional layout modifications to allow further analysis of multiple items or groups. Summary visualizations are shown in the Details tab of the menu pane, and in the comparison pane (see Figure 6).

The Details tab displays a summary visualization for the data selected and allows the selection of displayed attributes. The comparison pane consists of a three-column layout. The first column contains a compact vertical representation of all sequences (i.e., branches) that go through the selected item/group. The second column shows the summary visualization for each item/group selected. The third column of the side panel displays the differences that exist between two consecutive items/groups.

We now discuss the design of our generic summary and difference visualizations first and then elaborate on domain-specific variants.

4.3.1 Generic Visualizations for High-Dimensional Data

A generic approach of **summarizing attributes for a single item** is a tabular layout as outlined in Figure 3 (high-dimensional data). Similar to Probing Projections [23], we encode the values for numerical attributes as vertical lines on top of a density plot that shows the attribute's value distribution for the whole dataset. This lets users relate the characteristics of an item to the overall dataset. For categorical attributes, we simply show the category of the item.

For the default **summary visualization of groups**, we use density plots for numerical attributes, again overlaid with the overall distribution. For categorical attributes, we display each category by the count of items in descending order.

To make the tabular representation scale to datasets with many attributes, *ranking* the attributes is necessary. We rank table rows for continuous attributes by the normalized standard deviation (lowest first), and categorical attributes by the relative frequency of the largest category (highest first). This ranking lets users quickly identify the most or least relevant attribute depending on their analysis goal.

The **difference visualization of items/groups** uses diverging bar charts for categorical attributes and box plots for numerical ones. For the respective attribute, the diverging bar charts show the relative change in distribution, while the box plots show the distributions of the two items/groups. We made the difference visualizations self-contained to reduce back and forth comparison between the summary visualizations. The visualizations are generic to use for the items and groups comparisons. Additionally, we rank the attributes according to their change between the items/groups—from most to least.

4.3.2 Domain-Specific Visualizations

While we chose the above encodings as default, we have found that many specific application domains have more natural ways to encode the high-dimensional single items visually. For instance, in the case of the (spatially fixed)

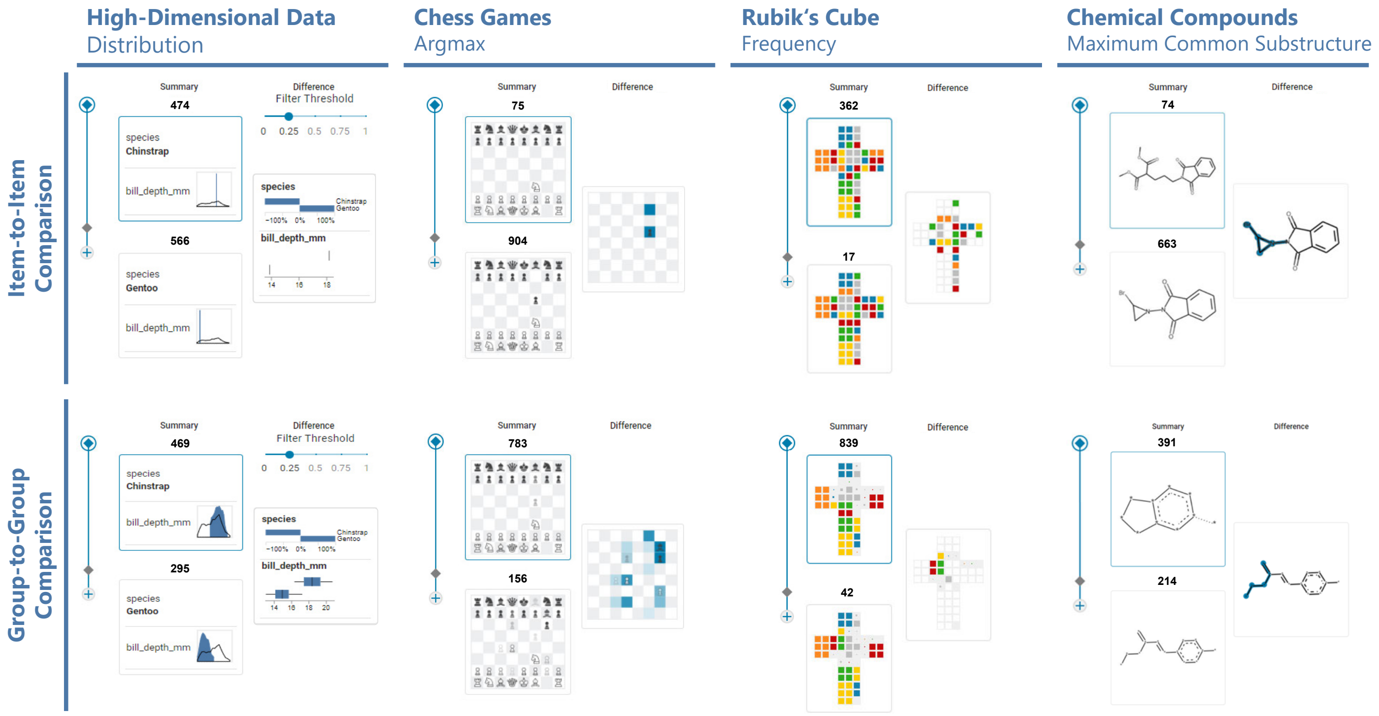


Fig. 3. Encodings for the summary and difference visualizations for a general tabular approach using high-dimensional datasets, chess games using the argument of the maxima, a Rubik's cube based on the relative frequency of occurrence, and chemical compounds using the maximum common substructure. At the top, we show the visual encoding for item-to-item and at the bottom for group-to-group comparison tasks.

categorical attributes that describe a Rubik's cube state, the folded-out cube is a straightforward representation (see Figure 3) [6]. For chess, the chessboard can be shown directly. In these cases, we found that a group visualization derived from the single-item case is more intuitive than the default tabular one. However, unlike in the default case, where the full histograms can be used as an aggregation for groups, the additional design constraints in these special cases require further aggregation.

Data can be aggregated by different aggregation functions. We found that aggregation by most common value (i.e., the *argmax* of a discrete distribution) together with its frequency/count works well for categorical data with additional local constraints. Examples of such constraints, which fix the placement of the mark's encoding the categorical values, are the Rubik's cube "cubies", or the squares of a chessboard. This argmax aggregation visually highlights the most frequently occurring category within an item group.

In the case of Rubik's cube, we determine the color of a "cubie" by this argmax aggregation and encode the value's frequency with the size of the rectangle (see Rubik's Cube in Figure 3). In the case of chess games, the argmax determines which chess piece is placed on a square in the summary visualization. The difference visualization for both of these categorical datasets with placement constraints is straightforward in the single-item case. The difference visualizations only display squares and pieces that differ between two items. For the group-to-group comparison in the case of a Rubik's cube, "cubies" that did not change are colored in white and the ones that changed are encoded by the respective color and size of the second state. For the chess games, group-to-group summary visualizations encode the frequency of a piece with opacity.

The concept of summary and difference visualizations can also be adapted to completely different data types from other domains. For example, consider a domain adaptation of our visualization that enables chemists to analyze chemical compounds in an embedding. For molecules, the structural formula—a 2D graph with atoms as nodes and bonds as links—is a well-known representation that can be used directly for single items. The group representation can be derived from the structural formula. We propose to use the maximum common substructure of all selected items in this case (see chemical compounds in Figure 3). For the group difference visualizations, the maximum common substructure between groups can be augmented with the structural parts that differ.

5 Workflow

Our proposed workflow for visually exploring high-dimensional data using low-dimensional embeddings is illustrated schematically in Figure 4. The user interface of our prototype implementation is depicted in Figure 6. The interface is divided into three components: the menu pane, the embedding scatterplot, and the comparison pane.

In our workflow, users first select a dataset and attributes of interest that serve as input for the DR. Our prototype implementation currently supports *t*-SNE [46] and UMAP [47], but other DR algorithms could be added easily, as discussed in Section 3. They then explore the resulting embedding scatterplot. In the early phase of their analysis, layout-enriching encodings directly in the scatterplot engage users to interact with the embedding. Users can start to explore and create groups within the embedding, using the enriched

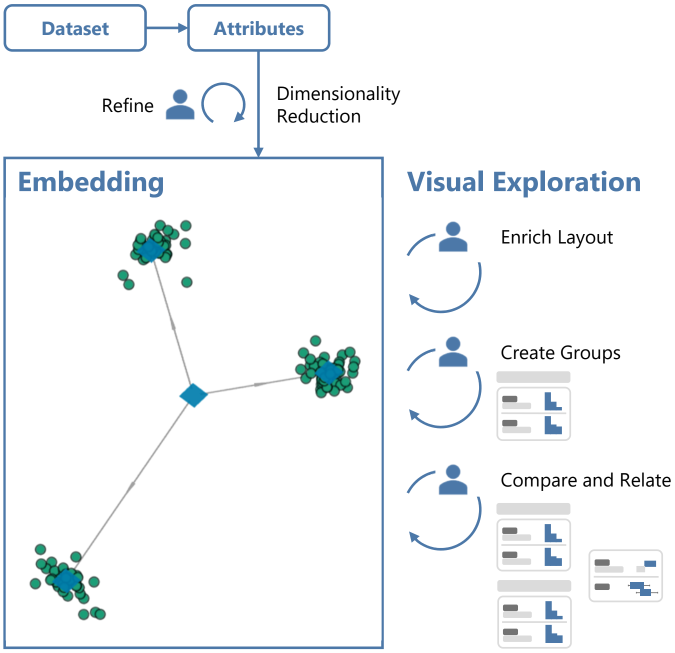


Fig. 4. Schematic workflow of our interactive embedding exploration approach starts by selecting the dataset and the attributes to visualize high-dimensional data in a low-dimensional embedding using dimensionality reduction. The visual exploration of relationships and structures can then be done by layout enrichment using attributes, by the creation of groups, and by comparing items or groups and introducing relationships. This process can be considered iterative.

layout and summary visualizations. Once users have acquainted themselves with the embedding space, they can start to use the difference visualizations to compare and relate the data and characterize group-to-group relationships. During each phase of their analysis, users can go back and use their updated knowledge to modify the projection attributes that should be taken into account by the embedding.

To introduce our approach, we use the Palmer's Penguin dataset [48] as a guiding example throughout this section. The dataset consists of three species of penguins residing on three islands: Biscoe, Dream, and Torgersen. The Adelie species is found on all three islands, whereas the remaining two penguin species—Gentoo and Chinstrap—each inhabit the latter two islands. The dataset additionally includes the sex of the penguins and various numerical body measurements (bill length, bill depth, flipper length, and weight). We load the dataset and use the UMAP DR technique (with a neighborhood of 15 and 300 iterations) to project the data based on the penguins' numerical body measurements (see Figure 5).

In the following sections, we discuss how users can adapt and interact with the visuals described in Section 4.

5.1 Enrich Layout

Marks in embedding scatterplots are initially indistinguishable. The color (hue and opacity), shape, and size channels of item marks can be encoded, through the Encoding tab of the menu pane. Users can enrich the scatterplot with *projected* or *meta*-attributes. As explained in Section 2, meta-attributes are those attributes that have not been used to calculate the embedding. In the penguin embedding shown in Figure 5, the data points are colored by the three species.

Users can also use the shape encoding to distinguish between start, intermediate, and end states along an item-to-item trajectory. The color channel of trajectories can be set to encode an attribute of the users' choice. In Figure 6, the states of the individual chess games are connected in order to follow the course of the game. The item marks and trajectories are colored according to the opening chosen by the player.

Hovering over items shows a visual summary of that item's high-dimensional data, while a lasso selection shows a summary of multiple items' data. The hue encoding is only applied to selected items and their trajectories. Users can select single trajectories to load them into the comparison pane, where consecutive items and their differences can be explored.

5.2 Create Groups

Groups can either be predefined in the dataset or constructed interactively on demand. Users can perform the interactive grouping: (*i*) by selecting items directly within the scatterplot through clicking or using a lasso selection; or (*ii*) by calling an automatic, density-based clustering algorithm (HDBSCAN [49]) that operates on the low-dimensional item coordinates. Groups can be given a label to reference them and maintain orientation after reprojecting the data. Groups are listed by their labels in the Groups tab of the menu pane, and the labels serve as headers for the summary visualizations. Groups can be deleted either via the context menu of the centroids or in the Groups tab of the menu pane.

The summary visualization is shown in the Details tab of the menu pane after selecting a group by clicking its centroid. Selecting a group also selects all items of that group and updates their hue encoding (see Section 5.1). The user can adjust the high-dimensional data displayed in the summary visualizations, which by default only include *projected attributes*. As discussed in Section 4.3, the attributes are ranked to let users quickly identify the most or least relevant attributes.

In our guiding example, the user groups the penguin populations by species. Assisted through summary visual-

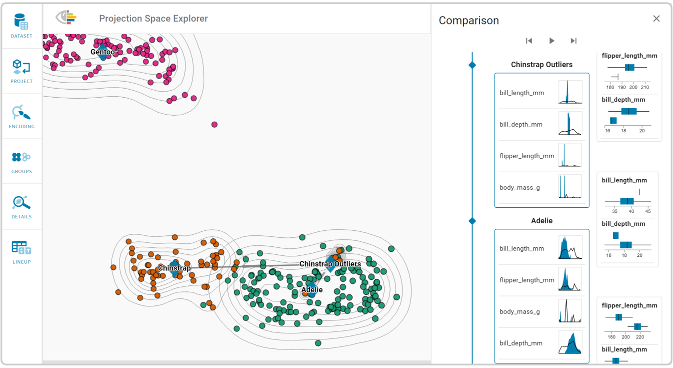


Fig. 5. Penguin embedding showing the three species encoded by color. In the comparison pane, the summary visualizations describe groups by their attributes, and the difference visualizations highlight distinctions between two consecutive groups.

izations and the enriched layout, the user labels the groups. The individual species are well separated within the embedding, apart from a few outliers of Chinstrap penguins, for which the user creates an additional group. For the penguin dataset, the generic summary visualization is used. In Figure 5, the *Adelie* penguin group has the attribute *bill length* ranked first in the summary visualization, as it is the least varying attribute in this group. In contrast, the *bill depth* varies most, making it the last ranked numerical attribute in the summary visualization.

5.3 Compare and Relate

As described in previous sections, summary visualizations are shown when hovering over items and selecting items or groups. After selecting multiple groups, they can be compared in the comparison pane using the context menu of centroids. The comparison pane displays multiple summary visualizations and compares the groups with difference visualizations. For items, difference visualizations are available for points along trajectories. With the stepper controls in the comparison pane, users can go through the data step by step, which also highlights the current item/group in the embedding.

The difference visualization uses the same attributes as the summary visualization and ranks them, as discussed in Section 4.3, to let users quickly identify attributes that vary the most or least between items/groups. We further enhance the information retrieval by adding a *threshold filter* (see top right in the high-dimensional data example in Figure 3) that lets users hide lower-ranked attributes in the difference visualization.

In order to compare groups of penguins, the user selects the groups and loads them into the comparison pane using the context menu of centroids (see Figure 5). When comparing the whole Chinstrap penguin population with the outlier group, the difference visualization shows that the groups differ in the distribution of all body measurements. The *Chinstrap Outliers* group has much more in common with the *Adelie* penguin group, showing only little deviations in the bill length and bill depth (see Figure 5). This confirms the positioning in the embedding and indicates a possible misclassification of the *Chinstrap Outliers*.

After creating and comparing groups, users can introduce edges between groups to set the data in relation. Edges are created by clicking and dragging from one group centroid to another. The resulting structure resembles a graph in which a group or item may have more than one outgoing or incoming link, as shown in Figure 2d.

In our guiding example, we have Chinstrap penguins that are much more similar to Adelie penguins than to their conspecifics. As shown in Figure 5, the user, therefore, connected the *Chinstrap Outliers* group to the group of all Adelie penguins and to the group of all Chinstrap penguins, to indicate these relationships.

Users can save groups and relationships as *sessions* to avoid visual clutter in the embedding space caused by too many connected groups from different analyses.

In summary, our approach allows users to explore existing or newly introduced relationships between items or groups. With the help of our comparison panel, users can analyze these relationships in detail to investigate differences, capture insights, and finally present their findings.

6 Implementation

We integrated our visual exploration approach into the *Projection Space Explorer* library. The library is open-source and available on GitHub: https://github.com/jku-vds-lab/projection-space-explorer.

The prototype is available at https://jku-vds-lab.at/apps/embedding-structure-explorer. The application extends and generalizes the ProjectionPathExplorer described in our earlier work [6].

The web application is written in TypeScript. We use three.js [50] for rendering the embeddings, Vega-Lite [51] for the summary and difference visualizations, D3's contour library [52] for the contour plots, and React [53] for creating the user interface.

Users can load tabular datasets as JSON or CSV files, containing the items and optional structural information. All datasets that are mentioned throughout this paper are preloaded in the prototype.

7 Use Cases

We demonstrate our visual exploration approach by means of two use cases from different domains: (1) the analysis of openings in professional chess games and (2) the analysis of cancer patient cohorts based on genomics data. We chose these two because they cover a broad variety of aspects in terms of data types, structural relationships, and analysis workflows. The chess dataset includes item-to-item relationships and requires custom summary and difference visualizations we have designed in consultation with a professional chess player. The genomics dataset is a real-world high-dimensional dataset, where hierarchical group-to-group relationships are the focus of the analysis.

7.1 Chess Games

The first use case targets the analysis of 450 professional chess games downloaded from the KingBase chess database [54]. As hobby chess players and inspired by the TV series *The Queen's Gambit*, we are eager to learn more about how professional players open their games and what influence openings have on the strategy, progression, and outcome of the games. We demonstrate the workflow and interactions in the supplementary video.

To prepare the dataset, we parsed the raw data files that are provided in the PGN format using the chess module of the pgn2gif Python package [55]. The resulting sequences of chessboard states are encoded in 64 categorical attributes—each representing a square on the chessboard. There are 13 categories in total: six for the different black pieces, six for the different white pieces, and one for empty squares. A chessboard is organized in *ranks* (rows) and *files* (columns) with the identifiers *1* to *8* and *a* to *h*, respectively. The white player's perspective defines the order of these identifiers. Letters go from left to right, and 1 is closest to the player. We additionally added the meta-attribute containing the opening move chosen by the white player.

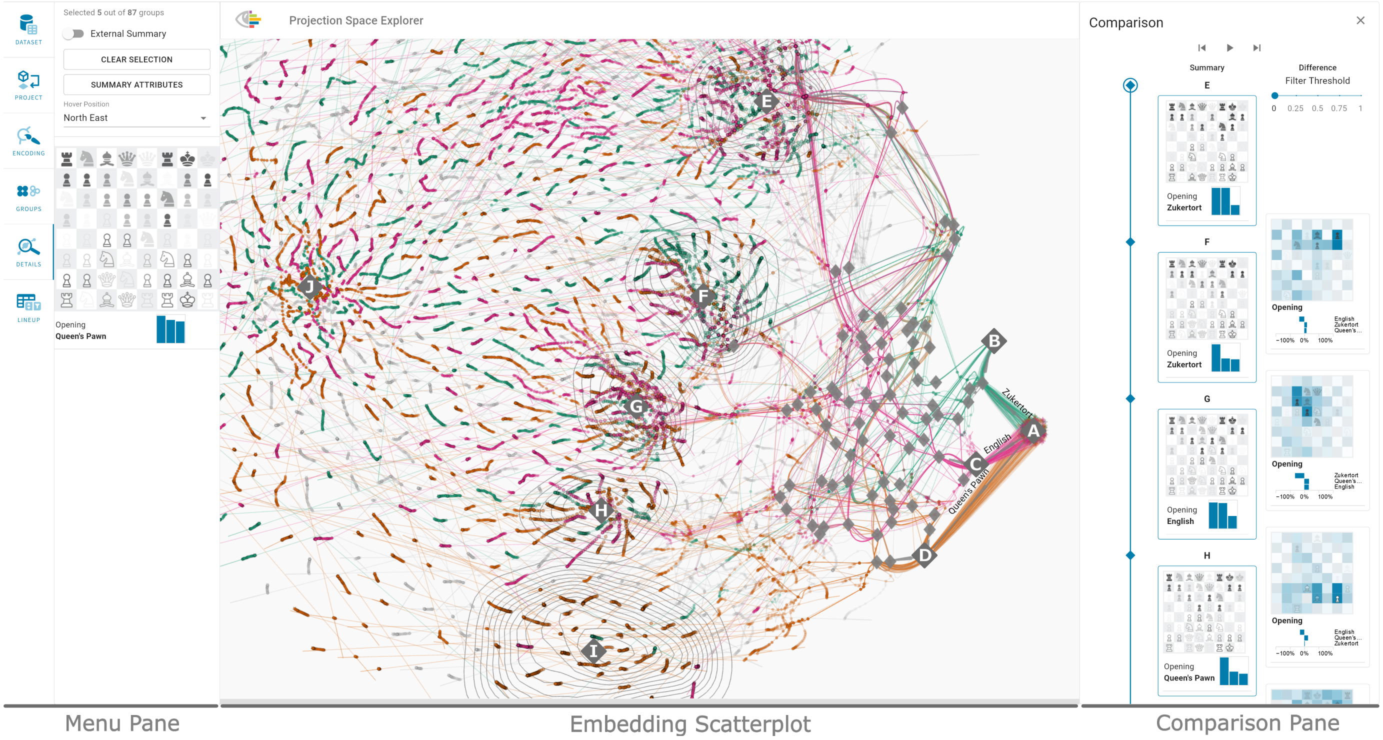


Fig. 6. The menu pane is used to select and project datasets, adapt visual encodings, manage groups of items, and get details on selected items and groups. The embedding scatterplot shows the dimensionality reduced data. Group A represents the start of all chess games, which splits up into groups for the three different openings: Zukertort (B, green), English (C, pink), and Queen's Pawn (D, orange) opening. Groups E–I contain middlegame moves, and group J endgame moves. In the comparison pane, differences between the connected middlegame groups are shown.

As a preprocessing step, we projected the chessboard data using t-SNE with a learning rate of 100 and a perplexity value of 50. Additionally, we added an identifier for each game that corresponds to one opening strategy (the first move by white). Furthermore, we created one group for the starting position and several groups for the most common chessboard states for the first ten moves and labeled them accordingly. To reflect the opening (first move) and the response (second move), we additionally introduce a hierarchy of group-to-group relationships that connect the first two moves sequentially. After loading the preprocessed dataset, the embedding scatterplot shows the approximately 40 000 projected chessboard states during the games (see Figure 6).

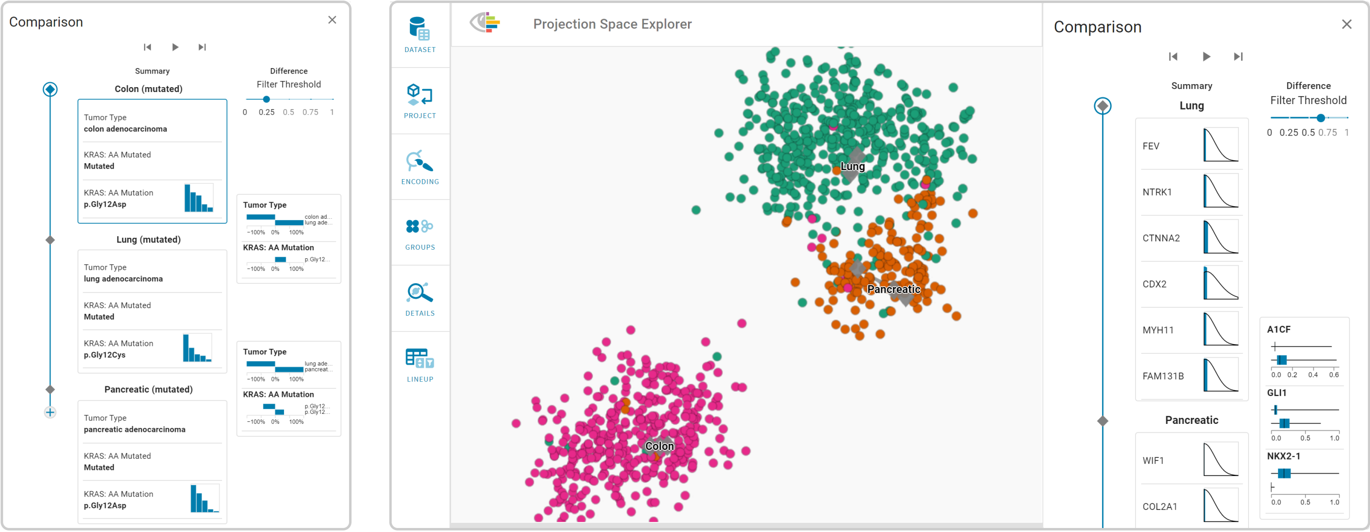
In our previous work [6], we explored the game trajectories and the main clusters of a similar but smaller dataset. We were already able to characterize the embedding space containing the start of the games, endgames with only a few pieces, and the distinct regions of densely threaded sequential states. We discovered that the number of pieces on the chessboard loosely corresponds to the position in the embedding space. Points on the right side tend to represent states with many pieces on the chessboard and points on the left side represent states with only a few. Furthermore, we characterized the main clusters that show up in the embedding. These clusters were the start position, endgames with a few pieces, three predominant openings, and clusters with different pawn positions. The comparison of these middlegame clusters was a tedious process because we had to recall the states of the chessboard or use screenshots in order to detect similarities and differences. The sore points and limitations of our earlier analysis motivated us to think about how the analysis of trajectories in embeddings can be carried out more effectively. The result of this process is presented in this paper.

We continue our analysis by loading the defined groups and relationships. We switch to the *Groups* menu and select the saved session. The games are colored by the three dominant openings chosen by professional players (see Figure 6): B *Zukertort*, C *English*, and D *Queen's Pawn*.

We now want to investigate the different openings by the white player and the response moves by the black player. We make use of the summary visualizations for the different groups and notice that two moves are used in response to all three openings: pawn from f7 to f5 or pawn from e7 to e6. We continue to further look into the groups of the first ten moves and observe that all three openings can lead to the same chessboard state which then splits up again as the game continues.

Next, we take a closer look at the middlegames by focusing on the remaining embedding region left to the predefined groups. As in our previous work [6], we are able to see the *castling* move (king and rook trade places) and the different pawn positions for these clusters, Figure 6 E to G.

To dig deeper, we create a group for each of the three clusters and open the comparison pane to analyze the summary visualizations and compare the groups in more detail. We see that the position of the *kingside* (files *e* to *h* on the chessboard) bishop changed from *g*7to *e*7 from group E



(a) Comparison by gene mutations.

(b) Comparison by gene expression.

Fig. 7. Analysis of a cancer genomics dataset. (a) In the comparison pane, we compare colon, lung, and pancreatic adenocarcinoma groups. Particularly the *Gly12Cys* mutation type changes between the groups. In the summary visualization of the lung group, we see that this is the group's most common mutation type. (b) Items in the embedding space are embedded based on gene expression values and color-coded by tumor type: *colon adenocarcinoma* (pink ●), *lung adenocarcinoma* (green ●), and *pancreatic adenocarcinoma* (orange ●). Here, the comparison pane shows a comparison of lung and pancreatic adenocarcinoma groups, which are located in the same cluster. Despite their adjacent positioning in the embedding, we can observe from the summary visualizations that the gene expression values are partially higher for the lung adenocarcinoma group. The difference visualization highlights that the expression values change most for genes A1CF, GLI1, and NKX2-1.

to group F. However, in group G both queens were moved from their starting position. We can also observe different distributions of the opening moves in the respective clusters. The majority of games in group F were opened with Zukertort opening, while games of groups E were mostly started with a Zukertort or English opening, and games of group G with an English or Queen's Pawn opening.

We extend the list of groups by creating groups H and I. As outlined in Figure 6, we compare all selected groups from the middlegame in the embedding scatterplot. The comparison pane reveals that in the groups from E to H, castling is used by both players on the kingside. Only group I differs, where castling is used by the white player on the *queenside* (files *a* to *d* on the chessboard). We also see a change in the white player's defensive strategy between group G and group H, since the player has placed a bishop in front of the king in these states. Group H and group I consist mainly of the Queen's Pawn opening.

This confirms our expectation that castling moves are essential strategic gameplay elements used in almost every professional game. Which type of castling to use, depends on the opening already, as we can see in the summary visualizations of groups E to H. Furthermore, we can see that the queenside castling is chosen rarely compared to the kingside castling because it is more dangerous and mostly done by white who moves first. As hobby players, we have learned from the analysis that we should use kingside castling, if possible.

7.2 Cohort Analysis

The second use case is motivated by a New York Times article [56] that features a new generation of drugs for cancer treatment, targeting the *KRAS* gene. Mutations of KRAS can lead to uncontrolled cell growth, i.e., cancer. In fact, KRAS mutations are one of the most frequent mutations found in cancer tissue samples. One particular mutation of this gene, *Gly12Cys*, is highly prevalent in lung, colorectal, and pancreatic cancer.

With the new therapy, the cell growth caused by the mutated KRAS gene can be prevented, even causing cancers to shrink. Based on these advances, we want to investigate genomics data from The Cancer Genome Atlas [57] that we previously already worked with [58], [59]. The dataset contains data extracted from tumor samples and includes meta-data such as the age and gender of the patients, gene expression, mutation, and copy number data (745 attributes in total). For the purpose of this use case, we extracted data for the three aforementioned tumor types from the Ordino application [58], resulting in 1 238 tissue samples. The goal of this analysis is to determine the prevalence of KRAS mutations, specifically Gly12Cys. Within the scope of this paper, we also investigate the expression of tumor-related genes and how they relate to different tumor types.

After loading the dataset, we use the UMAP DR technique (with a neighborhood of 15 and 250 iterations) to project the data based on the *KRAS AA Mutated* and *Tumor Type* attributes. *KRAS AA Mutated* indicates whether the sample has a mutated KRAS gene. By selecting all samples in the embedding scatterplot, we can investigate how the samples in our dataset are distributed regarding tumor type and KRAS mutation. The summary visualization shows that about 39.5 % of the tumor samples have a KRAS mutation, 48 % have no mutation, and 12.5 % don't have any information on KRAS mutations included. We color the items by the *Tumor Type* and observe that almost all clusters consist of samples with the same tumor type. To differentiate between mutated and non-mutated samples, we additionally map the *KRAS AA Mutated* attribute to the shape of the items. We remove samples with missing data from the embedding by deselecting the shape encoding for *KRAS AA Mutated* values. The remaining six clusters correspond to the three tumor types, each with and without a KRAS mutation. We apply density-based clustering to the samples in the embedding to have a group created for each cluster and label them via the *Groups* menu. The number of samples in each group is shown in the *Groups* menu, which reveals that the share of KRAS mutation differs depending on the tumor type.

To build up a semantic hierarchy, we create additional groups using a lasso selection, each containing all items of one tumor type, and connect them with the corresponding subgroups. By looking at the position of the tumor type centroids, we can observe whether the majority of samples are KRAS mutated. The centroid of all pancreatic adenocarcinoma samples, for example, is placed close to the centroid of the KRAS mutated subgroup, as 131 out of 174 samples with this tumor also have a KRAS mutation.

Finally, we investigate the specific KRAS mutations in the three tumor types. To this end, we add the *KRAS AA Mutation* attribute to the summary visualization. In addition to the presence of a mutation, *KRAS AA Mutation* also includes information about the specific type of mutation. The mutation type ultimately determines whether a patient responds to a drug or not. We select the groups of the three tumor types that have a mutated KRAS gene, to display them in the comparison pane using the context menu (see Figure 7a). The difference visualization shows that the tumor type changes from one group to the next, which is not surprising considering the groups are based on the tumor types. The more interesting part is the *KRAS AA Mutation* attribute, which shows high dissimilarities between the groups. This is depicted in the difference visualization and also indirectly in the summary visualization. We discover that tumor types differ by the specific cancer-causing KRAS mutations. For the three different tumor types *colon adenocarcinoma*, *lung adenocarcinoma*, and *pancreatic adenocarcinoma*, the most frequent KRAS mutations are *Gly12Asp*, *Gly12Cys*, and *Gly12Asp*, respectively. Less than 5 % of the colon adenocarcinoma samples have a Gly12Cys mutation, even less for pancreatic adenocarcinoma. We can therefore confirm the insights from the New York Times article that patients with lung adenocarcinoma—who are mostly current or former smokers—are the ones that can benefit the most from those new drugs.

As the tumor growth is ultimately triggered by the gene expression, we continue the analysis with the gene expression data and their correlation to the tumor types. We go back to the *Projection* menu and recalculate the UMAP embedding (with a neighborhood of 15 and 250 iterations), but this time including the expression data of all genes (720 attributes). Instead of random initialization, we use the current item positions as seed points. By enabling group trails in the projection menu, we can observe how the group centroids change their position in the embedding through reprojection (as illustrated in Figure 2f). In the newly calculated embedding space, we detect three well-separated clusters. One is almost entirely made up of colon adenocarcinoma samples, one of lung and pancreatic adenocarcinoma samples, and one very small cluster contains a mix of samples of all three tumor types. Using the summary visualizations, we observe that the small heterogeneous cluster has comparably low gene expression values across a large number of genes compared to the expression data in the other two clusters. On closer examination, we discover that the gene expression values are not only low but zero, indicating that these samples had no gene expressions recorded at all.

We continue the analysis by comparing the tumor type groups that we created before. We select the groups of *colon*, *lung*, and *pancreatic adenocarcinoma*, and use the comparison pane to investigate differences between their gene expressions. The spatial separation in the embedding scatterplot already highlights a difference across all genes in the dataset.

Figure 7b shows the embedding scatterplot zoomed in on the two clusters with valid gene expression data. By inspecting the summary visualizations in the comparison pane, we can observe that gene expressions of *lung adenocarcinoma* samples are distributed over a larger range for most of the 720 genes. By increasing the filter threshold from 0.25 to 0.6, we see that the biggest difference of lung and pancreatic adenocarcinoma samples are lower expression values for *A1CF* and *GLI1*, and higher expression values for *NKX2-1* (see Figure 7b). The gene expression of *GLI1* is promoted by a *KRAS* mutation and is involved in the formation of pancreatic cancer [60]. *NKX2-1* is a tumor biomarker for lung cancer [61].

Between the *Pancreatic* and *Colon* adenocarcinoma groups, we can observe higher expression values for genes *MYB* and *CDX2*. *MYB* is a regulator of certain cells of the colon, with increased expression occurring with colon cancer cells [62]. Additionally, the expression of *PER1* is lower for colon adenocarcinoma, compared to pancreatic adenocarcinoma. The expression of *PER1* prevents the programmed cell death of pancreatic cancer cells and thus promotes cancer [63]. These and the above findings are also in line with data provided by The Human Protein Atlas [64].

Based on these insights, we were able to confirm that the tumor types are characterized by the expression of genes as well as the frequency of different types of KRAS mutations.

7.3 Domain Expert Feedback

During the design and development of our approach, we organized multiple feedback sessions with two experts in the respective domains of our use cases: a tournament chess player and a senior scientist working in a drug discovery team at a pharmaceutical company. In these feedback sessions, we presented the tool, the use case, and discussed potential findings. The experts had access to the tool between the interviews and also explored the data themselves. Based on their feedback, we continuously increased the usability of the presented exploration approach. In the following, we first summarize general feedback given by the experts, followed by domain-specific feedback on the individual use cases.

Besides providing valuable input on the usability, the experts quickly came up with more ideas on how to use the tool for further exploring the data. Both experts highlighted the utility of the high-dimensional comparison, especially of groups created in the embedding. This comparison was initially limited to groups that users needed to manually connect via edges. We removed this restriction and users can now directly compare arbitrary groups, leading to the workflow described in Section 5.3. The chess expert also noted that keeping track of the currently compared data in the embedding is difficult for many comparisons. We, therefore, added stepper controls to the comparison pane and highlight the currently compared items/groups in the embedding and sidebar (see Figure 6). Based on the feedback from the cancer genomics expert, we also rank the attributes in the summary and difference visualization and provide a filter function to support a large number of attributes.

Specifically for chess, we designed the summary and difference visualization in close collaboration with the domain expert. In the future, we plan to integrate datasets that contain more professional chess games and additional metadata. We are also working on a better representation of the moves in the embedding through Parametric UMAP [65], as the irregular distances between states do not yet reflect the importance of the moves. The domain expert has also expressed that he would like to be assisted in identifying the most common moves between a start and an end state, by automatically creating groups along the game sequences. We describe our plans to generate group sequences along trajectory bundles in Section 8.2.

For the cohort analysis use case, the expert noted that creating and labeling the groups becomes more and more tedious as the number increases. We discuss our plans to further support group creation in Section 8.2 and are additionally working on combining the presented approach with a dedicated tool for cohort creation and characterization [59].

8 Discussion

In this section, we reflect on challenges and potential improvements of our work.

As demonstrated in both use cases, our embedding scales well with both the number of items and the number of attributes. The chess dataset consists of approximately 40 000 items and 70 attributes, and the projection of cancer genomics data is based on 1 238 items and 720 attributes.

Technically, the rendering of the embedding scales well up to 100 000 items. However, the scatterplot does not, due to overplotting. Some works described in Section 3.4 improve the readability of scatterplots with many items. However, large datasets also pose challenges in creating structures and relationships, as well as in exploring subsets of the data. We discuss our thoughts on Focus + Context analysis and give an outlook on narrative automatic clustering to better handle large amounts of data in the following sections.

8.1 Focus + Context Exploration

Once users have uncovered notable structures and identified interesting groups in a dataset, they may want to perform a subsequent analysis that focuses on a particular group or a single level of a hierarchy. As part of the reprojection feature of our prototype, we already support a *subspace reprojection*. Users select a group of items, for which a new embedding will be calculated within the items' bounding box. The remaining points of the scatterplot are left untouched. We have found that this subspace reprojection is useful to *focus* on a specific group, especially when a different attribute set has been identified as relevant within that group. However, this approach introduces the challenge of correctly providing *context*, since (apart from the coarse positioning within the previous bounding box) the newly created embedding space is completely decoupled from the rest of the embedding. Neighborhoods between points within and outside the reprojected part of the embedding may not be interpretable. We are currently investigating how to address this challenge arising from changing only parts of the embedding, potentially based on one of the promising strategies discussed by Höllt et al. [29] and Marcílio et al. [66].

A different approach that could allow users to focus on specific substructures is what Nonato and Aupetit call *spatially structured enrichment* [2]. Here, the embedding space is subdivided into segments that are enriched with additional information, resulting in a space-filling layout that resembles a tree map [67]. Depending on the application domain, it may be possible to optimize the design of our summary visualizations to better support such a subdivided, space-filling layout. If this approach is additionally made adaptive/interactive (e.g., based on the current zoom level, or directly bound to the data similarity in specific regions), it could potentially help users to navigate and create even deep hierarchies. The resulting problem of effectively comparing groupings across multiple levels of the hierarchy could be elegantly solved with storytelling, discussed in the next section, where the summary visualizations for higher levels could be "stored" while users zoom in to explore deeper levels.

8.2 Towards Automated Analysis and Storytelling

Our current prototype already supports automatic group generation by density-based clustering. Users can combine this approach with manual selections of items to set up hierarchical relationships. The feedback from the two domain experts also revealed opportunities for automation.

In future work, we aim to improve upon the automation capabilities and enable an online, automated creation of hierarchical structures. To this end, we plan to support techniques such as Haisu [27] or HSNE [25], which take preexisting hierarchical information into account, and techniques such as tree-SNE [26], which perform hierarchical clustering and DR simultaneously. Combinations of dimension reduction and clustering algorithms are also discussed by Wenskovitch et al. [68] and offer directions for further improvements.

In addition to automatically creating groups and structures, it may also be possible to automate the creation of meaningful relationships and stories by connecting groups based on predefined rules. In the case of sequentially connected items, we already experimented with trajectory data mining techniques [69]. This way, we could automatically detect clusters that form trajectory bundles along chess game trajectories.

After automatically creating clusters and finding relationships, an initial automatic arrangement could be presented to the users. Consequently, users can inspect the results and validate or improve the findings. We imagine that this validation step can be smoothly integrated into our existing comparison pane. We already realized the potential of expanding our prototype with storytelling features, such as the support for annotations or presentation following a Vistories approach [70]. This would not only close the loop between exploration, authoring, and presentation, but these features would be especially useful in the context of automation, as they would help users make sense of the suggested stories.

9 Conclusion

Extracting meaning from high-dimensional data is a major challenge [2], [3]. In this paper, we presented an interactive visual exploration approach for structural relationships in embeddings of high-dimensional data. We support the exploratory analysis of items in the low-dimensional space through layout enrichment, the interactive creation of groups and relationships, and means of comparison thereof. We introduced tailored summary and difference visualizations for various data types and semantic contexts. We are confident that the combination of embedding-based explorations with structural analysis and creation can be applied to various domains and applications.

Acknowledgments

We thank Johannes Fürnkranz and Thomas Zichner for their feedback as domain experts in chess and cancer genomics, respectively, and Michael Pühringer for his implementation support.

This work was supported in part by the Boehringer Ingelheim Regional Center Vienna, the State of Upper Austria and the Austrian Federal Ministry of Education, Science and Research via the LIT – Linz Institute of Technology (LIT-2019-7-SEE-117), the State of Upper Austria (Human-Interpretable Machine Learning), and the Austrian Science Fund (FWF DFH 23-N). This work was supported in part by the FFG, Contract No. 881844: ”Pro^{2}Future is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG.”.

References

[1] S. Yan, D. Xu, B. Zhang, H.-j. Zhang, Q. Yang, and S. Lin, “Graph Embedding and Extensions: A General Framework for Dimensionality Reduction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, Jan. 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4016549/

[2] L. G. Nonato and M. Aupetit, “Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2650–2673, 2019.

[3] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, “Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences,” in *Proceedings of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV ’14)*. ACM, 2014, pp. 1–8.

[4] T. Munzner, *Visualization Analysis and Design*. CRC Press, Taylor & Francis Group, 2014.

[5] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, “Time curves: Folding time to visualize patterns of temporal evolution in data,” *IEEE Transactions on Visualization and Computer Graphics (InfoVis ’15)*, vol. 22, no. 1, pp. 559–568, 2016.

[6] A. Hinterreiter, C. Steinparz, M. Schöfl, H. Stitz, and M. Streit, “Projection Path Explorer: Exploring Visual Patterns in Projected Decision-Making Paths,” *ACM Transactions on Interactive Intelligent Systems*, vol. 11, no. 3–4, p. Article 22, 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3387165

[7] F. V. Paulovich, M. C. F. Oliveira, and R. Minghim, “The Projection Explorer: A Flexible Tool for Projection-based Multidimensional Visualization,” in *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI ’07)*. IEEE, Oct. 2007, pp. 27–36. [Online]. Available: http://ieeexplore.ieee.org/document/4368165/

[8] N. Heulot, J.-D. Fekete, and M. Aupetit, “Proxilens: Interactive exploration of high-dimensional data using projections,” in *EuroVis Workshop on Visual Analytics using Multidimensional Projections (VAMP)*. The Eurographics Association, 2013.

[9] H. Liao, Y. Wu, L. Chen, and W. Chen, “Cluster-Based Visual Abstraction for Multivariate Scatterplots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 9, pp. 2531–2545, Sep. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8047300/

[10] A. Chatzimparmpas, R. M. Martins, and A. Kerren, “t-viSNE: Interactive Assessment and Interpretation of t-SNE Projections,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 8, pp. 2696–2714, Aug. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9064929/

[11] J. Xia, L. Gao, K. Kong, Y. Zhao, Y. Chen, X. Kui, and Y. Liang, “Exploring linear projections for revealing clusters, outliers, and trends in subsets of multi-dimensional datasets,” *Journal of Visual Languages & Computing*, vol. 48, pp. 52–60, Oct. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1045926X18301289

[12] R. Cutura, M. Aupetit, J.-D. Fekete, and M. Sedlmair, “Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations,” in *Proceedings of the International Conference on Advanced Visual Interfaces (AVI ’20)*. ACM, Sep. 2020, pp. 1–9. [Online]. Available: https://dl.acm.org/doi/10.1145/3399715.3399875

[13] D. L. Arendt, N. Nur, Z. Huang, G. Fair, and W. Dou, “Parallel Embeddings: A Visualization Technique for Contrasting Learned Representations,” in *Proceedings of the International Conference on Intelligent User Interfaces (IUI ’20)*. Association for Computing Machinery, 2020, pp. 259–274. [Online]. Available: https://doi.org/10.1145/3377325.3377514

[14] F. Heimerl, C. Kralj, T. Moller, and M. Gleicher, “embComp: Visual Interactive Comparison of Vector Embeddings,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9301222/

[15] T. Fujiwara, X. Wei, J. Zhao, and K.-L. Ma, “Interactive Dimensionality Reduction for Comparative Analysis,” *IEEE Transactions on Visualization and Computer Graphics (VIS ’21)*, vol. 28, no. 1, pp. 758–768, 2022.

[16] H. Kim, J. Choo, H. Park, and A. Endert, “InterAxis: Steering Scatterplot Axes via Observation-Level Interaction,” *IEEE Transactions on Visualization and Computer Graphics (VAST ’15)*, vol. 22, no. 1, pp. 131–140, 2016.

[17] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert, “AxiSketcher: Interactive Nonlinear Axis Mapping of Visualizations through User Drawings,” *IEEE Transactions on Visualization and Computer Graphics (VAST ’16)*, vol. 23, no. 1, pp. 221–230, 2017.

[18] Y. Ma and R. Maciejewski, “Visual Analysis of Class Separations With Locally Linear Segments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 241–253, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9146191/

[19] R. Faust, D. Glickenstein, and C. Scheidegger, “DimReader: Axis lines that explain non-linear projections,” *IEEE Transactions*

on Visualization and Computer Graphics (InfoVis '18), vol. 25, no. 1, pp. 481–490, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8440820/

[20] A. Bibal, V. M. Vu, G. Nanfack, and B. Frénay, "Explaining t-SNE Embeddings Locally by Adapting LIME," in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2020, pp. 393–398.

[21] W. E. Marcílio-Jr and D. M. Eler, "Explaining dimensionality reduction results using Shapley values," *Expert Systems with Applications*, vol. 178, p. 115020, Mar. 2021.

[22] Y. Liu, E. Jun, Q. Li, and J. Heer, "Latent Space Cartography: Visual Analysis of Vector Space Embeddings," *Computer Graphics Forum (EuroVis '19)*, vol. 38, no. 3, pp. 67–78, Jun. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13672

[23] 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 (InfoVis '15)*, vol. 22, no. 1, pp. 629–638, 2016.

[24] J.-T. Sohns, M. Schmitt, F. Jirasek, H. Hasse, and H. Leitte, "Attribute-based Explanation of Non-Linear Embeddings of High-Dimensional Data," *IEEE Transactions on Visualization and Computer Graphics (VIS '21)*, vol. 28, no. 1, pp. 540–550, Jan. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9552929/

[25] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova, "Hierarchical stochastic neighbor embedding," *Computer Graphics Forum (EuroVis '16)*, vol. 35, pp. 21–30, 2016.

[26] I. Robinson and E. Pierce-Hoffman, "Tree-SNE: Hierarchical Clustering and Visualization Using t-SNE," *arXiv:2002.05687*, Feb. 2020. [Online]. Available: http://arxiv.org/abs/2002.05687

[27] K. C. VanHorn and M. C. Çobanoğlu, "Haisu: Hierarchical Supervised Nonlinear Dimensionality Reduction," *bioRxiv*, vol. 10.05.324798, Oct. 2020. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/2020.10.05.324798

[28] G. Chao, Y. Luo, and W. Ding, "Recent Advances in Supervised Dimension Reduction: A Survey," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 341–358, Jan. 2019. [Online]. Available: https://www.mdpi.com/2504-4990/1/1/20

[29] T. Höllt, A. Vilanova, N. Pezzotti, B. Lelieveldt, and H. Hauser, "Focus+Context Exploration of Hierarchical Embeddings," *Computer Graphics Forum (EuroVis '19)*, vol. 38, no. 3, pp. 569–579, Jun. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/cgf.13711

[30] P. Cui, X. Wang, J. Pei, and W. Zhu, "A Survey on Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2019.

[31] Q. Li, K. S. Njotoprawiro, H. Haleem, Q. Chen, C. Yi, and X. Ma, "EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection," in *IEEE Conference on Visual Analytics Science and Technology (VAST '18)*, 2018, pp. 48–59.

[32] A. Boggust, B. Carter, and A. Satyanarayan, "Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples," *arXiv:1912.04853*, 2019. [Online]. Available: http://arxiv.org/abs/1912.04853

[33] Z. Jin, Y. Wang, Q. Wang, Y. Ming, T. Ma, and H. Qu, "GNNLens: A Visual Analytics Approach for Prediction Error Diagnosis of Graph Neural Networks," *arXiv:2011.11048*, 2021. [Online]. Available: http://arxiv.org/abs/2011.11048

[34] N. Cao, D. Gotz, J. Sun, and H. Qu, "DICON: Interactive Visual Analysis of Multidimensional Clusters," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2581–2590, 2011.

[35] D. J. Lehmann, F. Kemmler, T. Zhyhalava, M. Kirschke, and H. Theisel, "Visualnostics: Visual Guidance Pictograms for Analyzing Projections of High-dimensional Data," *Computer Graphics Forum (EuroVis '15)*, vol. 34, no. 3, pp. 291–300, Jun. 2015. [Online]. Available: http://doi.wiley.com/10.1111/cgf.12641

[36] P. Joia, F. Petronetto, and L. G. Nonato, "Uncovering Representative Groups in Multidimensional Projections," *Computer Graphics Forum (EuroVis '15)*, vol. 34, no. 3, pp. 281–290, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12640

[37] W. E. Marcílio-Jr, D. M. Eler, R. E. Garcia, R. C. M. Correia, L. F. Silva, and L. F. Silva, "A Hybrid Visualization Approach to Perform Analysis of Feature Spaces," in *International Conference on Information Technology–New Generations (ITNG '20)*, vol. 1134. Springer International Publishing, 2020, pp. 241–247. [Online]. Available: http://link.springer.com/10.1007/978-3-030-43020-7_32

[38] J. Jo, F. Vernier, P. Dragicevic, and J.-D. Fekete, "A Declarative Rendering Model for Multiclass Density Maps," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '18)*, vol. 25, no. 1, pp. 470–480, 2019.

[39] N. Elmqvist and J.-D. Fekete, "Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, 2010.

[40] A. Sarikaya, M. Gleicher, and D. A. Szafir, "Design Factors for Summary Visualization in Visual Analytics," *Computer Graphics Forum (EuroVis '18)*, vol. 37, no. 3, pp. 145–156, Jun. 2018. [Online]. Available: http://doi.wiley.com/10.1111/cgf.13408

[41] E. Catmull and R. Rom, "A class of local interpolating splines," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, 1974, pp. 317–326.

[42] R. Cabello, "Catmull–rom spline," 2021, https://threejs.org/docs/#api/en/extras/core/Interpolations.CatmullRom.

[43] A. Sarikaya and M. Gleicher, "Scatterplots: Tasks, Data, and Designs," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '17)*, vol. 24, no. 1, pp. 402–412, Jan. 2018.

[44] C. Partl, A. Lex, M. Streit, D. Kalkofen, K. Kashofer, and D. Schmalstieg, "enRoute: Dynamic Path Extraction from Biological Pathway Maps for Exploring Heterogeneous Experimental Datasets," *BMC Bioinformatics*, vol. 14, no. Suppl 19, p. S3, 2013. [Online]. Available: http://www.biomedcentral.com/1471-2105/14/S19/S3/abstract

[45] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '12)*. IEEE, 2012, pp. 1 –8.

[46] A. Karpathy, "tSNEJS," 2016, https://github.com/karpathy/tsnejs.

[47] People+AI Research (PAIR) Initiative, "Umap-js," 2019, https://github.com/PAIR-code/umap-js.

[48] A. M. Horst, A. P. Hill, and K. B. Gorman, "palmerpenguins: Palmer Archipelago (Antarctica) penguin data," 2020. [Online]. Available: https://allisonhorst.github.io/palmerpenguins/

[49] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, Mar. 2017. [Online]. Available: http://joss.theoj.org/papers/10.21105/joss.00205

[50] R. Cabello, "three.js – JavaScript 3D editor," 2021, https://github.com/mrdoob/three.js/.

[51] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '16)*, vol. 23, no. 1, pp. 341–350, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7539624/

[52] D3, "d3-contour," 2020, https://github.com/d3/d3-contour.

[53] Facebook Inc., "React – a JavaScript library for building user interfaces," 2021, https://reactjs.org/.

[54] P. Havard, "Kingbase – a free chess games database, updated monthly," 2019, https://www.kingbase-chess.net/.

[55] D. Kızılırmak, "pgn2gif," 2018, https://github.com/dn1z/pgn2gif.

[56] G. Kolata, "How Scientists Shot Down Cancer's 'Death Star'," *The New York Times*, Feb. 2021. [Online]. Available: https://www.nytimes.com/2021/02/05/health/lung-cancer-drug.html

[57] National Cancer Institute, "The cancer genome atlas program," 2019, https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga.

[58] M. Streit, S. Gratzl, H. Stitz, A. Wernitznig, T. Zichner, and C. Haslinger, "Ordino: a visual cancer analysis tool for ranking and exploring genes, cell lines and tissue samples," *Bioinformatics*, vol. 35, no. 17, pp. 3140–3142, 2019.

[59] P. Adelberger, K. Eckelt, M. J. Bauer, M. Streit, C. Haslinger, and T. Zichner, "Coral: a web-based visual analysis tool for creating and characterizing cohorts," *Bioinformatics*, vol. 37, no. 23, pp. 4559–4561, Dec. 2021. [Online]. Available: https://doi.org/10.1093/bioinformatics/btab695

[60] K. Kasai, "GLI1, a master regulator of the hallmark of pancreatic cancer." *Pathology international*, vol. 66, no. 12, pp. 653–660, Dec. 2016.

[61] L. Yang, M. Lin, W.-j. Ruan, L.-l. Dong, E.-g. Chen, X.-h. Wu, and K.-j. Ying, "Nkx2-1: a novel tumor biomarker of lung cancer," *Journal of Zhejiang University. Science. B*,

vol. 13, no. 11, pp. 855–866, Nov. 2012. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/23125078

[62] R. G. Ramsay and T. J. Gonda, "MYB function in normal and cancer cells," *Nature Reviews Cancer*, vol. 8, no. 7, pp. 523–534, Jul. 2008. [Online]. Available: https://doi.org/10.1038/nrc2439

[63] F. Sato, C. Nagata, Y. Liu, T. Suzuki, J. Kondo, S. Morohashi, T. Imaizumi, Y. Kato, and H. Kijima, "PERIOD1 is an Anti-apoptotic Factor in Human Pancreatic and Hepatic Cancer Cells," *The Journal of Biochemistry*, vol. 146, no. 6, pp. 833–838, Aug. 2009. [Online]. Available: https://doi.org/10.1093/jb/mvp126

[64] M. Uhlen, C. Zhang, S. Lee, E. Sjöstedt, L. Fagerberg, G. Bidkhori, R. Benfeitas, M. Arif, Z. Liu, F. Edfors, K. Sanli, K. v. Feilitzen, P. Oksvold, E. Lundberg, S. Hober, P. Nilsson, J. Mattsson, J. M. Schwenk, H. Brunnström, B. Glimelius, T. Sjöblom, P.-H. Edqvist, D. Djureinovic, P. Micke, C. Lindskog, A. Mardinoglu, and F. Ponten, "A pathology atlas of the human cancer transcriptome," *Science*, vol. 357, no. 6352, p. eaan2507, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aan2507

[65] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric UMAP Embeddings for Representation and Semisupervised Learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, Oct. 2021. [Online]. Available: https://doi.org/10.1162/neco_a_01434

[66] W. E. Marcílio-Jr, D. M. Eler, F. V. Paulovich, J. F. Rodrigues-Jr, and A. O. Artero, "ExplorerTree: A Focus+Context Exploration Approach for 2D Embeddings," *Big Data Research*, vol. 25, p. 100239, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214579621000563

[67] E. Gomez-Nieto, W. Casaca, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato, "Dealing with Multiple Requirements in Geometric Arrangements," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1223–1235, Mar. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7296669/

[68] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North, "Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics (VAST '17)*, vol. 24, no. 1, pp. 131–141, 2018.

[69] Y. Zheng, "Trajectory Data Mining: An Overview," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, pp. 1–41, May 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2764959.2743025

[70] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit, "From Visual Exploration to Storytelling and Back Again," *Computer Graphics Forum (EuroVis '16)*, vol. 35, no. 3, pp. 491–500, 2016.

Klaus Eckelt is a PhD student at the Institute of Computer Graphics at Johannes Kepler University Linz, Austria. His main research interests include biomedical data visualization, visual analytics, and statistics. He received his Diplomingenieur (MSc) in Medical Informatics from TU Wien, Vienna, Austria.
For more information see https://eckelt.info.

Andreas Hinterreiter is a PhD student at the Institute of Computer Graphics, Johannes Kepler University (JKU) Linz. His research interests include dimensionality reduction and explainable AI. He spent parts of his PhD at the Biomedical Image Analysis Group at Imperial College London. He received his Diplomingenieur (MSc) in Technical Physics from JKU.

Patrick Adelberger received his master's degree in Medical Informatics at TU Wien, Vienna, Austria. Currently, he is a PhD student at the Institute of Computer Graphics, Johannes Kepler University (JKU) Linz, Austria. His research interests include visualization for biomedical data and visual analytics.

Conny Walchshofer is a PhD student at the Institute of Computer Graphics at Johannes Kepler University Linz, Austria. In her prior research, she focused on the perception and handling of multidimensional visualizations. She applies an interdisciplinary approach to judge cognitive load during the interpretation of visual representations by using physiological measurement methods (e.g., eye-tracking, heart rate variability).

Vaishali Dhanoa is a researcher at Pro2Future GmbH and PhD student at the Institute of Computer Graphics of the Johannes Kepler University (JKU) Linz, Austria. Her research interests include visual analytics, comprehensible approaches to user onboarding, and developing performance optimized interactive tools. In the past, she worked as a software developer at Intel GmbH Linz, Austria. She received her Diplomingenieur (MSc) in Informatics from JKU.

Christina Humer is a PhD student at the Institute of Computer Graphics, Johannes Kepler University Linz, Austria. Her research interests include explainable AI and visual analytics. She received her Diplomingenieur (MSc) in Computer Science with a focus on Data Science from JKU.

Moritz Heckmann is a student undertaking a master's degree in computer science at the Johannes Kepler University Linz. He works part time as technical support at the Institute of Computer Graphics and as a front-end developer for datavisyn.

Christian Steinparz received his master's degree in Artificial Intelligence at Johannes Kepler University Linz, Austria. He is now working at the Institute of Computer Graphics at JKU. His scientific areas of interest include explainable AI, reinforcement learning, and dimensionality reduction.

Marc Streit is a Full Professor at the Johannes Kepler University Linz in Austria where he leads the JKU Visual Data Science Lab (https://jku-vds-lab.at/). He finished his PhD at the Graz University of Technology in 2011. His scientific areas of interest include visualization, visual analytics, and explainable AI. He won multiple best paper and honorable mention awards at major conferences in the field. Marc is also co-founder and CEO of datavisyn, a spin-off company that develops data visualization solutions for pharmaceutical and biomedical R&D. For more information see http://marc-streit.com.