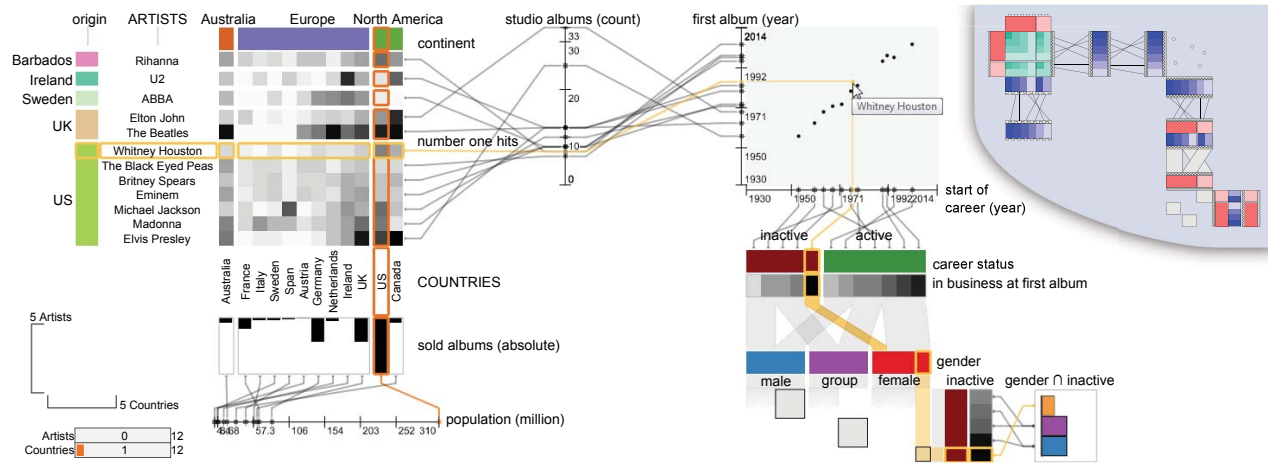


# Domino: Extracting, Comparing, and Manipulating Subsets across Multiple Tabular Datasets

Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hanspeter Pfister and Marc Streit



**Fig. 1:** Domino showing relationships between subsets of a music charts dataset. The visualization illustrates that Whitney Houston is a female, inactive artist who has had many number-one hits in English speaking countries but produced fewer than 10 studio albums. The schematic illustration in the top-right corner shows the setup using a graphical notation.

**Abstract**— Answering questions about complex issues often requires analysts to take into account information contained in multiple interconnected datasets. A common strategy in analyzing and visualizing large and heterogeneous data is dividing it into meaningful subsets. Interesting subsets can then be selected and the associated data and the relationships between the subsets visualized. However, neither the extraction and manipulation nor the comparison of subsets is well supported by state-of-the-art techniques. In this paper we present *Domino*, a novel multimodal visualization technique for effectively representing subsets and the relationships between them. By providing comprehensive tools to arrange, combine, and extract subsets, *Domino* allows users to create both common visualization techniques and advanced visualizations tailored to specific use cases. In addition to the novel technique, we present an implementation that enables analysts to manage the wide range of options that our approach offers. Innovative interactive features such as placeholders and live previews support rapid creation of complex analysis setups. We introduce the technique and the implementation using a simple example and demonstrate scalability and effectiveness in a use case from the field of cancer genomics.

**Index Terms**—Multiple coordinated views, visual linking, relationships, heterogeneous data, categorical data

## 1 INTRODUCTION

Common heterogeneous data visualization methods show data associated with a single shared item type. The popular cars dataset [12], for example, is defined over the item type *car* and contains multiple observations across several data types for each car. A parallel coordinates plot could be used to visualize such data. Multi-dataset visualizations typically follow the same pattern: they refer to a single, shared item type. A medical dataset may contain data about patients, such as gender, birth date, or height, while a blood test dataset for the same patients will contain measurements such as blood type or white blood

cell count. These datasets follow a relational data model where each dataset contains one or more observations for a given item type.

In practice, however, an increasing number of domains contain rich data for multiple item types. Figure 1 shows a simple illustrative example from the music industry<sup>1</sup> that includes multiple datasets defined in terms of the item types *artist* (e.g., count of number-one hits, gender, origin) and *country* (e.g., number-one hits per artists, total number of albums sold, sales normalized to population).

The **primary contribution** of this paper is *Domino*, a novel visualization technique that enables analysts to explore a mix of numerical and categorical data connected via various item types. This is achieved by allowing users to freely arrange, combine, and manipulate subsets (blocks), visualize associated data, and explicitly represent the relationships between these subsets. *Domino* enables analysts to rapidly assemble both established visualization techniques and novel combinations specifically tailored to the task at hand.

Our **secondary contribution** is a prototype implementation that enables users to exploit the wide spectrum of possibilities that the technique offers. To assist users in the process of adding new subsets to

- Samuel Gratzl and Marc Streit are with Johannes Kepler University Linz. E-mail: {samuel.gratzl, marc.streit}@jku.at.
- Alexander Lex and Hanspeter Pfister are with Harvard University. E-mail: {alex, pfister}@seas.harvard.edu.
- Nils Gehlenborg is with Harvard Medical School. E-mail: nils@hms.harvard.edu.

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

Digital Object Identifier 10.1109/TVCG.2014.2346260

<sup>1</sup>The dataset was collected from multiple web sources. References and the dataset are available at <http://domino.caleydo.org>.

the setup and combining previously defined ones, we introduce *placeholders* that indicate possible options for placing a subset, and *live previews* that show possible visual encodings of the associated data.

We demonstrate the utility and flexibility of the system by means of two use cases: the first discusses a small music charts dataset, while the second uses a collection of datasets from cancer genomics.

## 2 RELATED WORK

Domino is a visualization technique designed for multiple heterogeneous, high-dimensional datasets and relationships between them. These relationships can also be interpreted as set relationships. Therefore, the relevant body of related work comprises visualization techniques for both sets and high-dimensional heterogeneous data. As Domino is a meta-visualization technique that enables users to create new visualizations which are interlinked, we also discuss multiple coordinated view systems in general, and integrated views systems that explicitly represent relationships between linked visualizations in particular.

**Set and Subset Relationships** Visualizing sets and their relationships is a problem frequently encountered in many domains. The most widely used set visualization techniques are Venn and Euler diagrams, which, however, do not scale beyond very small numbers of sets. Consequently, the visualization of relationships between sets has been, and continues to be, an active area of research. In a recent survey paper [1], Alsallakh et al. reviewed and classified existing set visualization techniques based on three classes of tasks: element-related, set-related, and attribute-related tasks. Most techniques that address attribute-related tasks can handle one or a few attributes. Domino, however, is designed to deal with relationships between tabular datasets in which each of potentially thousands of dimensions could be seen as a separate attribute.

A technique that effectively compares groups of sets (categories) with each other is parallel sets [18]. Parallel sets arranges the (non-overlapping) sets of a group in a column and compares them to adjacent groups of sets using bands. The width of a band encodes the elements shared between two sets. In previous work, we introduced techniques that use this parallel sets metaphor for comparing relationships of partitioned datasets and embed the tabular data from which the partitions were derived, thus visualizing both, set relationships and attributes efficiently. A common example is clustering performed on a multi-dimensional dataset (e.g., gene expression data) where items (*genes*) are partitioned according to similarities in the data. Analysts are usually not only interested in the partitioning of the items, but also in the actual data associated with the items. The *Matchmaker* technique [21] supports this task by juxtaposing multiple partitioned tabular datasets, represented as heatmaps, and connecting the groups with bands and the items with individual splines. This column-based approach enables analysts to compare different partitions of the same dataset (e.g., the results of different clustering algorithms) or to explore relationships among multiple partitioned datasets containing the same items (e.g., multiple clustered expression datasets from different groups of patients containing the same genes). *VisBricks* [19] generalized this approach by allowing users to switch between various visualization techniques independently for each subset – a concept known as multiform visualization [19, 28]. Analysts can, for instance, choose parallel coordinates as the representation for individual groups to perform filtering tasks and a heatmap to see the overall pattern. *StratomeX* [22] introduced subset visualization for multiple heterogeneous datasets.

All three techniques—Matchmaker, VisBricks, and StratomeX—show the relationships between multiple partitioned sets together with the multi-dimensional datasets on which the partitions are defined. However, due to the column-based arrangement of the tabular datasets, the analyst is limited to relationships of one type. For instance, if two clustered gene expression tables are positioned side by side (*genes x patients*), the analyst must choose to visualize either gene or patient relationships, depending on the arrangement of the datasets. The free arrangement of subsets in Domino removes this restriction.

Another limitation of these techniques is that they require partitioned (categorical) datasets as columns, while Domino allows analysts to mix partitioned and numerical data. While our previous work focused on relationships between the partitioned sets, drawn as bands connecting the columns, Domino is able to show relationships between datasets at different levels of granularity: at the level of items, groups of items, or whole datasets. Finally, our earlier techniques do not allow users to extract and manipulate subsets, which is a central aspect of Domino.

In recent work we described *Furby* [31], a method for visualizing biclustering results in a force-directed layout. Bands between the clusters represent their overlaps in both rows and columns. However, similar to the approaches discussed above, Furby only encodes the overlap between whole biclusters and does not enable users to see individual relationships between shared items (rows/columns). Furthermore, Furby focuses on the overlap between matrix subsets, preventing users from including partitioned and numerical subsets in the sense-making process. Furby also does not address the extraction and manipulation of subsets.

**Multiple Coordinated and Integrated Views** An alternative to the multiform approaches described above are standard multiple coordinated views (MCV) [28], where each subset is visualized as a separate view. Using linking & brushing, users can then explore the relationships between the subsets. MCV is also applicable to scenarios that include multiple datasets connected by different item types. However, as MCV systems keep relationships between linked views [15] only implicitly, the approach has a major limitation: it requires users to actively select or filter items in order to see relationships between the subsets. Further, as the selected items from one view are simultaneously highlighted in all other views, users can generally only see that the item is part of the selection, but cannot determine which item in one view corresponds to which item in the other views if multiple items are selected.

Instead of implicitly linking items in a MCV setup, explicit links can be drawn to connect items across the views—an approach Javed and Elmqvist call integrated views [15]. Examples of integrated views are semantic substrates [29], the VisLink method [7], or context-preserving visual links [30]. Note that the multiform approaches described earlier in this section can also be seen as integrated views. While explicit links remove the need for interaction to see relationships, it comes at the cost of added visual clutter. In Domino we also make use of explicitly connected multiform visualizations; however, we reduce visual clutter by representing the relationships at various levels of granularity. In addition to representing relationships between individual items as single lines, Domino aggregates lines from the same group of a partition to bands.

**Meta-Visualization Techniques** The approaches most closely related to Domino are *Flexible Linked Axes (FLINA)* [6] by Claessen and van Wijk and *ConnectedCharts* [35] by Viau and McGuffin. Domino is akin to FLINA and ConnectedCharts, as it is also a meta-visualization for creating advanced visualization setups. However, due to its holistic conceptual approach that integrates one-dimensional numerical, categorical, and tabular data by connecting them at multiple levels of granularity, Domino can be applied to a broader spectrum of tasks.

FLINA [6] is a meta-visualization approach for creating axis-based visualization techniques, whose axis relationships can be described using the ARGOI formalism. Although it is a general and powerful concept, FLINA is restricted to showing item relationships between one-dimensional numerical data represented as 1D scatterplots (axes). ConnectedCharts [35] are related to FLINA in the sense that they also explicitly draw item-based relationships between visualizations. However, ConnectedCharts and Domino conceptually go beyond FLINA, as an axis-based representation is only one of many possible visualization techniques for displaying a subset. Another important difference from FLINA is that ConnectedCharts and Domino include one-dimensional and tabular datasets. While FLINA and ConnectedCharts are both limited to item-based relationships between the visualiza-

tions, Domino visually links them also on coarser abstractions, making it possible to show relationships between single items, between groups of categorical data, or between whole datasets. Another difference between the Domino technique from both FLINA and ConnectedCharts is its ability to let users not only explore relationships between subsets, but also refine existing and extract new subsets, which provides a great deal of flexibility in exploratory analysis scenarios.

### 3 DOMINO VISUALIZATION TECHNIQUE

Domino is a meta-visualization technique that enables analysts to arrange, extract, and manipulate subsets of interest and show the relationships between them at multiple levels of detail. The approach is designed for scenarios that include multiple heterogeneous datasets which are connected by shared identifiers (artist, country, etc.). The goal of Domino is to effectively represent **subsets**, the **data associated with the subsets**, and the **relationships between the subsets**.

The basic visual elements of the Domino technique are **blocks**, similar to tiles in a dominoes game, and **block relationships**, which connect the individual blocks. Figure 1 shows a snapshot of an analysis performed with the previously introduced music charts dataset. Blocks represent one or multiple combined subsets, where subsets can be numerical (e.g., *population* of countries), partitioned (artists’ *gender*), or matrix data (*# of number-one hits* by artists in various countries). Depending on the data and the analysis task, various visualization techniques can be used to represent the subsets. While the matrix, for instance, is represented as a heatmap, the population of countries is visualized as a 1D scatterplot. Blocks can be arbitrarily positioned and rotated. The visual representations of the relationships between blocks are automatically displayed if the item types of two neighboring blocks match, analogously to matching dominoes. We consider two blocks as neighbors if they can be connected by a line without cutting across another block. The rows in the matrix that correspond to the *artists*, for example, are visually linked to the 1D scatterplot showing the *# of studio albums*.

In Domino, users are able to switch between three levels of granularity at which the relationships can be represented—focusing on the overlap between whole blocks, groups within blocks, or down to the lowest level of individual items. Depending on the arrangement of the blocks and the chosen granularity level, relationships are represented using lines, bands (bundles of lines), points, or rectangular regions (see Figure 1). The flexibility of freely arranging and associating blocks in Domino allows analysts to both quickly produce established visualization techniques, such as parallel sets, parallel coordinates, 2D scatterplots, and to create tailored, complex visualizations to address the needs of a particular use case.

#### 3.1 Blocks

The basic visual unit of the Domino technique are **blocks**, rectangular regions that represent subsets. We define  $i$  as an item that is part of the set  $I$ .  $I$  contains only items of the same type (e.g., *country*). A subset  $S$  is defined as a mathematical subset of  $S \subseteq I$  (*countries in Europe*). We distinguish between three functions that determine the possible types of blocks in Domino (see Figure 2):

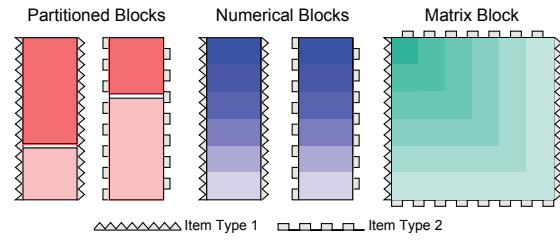
A **partitioned block** is defined by the function  $f_p$  that maps items to groups that are associated with unique natural numbers (e.g.,  $\{Africa(0), Europe(1), \dots, \{male(0), female(1)\}\}$ ):

$$f_p(i \in I) \mapsto g \in P,$$

where  $P$  is a partition of items  $I$ . The subset  $S_g$  of one specific group  $g \in P$  is defined as  $S_g = \{i | f_p(i) = g\}$ . A partitioned block can be viewed as a grouping of items without any associated data. The grouping can be the output of a clustering algorithm, derived from a categorical data attribute, or a quantization of a numerical block.

A **numerical block** is defined by the function  $f_n$  that maps items to numerical values (*population*, *# of studio albums*, or *retail value*):

$$f_n(i \in I) \mapsto v \in \mathbb{R}$$



**Fig. 2:** The three block types in Domino: partitioned (red), numerical (blue), and matrix blocks (green). The border style indicates the type of the items contained in the subset. While partitioned and numerical blocks are defined by one item type, matrix blocks have two, one in each direction. Only blocks with matching item types can be combined.

A **matrix block** is defined by the function  $f_m$  that maps the product set of two subsets of items to numerical values:

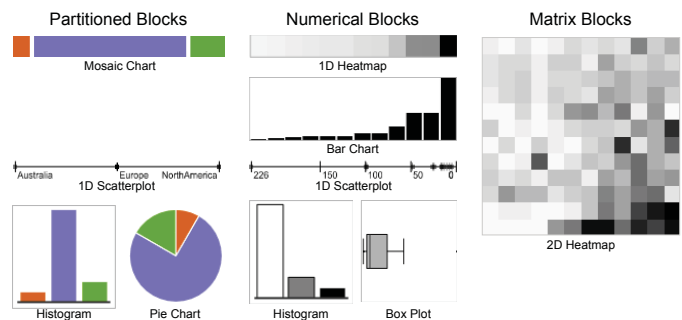
$$f_m((i, j) \in I_1 \times I_2) \mapsto v \in \mathbb{R}$$

In contrast to partitioned and numerical blocks, in which all items are part of the same item set, a matrix block consists of two item sets ( $I_1$  and  $I_2$ ). The item sets can be identical (e.g., in a *city distance matrix*) or different (*number-one hits for countries* or *gene expression of cancer patients*). However, the item sets themselves need to be homogeneous with respect to the item type. According to our definition of blocks, the prominent cars dataset [12], for example, is not a single matrix block, but a collection of individual numerical and partitioned blocks for each variable (*horse power*, *year of construction*, etc.)

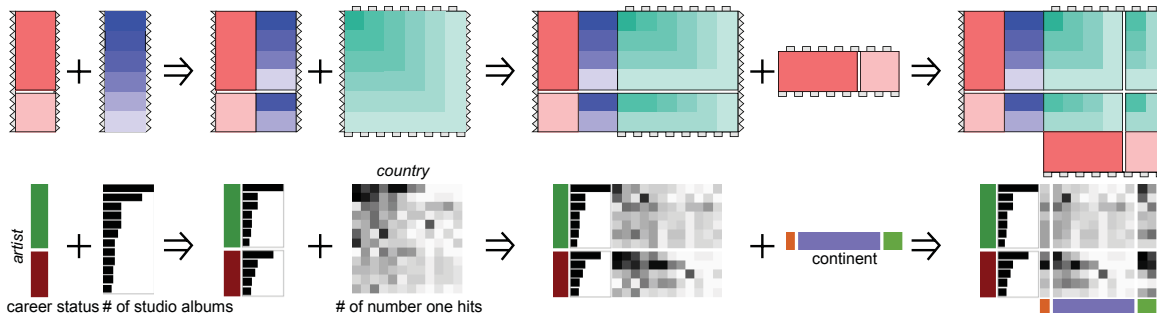
Figure 2 illustrates the three block types. The border style of the blocks encodes the different item types (e.g., *country* and *artist*). The sorting of items within blocks is essential for users to see patterns, for instance, in a clustered tabular dataset, or to see the distribution of items. For our technique, the sorting is also relevant because it determines the relationship representations that connect the blocks, as described in Section 3.2.2. From a mathematical point of view, sets do not define an order on the items that are contained in them. In Domino, items in blocks are sorted according to the functions defined above. In numerical blocks, items are sorted according to the associated data value. In partitioned blocks, the groups themselves are sorted according to the assigned natural number, while the individual items inside the group have no order. In matrix blocks, the two dimensions (horizontal and vertical) are sorted independently according to the mean value of the items. Sorting within blocks is illustrated by varying levels of brightness, as shown in Figure 2.

#### 3.1.1 Multiform Visualization

An essential characteristic of blocks in Domino is that they can be used to visualize subsets using multiform visualization, i.e., that users can switch between various visualization techniques.



**Fig. 3:** Block visualization techniques categorized by block type (partitioned, numerical, and matrix) including 1D and 2D heatmaps, mosaic and bar charts, 1D scatterplots, histograms, pie charts, and boxplots.



**Fig. 4:** Example sequence demonstrating the creation of a combined block. The user starts by attaching a partitioned block to a numerical block, which inherits the partitioning. Then the user adds a matrix block that is also partitioned accordingly. In the final step, the matrix column items are partitioned based on a partitioned block matching the column item type.

Domino is not bound to a predefined set of visualization techniques. Throughout the paper, however, we use a set of standard techniques, which can be seen in Figure 3, from which the analyst can freely choose. Depending on the type of the data represented by the block, only certain visualization types are suitable to encode data. While a heatmap representation can be applied to all three block types, 1D scatterplots and bar charts are available for partitioned and numerical blocks. Histograms and box plots are provided to represent numerical subsets, and pie charts can only be used to encode partitioned blocks. Although our current implementation is focused on these visualization types, the set of multiform techniques supported can be extended arbitrarily, depending on use case and application context. For the cancer genomics use case described in Section 6.2, for instance, we added Kaplan-Meier plots [26] to encode patient survival data.

### 3.1.2 Combined Blocks

Blocks that share the same item type, as indicated by the same border style in the illustrations, can be attached to each other, to form a **combined block**. The resulting combined block contains the union of the two item subsets. Figure 4 demonstrates by means of the music charts example how the stitching of blocks works. The analyst starts with the partitioned block *career status*, which separates active from inactive artists. In a second step, the analyst adds the numerical block *# of studio albums*, whose items are automatically partitioned according to the artists' *career status*. The two blocks are then combined with the matrix block that holds the absolute *# of number-one hits* for 12 countries. Again, as only one partitioning can drive the grouping of items in a combined block, the matrix block also inherits the *career status* partitioning. Finally, the analyst attaches the partitioned block *continent*, which splits the countries into three groups (Australia, Europe, and North America).

As described in Section 3.1, items within blocks are sorted. In the case of a combined block, the item order is determined by a hierarchical sorting strategy. For instance, if the user defines a primary and a secondary sorting criterion, the latter is only used if the order cannot be resolved using the former.

## 3.2 Block Relationships

In addition to blocks, the second class of visual elements in Domino are explicit representations of relationships between blocks. We consider blocks to be related to each other if their subsets contain items of the same type. In the previous section, we discussed that a partitioned block can only be subdivided into groups according to one primary partition, and a numerical block can only be sorted according to one primary sorting criterion. In the case of matrix blocks, these properties apply equally to each dimension. Consequently, the ability to create combined blocks by attaching multiple separate blocks requires re-sorting and/or re-distributing of the items in one or multiple blocks. Applying multiple sorting criteria hierarchically does not solve the issue, as only one can be the primary sorting criterion. However, in many analysis scenarios, users want to investigate the relationships between multiple blocks without giving up their designated sorting or partitioning. To address this issue, we introduce four degrees of relationships.

### 3.2.1 Relationship Degrees

A relationship degree defines the strength of the relationship between two blocks. We distinguish between four degrees: none, weak, medium, and strong, as shown in Table 1. The stronger the relationship, the more properties the blocks share. In Domino, the relationship degree is reflected by the proximity of two blocks in the layout. Users can freely switch between the four relationship degrees by moving blocks in the layout using drag-and-drop operations.

Two blocks have no relationship if they do not share the same item type (**none**). In the example used in Table 1, the artist's *career status* cannot be matched with the *population* of countries. If the blocks contain items of the same type but retain their individual sorting or partitioning, their relationship is **weak**. For instance, the artists in the *career status* block are assigned to the *active/inactive* group and are then weakly connected to the corresponding bar in the *# of studio albums* bar chart, which is sorted by the height of the bars. Blocks that have a **medium** relationship degree have the same partitioning but do not require the same sorting or visualization technique (see Section 3.1.1), which means that items are aggregated on a per-group basis. In the example, the same subsets are used as in the weak case, but this time the *# of studio albums* block inherits the partitioning from the artists' *career status* block. However, since both blocks can retain their individual sortings, different visualization techniques can be used. Finally, in the **strong** case both blocks also have to use the same sorting and must therefore apply the same visualization technique. In the example, the bars encoding the *# of studio albums* are sorted within the *career status* group.

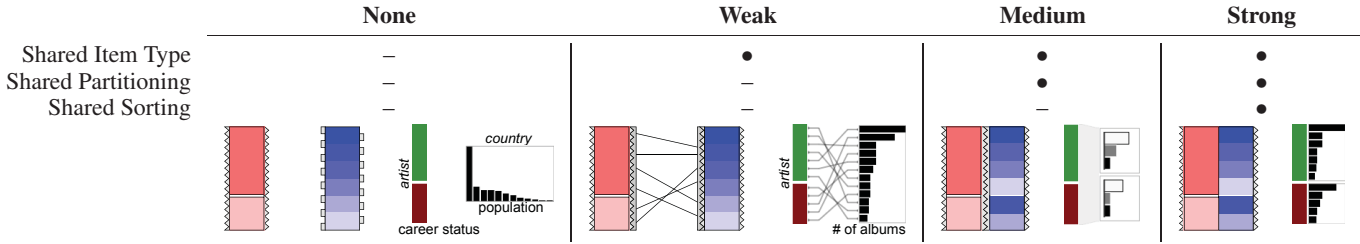
As blocks with a strong relationship are directly attached to each other (see Section 3.1.2 on combined blocks), no explicit visual representation of the relationship is needed. Consequently, the following discussion on how to represent relationships only applies to medium and weak relationship degrees.

### 3.2.2 Relationship Representation

We characterize the visual representation of relationships by two aspects: the **direction** of blocks—either **parallel** or **orthogonal**—and the **granularity** of the relationship. Depending on the user's task, we differentiate between three levels of granularity: **block**, **group**, and **item** relationships. Figure 5 illustrates the possible combinations using our graphical notation introduced in Figure 2.

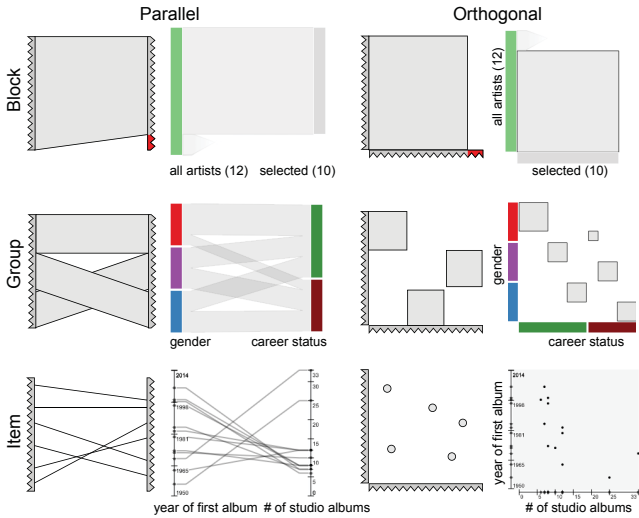
At the finest granularity level for visualizing relationships between items, we connect the items by drawing lines between parallel blocks or by drawing points at the hypothetical intersection point of related items of orthogonal blocks. While the former results in parallel coordinate-like setups, the latter results in scatterplots. The examples in the bottom row of Figure 5 show the correlation between the numerical 1D scatterplot *# of studio albums* and *year of first album*. The positions of the points in the scatterplot as well as the starting and end points of lines in the parallel coordinates plot depend on the currently applied visualization technique in the two blocks involved. For histograms, for example, the bin position is used.

To represent relationships at the coarser granularity levels, between groups and whole blocks, we draw bands and rectangular regions, re-



**Table 1:** Properties of the possible block relationship degrees (none, weak, medium, and strong) along with both a schematic diagram and an application example of each relationship degree. The stronger the relationship, the more properties two related blocks share.

spectively, depending on the arrangement of the blocks. The width of a band or the size of a region is proportional to the overlap between the subsets. In earlier work [19, 21, 22], we already made use of bands to relate groups between partitioned datasets. In the parallel case, blocks connected by bands produce the parallel sets technique [18]. The “parallel-group” example in Figure 5 shows the relationships between the groups of the partitioned blocks *gender* and *career status*, which are arranged in parallel. In the orthogonal case, the size of the regions is proportional to the number of shared items.



**Fig. 5:** Overview of possible relationship representations characterized by the granularity of the representation (block, group, and item) and the orientation of the block arrangement (parallel, orthogonal). All combinations are illustrated using the corresponding relationship representation and an example from the music charts dataset.

Subsets that do not fully overlap result in unshared items  $S_1 \setminus S_2$  and  $S_2 \setminus S_1$ . In the “block” examples in Figure 5, for instance, the relationships of “all artists” (green) to a subset of ten selected artists (gray) are shown. In the illustrations, unshared items are indicated in red. In our implementation a fading band is used to represent them.

Note that since the positioning of blocks in Domino is not restricted to perfect vertical or horizontal alignments, all relationship representations can be sheared.

### 3.3 Subset Extraction and Manipulation

Up to this point, we have focused our discussion on the exploration of predefined subsets, relationships between these subsets, and the different ways of visualizing them. Beyond predefined subsets, Domino also supports users in the process of creating new subsets. Users can define new subsets either by combining existing ones using logical operations (union, intersection, set difference) or by extracting parts of subsets based on selected items or groups. Depending on the visual representation of a block or relationship, users can, for example, extract items above a certain threshold in a numerical 1D scatterplot, items aggregated in a single bin of a histogram, or a collection of points in a

scatterplot that is formed by orthogonally arranged numerical blocks. In Figure 10, for instance, we selected the group *G-CIMP* in the matrix block and extracted all items to a separate block.

Enabling analysts to extract and manipulate subsets as needed during the analysis is a powerful approach in many scenarios. Ensuring that the user is always aware of the origin of the created subset is essential. In general, this provenance information can be provided either in a *temporal* or in a *spatial* manner. Domino offers both options.

In order to be able to present the provenance information, the user’s actions must be tracked. To understand what steps lead to the current subset configuration, users can then navigate through the history by using animation, or also replay the actions. Alternatively, provenance information can be encoded directly in the visualization, which we call *spatial* provenance representation. One example of such a spatial provenance presentation is the technique by van Elzen and van Wijk [33], where they present previous analysis steps in a linear fashion, reminiscent of a filmstrip. Other examples are *VisTrails* [2], *Ex-Plates* [16], and *GraphTrail* [8].

Instead of arranging the steps along a time line, the provenance information is contained implicitly in the design of the Domino technique. The relationship representation between subsets (blocks) and the colors of blocks allow analysts to track from which original subsets new subsets have been derived. In Figure 10, for instance, it can be seen that we extracted all items in the *G-CIMP* group to a separate block. Further, if a newly extracted subset turns out to be irrelevant, users can go back to the original subset and follow a different path in the analysis. In the use cases presented in Section 6, we demonstrate how analysts make use of the temporal replay capabilities and the implicit encoding for tracking the evolution of the current subset configuration.

## 4 SPECTRUM OF SUPPORTED VISUALIZATIONS

The modular block concept of Domino enables analysts to create sophisticated visualization setups. In addition to using the inherent visualization techniques realized in multiform blocks (see Section 3.1.1), analysts can easily assemble standard techniques such as scatterplots, scatterplot matrices (SPLOMs) [4], parallel coordinate plots [14], parallel sets [18], mosaic plots [11], Sankey diagrams [27], GPLOMs [13], Flexible Linked Axis [6] (although our current implementation does not support free rotation), ConnectedCharts (with the exception of stacking and nesting) [35], Table Lens [25], and scattering points in parallel coordinates [37]—to name but a few. Our technique also conceptually subsumes our previously published Matchmaker [21], VisBricks [19], StratomeX [22], and Furby [31] techniques. Figure 6 shows a collection of techniques created with Domino. Each figure contains a schematic illustration together with an example generated with the music charts dataset.

However, in addition to the flexible composition of well-known techniques, another strength of Domino is that these techniques can be combined arbitrarily to form a wide spectrum of new hybrid visualizations which the analyst can tailor specifically to the task at hand, as demonstrated in the music charts visualization shown in Figure 1 and the example setup from our cancer genomics use case presented in Figure 10.

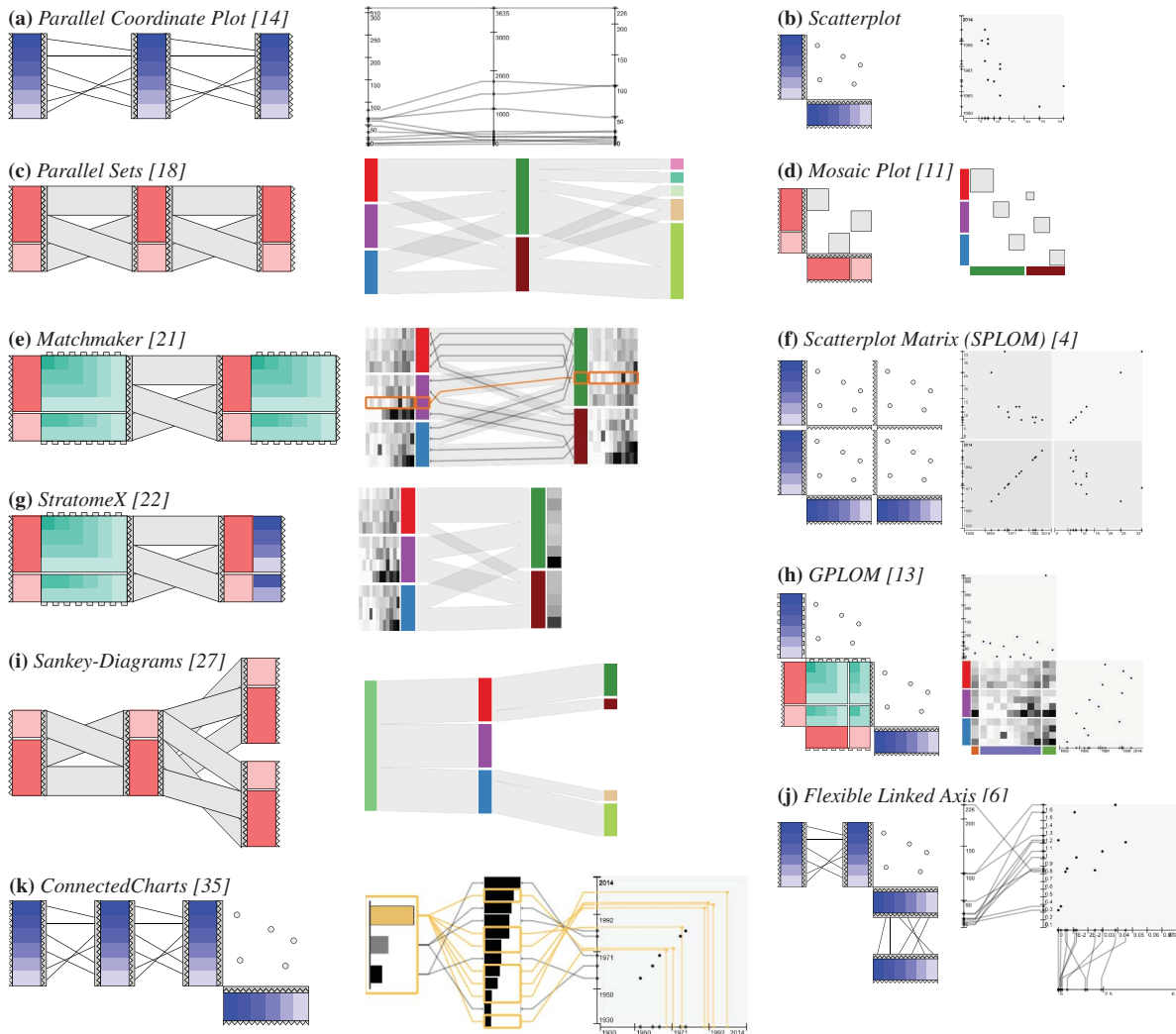


Fig. 6: Examples of supported visualization techniques demonstrating the flexibility of the Domino approach.

## 5 INTERFACE AND INTERACTION DESIGN

The Domino technique opens up a wide range of possibilities for exploring, relating, and manipulating subsets. To ensure that users can apply this generic approach efficiently, a well-designed user interface combined with a specifically tailored interaction concept is essential. To assist users in the process of adding new blocks to the setup and combining existing blocks, we introduce *placeholders* that indicate the various positions for placing a subset and *live previews* for showing possible visualization outcomes (see Section 5.1). Various interaction modes let users select and relate blocks, groups, or individual items (see Section 5.2). Interactions for block scaling (see Section 5.3) ensure that users can effectively handle scenarios of different complexity and scale—from small datasets like the music charts dataset with only 12 items to large scenarios including a multitude of datasets such as that discussed in the cancer use case described in Section 6.2.

The Domino prototype implementation is part of *Caleydo*, an open-source data visualization framework [20]. *Caleydo* is implemented in Java and uses OpenGL/JOGL for rendering. A demo version of Domino for Windows, Linux, and Mac OS X is freely available at <http://domino.caleydo.org>. Interaction with the system is demonstrated in the accompanying video.

As shown in Figure 7, the main user interface consists of two parts: the *Domino Board* and the *Block Browser*. In addition to the two main interfaces, support views provide information about the data and current selections.

The *Domino Board* is the central visualization and interaction space of Domino, in which blocks can be arbitrarily positioned, manipulated, and combined using drag and drop. To handle setups with many related subsets, the board is realized as an infinite canvas.

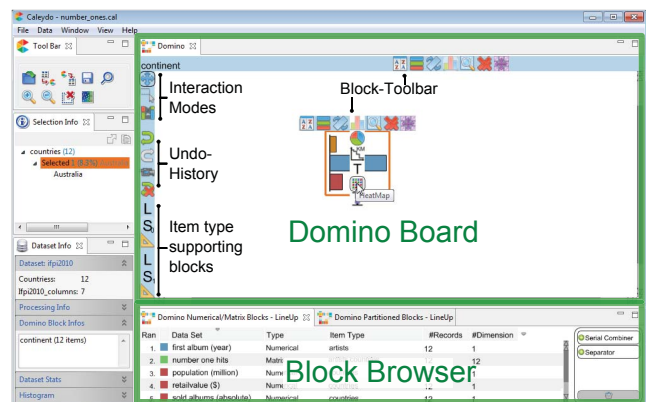


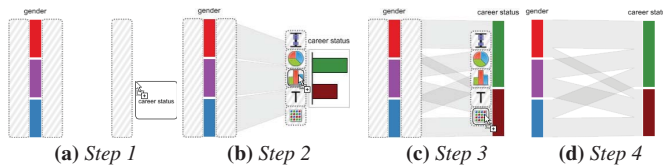
Fig. 7: Interface of the Domino prototype. The *Block Browser* presents lists of possible subsets organized by type. Users can drag subsets to arbitrary positions on the *Domino Board*, where they appear as a new block. Support views on the left show selections and meta information.

The *Block Browser* is a list-based interface for adding subsets to the board. Users can switch between two different lists: the first presents subsets that are associated with data (numerical and matrix subsets), and the second holds the partitioned subsets. Depending on the type of the subset, additional columns present metadata, such as size and item type. Users can add blocks to the board by dragging entries from the list to an arbitrary location on the board. The block browser is based on the LineUp [9] implementation.

Block-specific and global toolbars allow the analyst to trigger actions (e.g., sort, transpose, remove), switch between interaction modes (see Section 5.2), and provide access to a linear undo-history. In addition, item-type-specific supporting blocks are available, such as labels and rulers (Section 5.3).

## 5.1 Placeholders and Live Previews

When adding a new block to the *Domino Board*, the user needs to decide where to place it and how to represent it. To support this process, we make use of placeholders and live previews, as illustrated in Figure 8. Placeholders assist analysts in the process of combining blocks by indicating the various potential positions for placing a subset in the current Domino setup. When a user hovers over the placeholder, live previews [33] of the possible visualization outcomes are presented to help the user choose the appropriate technique for the subset.



**Fig. 8:** Usage of placeholders and live previews to add new blocks to Domino. When a block is dragged, placeholders (hatched areas) indicate possible drop locations. Live previews help users to choose a visualization technique.

While a block is dragged, all possible drop locations with a strong or weak block relationship are highlighted, as shown in Step 1 of Figure 8. The proximity of the placeholder to the block indicates the relationship degree, as introduced in Section 3.2.1. When the user drags a block or subset onto a placeholder, the applicable multiform visualization techniques are presented (Step 2). Hovering over one of the icons representing the techniques renders a live preview of the resulting visualization (Step 3). Dropping the block onto a specific icon confirms the selection and removes all other placeholders (Step 4).

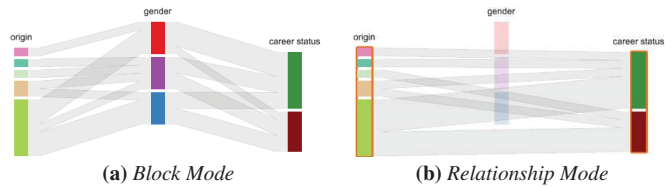
## 5.2 Interaction Modes

In our implementation of the Domino technique, analysts can switch between three different interaction modes by using the toolbar, as indicated in Figure 7. The changing background color of the toolbar area emphasizes the active interaction mode.

As blocks and their relationships are the primary visual elements in Domino, the **block interaction mode** is designed for adding, moving, combining, and removing blocks.

In the **item interaction mode**, users can select and explore individual items within the blocks and within block relationships. Depending on the visualization technique employed, the analyst can directly select items, such as columns or rows in a heatmap, an aggregation of items represented by a bin in a histogram or bar chart, or parts of relationship representations. Selected items are highlighted in all matching blocks and block relationships. While item block relationships highlight the whole matching line or point, the coarser granularities highlight only a portion, depending on the represented subset. In Figure 1, for example, we selected the artist *Whitney Houston*, highlighted in all blocks and block relationships.

Weak block relationships are shown for all blocks with matching item types. However, by default, a block relationship is culled automatically if it intersects with any other block or an artificially introduced separator. On the one hand, this avoids clutter and can be helpful to create independent sections on the board. On the other hand,



**Fig. 9:** Relationship interaction mode. (a) To reduce visual clutter, relationships between blocks that intersect with another block are culled. (b) By actively selecting blocks, users can reveal the otherwise hidden relationships. Relationships to unselected blocks are faded out.

this also prevents analysts from investigating block relationships of non-neighboring blocks. Therefore, the purpose of the **relationship interaction mode** is to let users explore these hidden relationships between blocks. In this mode, all block relationships between selected blocks are shown, regardless of whether they intersect with any other blocks. To reduce clutter, all unselected blocks are faded, as illustrated in Figure 9.

## 5.3 Scaling and Ruler

Accommodating a range of dataset sizes on various screen resolutions is challenging. Domino addresses scalability by its flexible zoom capabilities and initial scaling heuristics, which make blocks with various sizes fit on the board. In our implementation, blocks can be scaled freely in the horizontal or the vertical direction. When the user hovers over a block while performing a scaling operation, only this block and strongly related blocks are affected by the operation; otherwise, all blocks on the board are scaled.

While the scaling approach ensures flexibility to deal with datasets of various sizes, it hampers the user’s ability to compare the sizes of multiple independent blocks, since they may have different scaling factors. To address this problem, we provide *rulers* that illustrate how much space on the board corresponds to how many items. They also serve as global scaling factors for a specific item type. In Figure 1, rulers for *artists* and *countries* are shown in the bottom left corner. When a scaling operation is applied to a ruler, all 1D and 2D heatmap and bar chart blocks are scaled simultaneously.

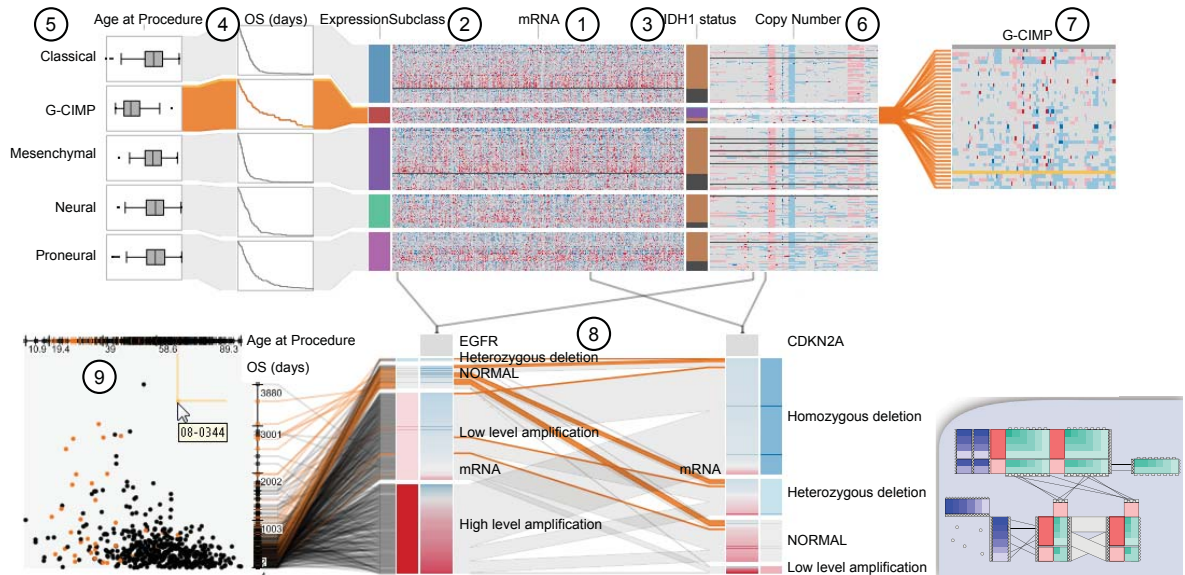
## 5.4 Subset Manipulation

Subsets can be created and manipulated in several ways. Set operations between blocks can be triggered by dragging a block onto another one. The desired operation (union, intersection, or set difference) is determined by dropping the block on an icon, similarly to choosing the visualization type as described in Section 5.1. Since relationships at the group and block granularity levels already encode set operations between two blocks, they can also be extracted into a new block by dragging the relationship representation to an open area on the board [17]. In a similar fashion, individual groups of a partitioned block can be extracted into a new block. Finally, we use a small widget for extracting selected items as new blocks, as shown in the bottom left corner of Figure 1. The selected items of the corresponding item type are represented by a bar whose size corresponds to the number of selected items. By dragging the bar to an arbitrary position on the board, the selected items are added as a new block.

As introduced in Section 3.3, Domino distinguishes between temporal and spatial provenance. While spatial provenance is encoded directly by the Domino visualization, temporal provenance is implemented by storing all operations in a linear history. Users can undo and redo operations, revert all operations at once, and replay them as an animation.

## 6 USE CASES

We demonstrate the utility of Domino in two use cases. In the first scenario, we utilize our technique to explore the music charts dataset introduced at the beginning of the paper (see Figure 1). In the second use case, we use Domino to characterize tumor subtypes.



**Fig. 10:** Key findings on subtypes in glioblastoma multiforme. Orange highlighting indicates patients with the G-CIMP subtype. (1) The mRNA gene expression matrix. In this heatmap, red and blue indicate expression levels higher and lower than the cohort average, respectively. Gray represents the average. (2) Partitioning block representing the tumor subtypes defined by Brennan et al. [3]. (3) The mutation status of the IDH1 gene. Purple represents mutations; orange represents patients without mutation, and gray represents missing data. (4) Patient survival illustrated by Kaplan-Meier plots. (5) Patient age as boxplots. (6) Copy number for each patient and gene sorted by genomic location. Blue and red hues represent copy number losses and amplifications, respectively. Gray represents the normal state of two copies per gene. (7) Block extracted from the copy number heatmap to highlight uncommon copy number patterns observed in the G-CIMP subtype. (8) Copy number and gene expression levels for CDKN2A and EGFR. (9) Scatterplot showing the relationship between patient age and overall survival time.

## 6.1 Music Charts

At the beginning of the analysis, we want to know whether there are any exceptional correlations between the count of number-one hits in a country and the artists' country of origin. We therefore begin by adding the *number-one hits* matrix block and partition the artists by their country of *origin* while partitioning the countries by *continent*. To inspect individual items, we switch to the item interaction mode. Looking at the partitioned matrix block, we observe that the band *U2* has the most number-one hits in their home country *Ireland*, while *ABBA*, for instance, is far more popular in *Germany* than in their home country *Sweden*. To understand the relevance of a number-one hit in a specific market, we add the *sold albums* numerical blocks as a bar chart below the matrix block and observe that the *US* leads by a wide margin. However, we assume that the absolute number of albums sold depends on the population size of a country. To confirm this, we add the *population* of the countries as an additional 1D scatterplot.

Another interesting question we seek to answer is how the count of *number-one hits* depends on the count of published *studio albums*. By adding the corresponding numerical blocks, we find that *Elton John* has released an impressive 33 albums. When looking at this number in the context of the years of his *first album* and his *start of career*, it becomes obvious that he has been successful in the industry for a long time. By looking at the other artists, we see that only *Elvis Presley* published an album earlier than *Elton John*. However, as *Elvis Presley* (presumably) passed away, we continue to look for additional artists that are no longer active. According to the *career status* block, five out of twelve are inactive. To examine the inactive artists more closely, we correlate the status with the artists' *gender*. We further investigate the distribution of the gender in the inactive group by performing a logical intersection operation, which shows us that only a single female artist is no longer active: *Whitney Houston*. Figure 1 presents a screenshot of our analysis, in which *Whitney Houston* as well as the *US* are selected and highlighted across all blocks.

## 6.2 Cancer Subtype Analysis

A second dataset we studied using Domino is the glioblastoma multiforme (GBM) dataset generated by *The Cancer Genome Atlas* (TCGA, <http://cancergenome.nih.gov>) project and described in

several publications by that consortium [3, 32, 34]. Glioblastomas are aggressive brain tumors, and patients with this type of cancer have a median expected survival time of around 12 months after diagnosis [32]. There are, however, different subtypes of GBM, which are driven by different molecular changes and associated with different patient survival times.

Our goal for this case study is to recapitulate some of the findings of the most recent TCGA study on GBM [3], which are presented in Figure 10. We start by adding the gene expression matrix (*mRNA*) to the board as a heatmap (1). This matrix contains data for over 12,000 genes and 528 patients, and each cell of the matrix corresponds to the activity of a given gene in the tumor sample from a given patient. Without clustering or sorting, the heatmap does not reveal any particular patterns. Using the gene expression subtypes published by Brennan et al. [3], we partition the heatmap into five groups: *Classical*, *G-CIMP*, *Mesenchymal*, *Neural*, and *Proneural* (2). We further combine the mutation status of the *IDH1* gene with the expression subtype block (3). Mutations in *IDH1* are known to play a role in GBM and for their association with the *G-CIMP* subtype. As Figure 10 illustrates, this relationship becomes immediately evident in Domino.

In order to compare patient survival times in the five gene expression subtypes, we drag the overall survival variable (*OS (days)*) from the block browser onto the Domino board and create a medium relationship between this and the partitioned gene expression matrix so that the survival times are partitioned by expression subtype (4). We select the Kaplan-Meier survival plot as the block visualization, which shows that the overall survival times for the *G-CIMP* group seem to be better than for the other groups. This is a known characteristic of both the *G-CIMP* subtype and the *IDH1 mutation* [3]. Another finding related to this subtype is that the patients in this group tend to be younger. We confirm this by visualizing the age of the patients in each expression subtype as a box plot next to the Kaplan-Meier plots, and observe a notable difference between the *G-CIMP* group and the four other groups (5).

Next, we study the copy number changes of genes in the patient genomes. We drag the copy number matrix for 560 patients and over 24,000 genes onto the board, creating a strong relationship between this block and the existing gene expression matrix block (6). Sort-



ing the genes by their location within the genome reveals a number of common large scale copy number losses and amplifications in GBM tumor genomes. As the copy number matrix is partitioned by gene expression subtypes, it also reveals that the copy number pattern for the *G-CIMP* is remarkably different from the other groups. We study these patterns more closely by extracting and enlarging the corresponding block of all genes and *G-CIMP* patients, noting what appears to be two different types of copy number patterns in this patient group (7). Two genes that are known to be affected by copy number changes in GBM are the tumor suppressor gene *CDKN2A* and the oncogene *EGFR*. We extract these genes from both the copy number and the gene expression matrix and create a combined block for each gene (8). We find that the copy number of *EGFR* is in fact highly amplified in close to 50% of the patients, and that this copy number change has an impact on the corresponding gene expression levels. For *CDKN2A* we observe that frequently either one copy is deleted (*heterozygous deletion*) or both copies are deleted (*homozygous deletion*), with the expected effect on the gene expression levels.

We are particularly interested in the patients with the *G-CIMP* subtype of GBM. In order to see their copy number status and gene expression levels for *CDKN2A* and *EGFR*, we select this patient group in the partitioned gene expression matrix (*mRNA*). Due to the highlighting of the patients in all blocks and bands, it is evident that overall the *G-CIMP* patients tend to have no or low-level copy number changes in the two genes. This indicates that the tumors of these patients are most likely driven by a mechanism that does not involve molecular alterations of *EGFR* or *CDKN2A*.

Finally, due to our earlier observation that patients in the *G-CIMP* group tend to be younger and have longer survival times, we are interested in whether there is an overall correlation between these two variables. We drag both the overall survival and patient age variables from the block browser onto the board. By visualizing them as axis plots with an orthogonal alignment, we create a scatterplot of the two variables (9). The scatterplot indicates that there might be a weak correlation between younger age and longer survival times. The selected *G-CIMP* patients are highlighted in orange, emphasizing the distinct age distribution in that group.

## 7 DISCUSSION

**Visualization Grammar and Templates** Domino provides a comprehensive toolset to assemble both established visualization techniques and novel combinations. The concept could be generalized to a visualization grammar [36] for the exploration of tabular datasets. The grammar could be utilized to define templates that serve as a blueprint for the creation of sophisticated visualizations. This would make it possible to customize a specific implementation of the Domino concept according to the template. In this approach, the feature set of Domino could be restricted to certain operations in order to allow users to create only specific kinds of composite visualizations, such as parallel coordinates or parallel sets, or more complex ones, depending on the task. Furthermore, a visual editor could be created that enables users to interactively define templates for specialized visualizations.

**Use of colors** Assigning unique colors to partitioned blocks is a challenging task. By default, we automatically assign colors by using predefined color schemes provided by ColorBrewer [10]. However, users can manually override the chosen colors if desired. The automatic color selection guarantees that groups belonging to the same partitioned block have different colors. However, depending on the number of partitioned blocks and contained groups, it might not be possible to choose different colors for all groups in all partitions. Assignment of the same color to semantically unrelated groups can cause confusion. A compromise would be to at least ensure that colors within a combined block are unique. However, this could result in situations where the color assignment to blocks needs to be adapted after the user interactively changes the configuration of combined blocks. As this change of colors would destroy the user's mental map, we opted to allow duplicate colors instead.

**Crossing lines and bands** As defined at the end of Section 3.1 on blocks, the item order within a partitioned block is undefined. When visualizing relationships between blocks, this can lead to unnecessary line crossings between two weakly related blocks that use the parallel item block relationship granularity. In the current implementation, one has to strongly relate a copy of the opposite block and apply it as secondary sorting criterion. This will reduce line crossings, since the item order within a group will match the item order of the opposite block. To remedy this issue, more sophisticated reordering operations could be applied [24, 23]. Relationships at the group granularity level can also suffer from a similar ordering problem, resulting in crossing of bands. Again, reordering strategies need to be applied to minimize the crossings.

**Partitioned matrix blocks** The presented technique introduces three different block types: partitioned, numerical, and matrix blocks. Matrix blocks can be treated as the two-dimensional version of a numerical block. Therefore, a logical consequence would be to also introduce an analogous two-dimensional block type for a partitioned block. However, a partitioned matrix block is currently not supported in Domino, as the mapping of an item pair ( $f((i, j) \in I_1 \times I_2) \mapsto g \in P$ ) to a group would result in a biclustering [5] in which one item of  $I_1$  can be mapped to multiple groups. This prohibits splitting up the block horizontally or vertically, which is needed for propagating the partitioning to strongly related neighboring blocks. However, in Domino such scenarios can be handled by representing every group as a single block in the layout, where overlapping items between blocks are visualized using lines or bands [31].

**Scalability and Performance** We ensure visual scalability through the flexible zooming concept described in Section 5.3. Regarding data scalability, the size of matrix blocks is the critical issue. In the cancer subtype analysis case study presented here, matrix blocks have up to  $560 \times 24,000$  items, demonstrating Domino's applicability to larger datasets. Item relationships for large subsets are, however, a performance issue, since individual lines for each data item need to be drawn. This issue can be addressed by using a higher relationship granularity.

## 8 CONCLUSION AND FUTURE WORK

We have presented Domino, a novel technique that visualizes subsets together with associated data, and relationships between them. Domino allows users to explore, extract, and manipulate subsets of multiple item types at various levels of granularity. Analysts can not only rapidly assemble common visualization techniques, but also create new combinations tailored to specific tasks and datasets. The prototype implementation uses live previews and placeholders to support users in managing the wide range of possibilities that the technique offers.

In our implementation, placeholders support analysts in creating combined blocks by indicating possible positions for placing a subset in the current Domino setup. In the future, we plan to enrich the placeholder interaction concept with further guidance capabilities that suggest potentially interesting relationships. Depending on the task at hand, possible drop positions could be ranked according to similarity measures, such as the correlation between groups of two partitioned blocks. Another approach to representing partitioned data is to interleave or overlay partitioned data to create a single block, which would complement the currently implemented approach of creating one block per group.

## ACKNOWLEDGMENTS

This work was supported in part by the Austrian Research Promotion Agency (840232), the Austrian Science Fund (J 3437-N15), the Air Force Research Laboratory and DARPA grant FA8750-12-C-0300, and the United States NIH/National Human Genome Research Institute (K99 HG007583).

## REFERENCES

- [1] B. Alsallakh, L. Micalef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. Visualizing sets and set-typed data: State-of-the-art and future challenges. In *Eurographics conference on Visualization (EuroVis)–State of The Art Reports*, pages 1–21. Eurographics, 2014.
- [2] L. Bavoil, S. Callahan, C. Scheidegger, H. Vo, P. Crossno, C. Silva, and J. Freire. VisTrails: enabling interactive multiple-view visualizations. In *Proceedings of the IEEE Conference on Visualization (VIS '05)*, pages 135–142. IEEE, 2005.
- [3] C. W. Brennan, R. G. W. Verhaak, A. McKenna, B. Campos, H. Nounshmehr, S. R. Salama, S. Zheng, D. Chakravarty, J. Z. Sanborn, S. H. Berman, R. Beroukhi, B. Bernard, C.-J. Wu, G. Genovese, I. Shmulevich, J. Barnholtz-Sloan, L. Zou, R. Vegesna, S. A. Shukla, G. Ciriello, W. K. Yung, W. Zhang, C. Sougnez, T. Mikkelsen, K. Aldape, D. D. Bigner, E. G. Van Meir, M. Prados, A. Sloan, K. L. Black, J. Eschbacher, G. Finocchiaro, W. Friedman, D. W. Andrews, A. Guha, M. Iacocca, B. P. O'Neill, G. Foltz, J. Myers, D. J. Weisenberger, R. Penny, R. Kucherlapati, C. M. Perou, D. N. Hayes, R. Gibbs, M. Marra, G. B. Mills, E. Lander, P. Spellman, R. Wilson, C. Sander, J. Weinstein, M. Meyerson, S. Gabriel, P. W. Laird, D. Haussler, G. Getz, L. Chin, and TCGA Research Network. The somatic genomic landscape of glioblastoma. *Cell*, 155(2):462–477, 2013.
- [4] D. B. Carr, R. J. Littlefield, and W. L. Nicholson. Scatterplot matrix techniques for large n. In *Proceedings of the Symposium on the Interface of Computer Sciences and Statistics*, pages 297–306. Elsevier North-Holland, 1986.
- [5] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, page 93–103. AAAI Press, 2000.
- [6] J. H. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2310–2316, 2011.
- [7] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1192–1199, 2007.
- [8] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. GraphTrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, page 1663–1672. ACM, 2012.
- [9] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.
- [10] M. Harrower and C. A. Brewer. ColorBrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.
- [11] J. A. Hartigan and B. Kleiner. Mosaics for contingency tables. *Proceedings of the Symposium on the Interface*, page 268–273, 1981.
- [12] H. V. Henderson and P. F. Velleman. Building multiple regression models interactively. *Biometrics*, 37(2):391–411, 1981.
- [13] J.-F. Im, M. McGuffin, and R. Leung. GPLOM: the generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013.
- [14] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization (Vis '90)*, pages 361–378, 1990.
- [15] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '12)*, pages 1–8. IEEE, 2012.
- [16] W. Javed and N. Elmqvist. ExPlates: spatializing interactive analysis to scaffold visual exploration. *Computer Graphics Forum (EuroVis '13)*, 32(3pt4):441–450, 2013.
- [17] J. Kolojechick, S. F. Roth, and P. Lucas. Information appliances and tools in visage. *IEEE Computer Graphics and Applications*, 17(4):32–41, 1997.
- [18] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.
- [19] A. Lex, H.-J. Schulz, M. Streit, C. Partl, and D. Schmalstieg. VisBricks: multiform visualization of large, inhomogeneous data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2291–2300, 2011.
- [20] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE, 2010.
- [21] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):1027–1035, 2010.
- [22] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. StratomeX: visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Computer Graphics Forum (EuroVis '12)*, 31(3):1175–1184, 2012.
- [23] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, page 89–96, Washington, DC, USA, 2004. IEEE Computer Society.
- [24] A. Pilhofer, A. Gribov, and A. Unwin. Comparing clusterings using bertin's idea. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '12)*, 18(12):2506–2515, 2012.
- [25] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*, pages 318–322. ACM, 1994.
- [26] J. T. Rich, J. G. Neely, R. C. Paniello, C. C. J. Voelker, B. Nussenbaum, and E. W. Wang. A practical guide to understanding kaplan-meier curves. *Otolaryngology–Head and Neck Surgery*, 143(3):331–336, 2010.
- [27] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive sankey diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pages 233–240. IEEE, 2005.
- [28] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '07)*, pages 61–71. IEEE, 2007.
- [29] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, 12(5):733–740, 2006.
- [30] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2249–2258, 2011.
- [31] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter. Furby: Fuzzy force-directed bicluster visualization. *BMC Bioinformatics*, 15(Suppl 6):S4, 2014.
- [32] The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.
- [33] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum*, 32(3pt2):191–200, 2013.
- [34] R. G. W. Verhaak, K. A. Hoadley, E. Purdom, V. Wang, Y. Qi, M. D. Wilkerson, C. R. Miller, L. Ding, T. Golub, J. P. Mesirov, G. Alexe, M. Lawrence, M. O'Kelly, P. Tamayo, B. A. Weir, S. Gabriel, W. Winckler, S. Gupta, L. Jakkula, H. S. Feiler, J. G. Hodgson, C. D. James, J. N. Sarkaria, C. Brennan, A. Kahn, P. T. Spellman, R. K. Wilson, T. P. Speed, J. W. Gray, M. Meyerson, G. Getz, C. M. Perou, D. N. Hayes, and TCGA Research Network. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110, 2010.
- [35] C. Viau and M. J. McGuffin. ConnectedCharts: explicit visualization of relationships between data graphics. *Computer Graphics Forum*, 31(3pt4):1285–1294, 2012.
- [36] L. Wilkinson. *The grammar of graphics*. Springer, 2nd edition, 2005.
- [37] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):1001–1008, 2009.