TACO: Visualizing Changes in Tables Over Time

Christina Niederer, Holger Stitz, Reem Hourieh, Florian Grassinger,
Wolfgang Aigner, and Marc Streit

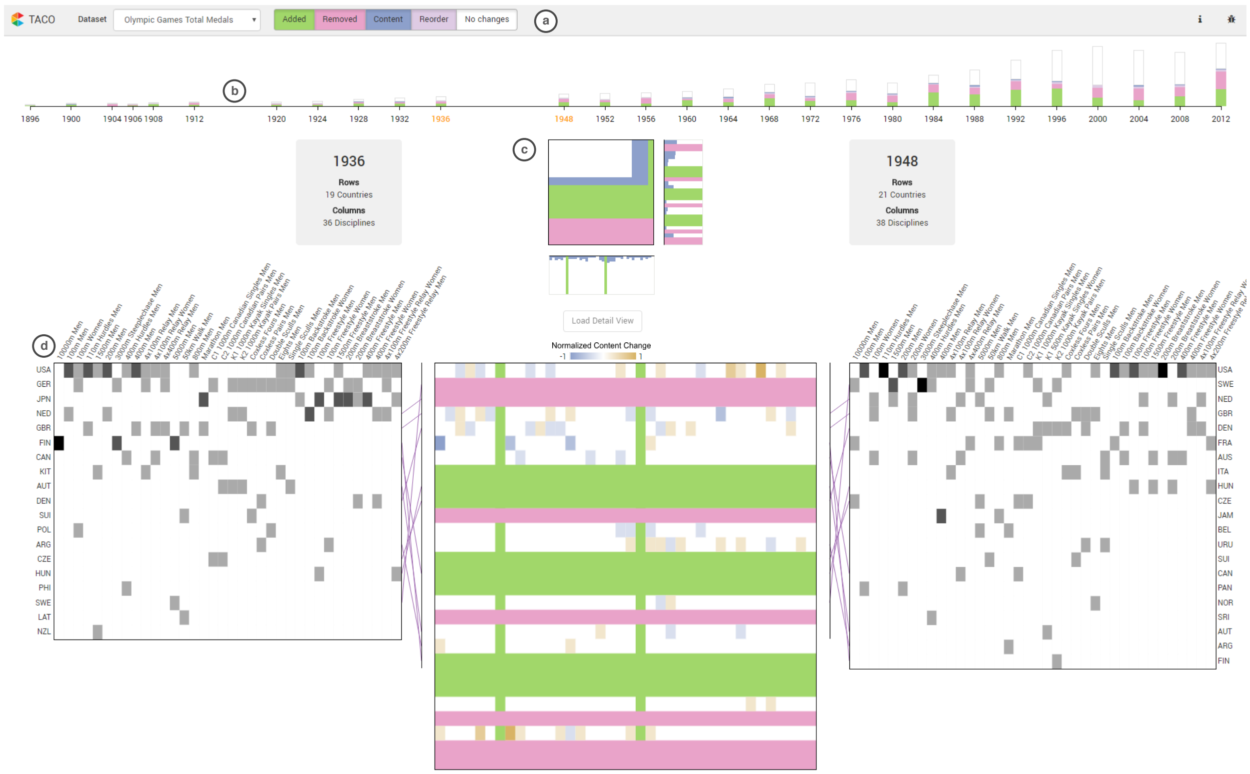

Fig. 1: *TACO visualizes differences between Summer Olympic Games medal tables over time at multiple levels of detail. The interface is structured along three levels following an overview+detail concept. (a) Change switches in the header bar allow users to hide and show specific types of changes in the visualizations. (b) At the overview level, we present a timeline that shows stacked bar charts for indicating the temporal progression of the medal table between 1896 and 2012. (c) The second level visualizes aggregated changes for the two selected time points, 1936 and 1948, as a 2D ratio chart with attached diff histograms. (d) At the third and most detailed level, we show a difference heatmap together with raw heatmaps for the two selected medal tables. Link to TACO state shown in this figure:* `http://vistories.org/taco-olympic-games`

Abstract—Multivariate, tabular data is one of the most common data structures used in many different domains. Over time, tables can undergo changes in both structure and content, which results in multiple versions of the same table. A challenging task when working with such derived tables is to understand what exactly has changed between versions in terms of additions/deletions, reorder, merge/split, and content changes. For textual data, a variety of commonplace "diff" tools exist that support the task of investigating changes between revisions of a text. Although there are some comparison tools which assist users in inspecting differences between multiple table instances, the resulting visualizations are often difficult to interpret or do not scale to large tables with thousands of rows and columns. To address these challenges, we developed TACO, an interactive comparison tool that visualizes the differences between multiple tables at various levels of detail. With TACO we show (1) the aggregated differences between multiple table versions over time, (2) the aggregated changes between two selected table versions, and (3) detailed changes between the selected tables. To demonstrate the effectiveness of our approach, we show its application by means of two usage scenarios.

Index Terms—Table comparison, matrix, difference visualization.

1 Introduction

Understanding tabular data is essential in many domains, such as accounting, biology, and computer science. An important task when making sense of such data is to investigate the difference between multiple versions of a table, for example, to detect modifications in monthly payroll tables or to observe differences in multiple biological exper-

- *Christina Niederer and Florian Grassinger are with St. Pölten University of Applied Sciences, Austria. E-mail: {firstname.lastname}@fhstp.ac.at.*
- *Holger Stitz, Reem Hourieh, and Marc Streit are with Johannes Kepler University Linz, Austria. E-mail: {firstname.lastname}@jku.at.*
- *Wolfgang Aigner is with St. Pölten University of Applied Sciences, Austria and TU Wien, Austria. E-mail: {firstname.lastname}@fhstp.ac.at.*

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

iments. We differentiate between four types of changes: **structural** (add/remove), **content**, **reorder**, and **split/merge** changes.

The primary contribution of this paper is TACO (***TA**ble **CO**mparison*), a novel visual comparison tool. First, it calculates the difference between pairs of tabular data. Based on that it provides interactive visualizations to encode the differences over time at multiple levels of detail for different change types.

We demonstrate the effectiveness of our tool by describing two usage scenarios: Our first dataset, which we also use as a guiding example throughout the paper, consists of the medal tables of the Summer Olympic Games from 1986 to 2012. The second usage scenario demonstrates the use of TACO for visualizing the evolution of a large biomedical dataset over the course of a multi-year project called *The Cancer Genome Atlas (TCGA)*[1]. This domain problem was our motivation for developing TACO.

To better understand the problem domain and associated requirements, we start by characterizing tabular data. We then continue with a discussion of the four change types and related user tasks that should be supported by an effective table comparison solution.

2 Tabular Data Characterization and Change Types

A table is a dataset composed of *rows* and *columns*. The intersection of a row and a column identifies a *cell*. In tables, each row and column is identified by a key. Each cell is therefore identified by a pair of unique row and column identifiers. Rows and columns have an order that can be meaningful in some applications. For example, the order plays a major role in statistics and ranking applications. In our work, we consider the order of rows and columns as an important characteristic.

The definition of tabular data constrains how data is organized, but it neither constrains the content of the table nor the data types contained. In this paper, we differentiate between *heterogeneous* and *homogeneous* tables. In a heterogeneous table each column (or row) can have a different data type and different semantics. In and a homogeneous table all columns and rows have the same data type and semantics and thus all cells contain values of the same data type. A homogeneous table can also be referred to as matrix. Table columns can be either categorical or ordered. Categorical data does typically not imply an order, while ordered data can be further subdivided into ordinal and quantitative data [22]. In this work, we focus on quantitative homogeneous tables whose columns have the same quantitative data type and to which both ordering and arithmetic operations can be applied. However, our concept developed can also be applied to heterogeneous tables with different value ranges and units of the cells, as discussed later on in Section 9.

In standard tables, four different **types of changes** can be observed in the data:

1. **Structural changes**, where a row or column is either *added* or *removed*. In our Olympic Games dataset, for instance, both the participating countries and the disciplines change over time.
2. **Content changes** resulting from modifying the values of cells that have the same row and column identifiers. Some data operations affect the values of entire columns or rows, such as normalizations, others (e.g., manual edits and corrections) just single cells.
3. **Reorder changes**, where a row or a column is shifted from its original position [21]. This does not include shifts resulting from additions or removals. An example of an operation that introduces reorder changes is a hierarchical clustering applied to rows or columns. In other cases, the order changes naturally over time, for instance, in the Olympics medal table, which is ranked by the overall number of medals per country.
4. **Merge and split changes** caused by multiple single rows (or columns) that are joined into one, or single rows (or columns) that are divided into multiple rows (columns). In biology, for instance, experiments are often performed multiple times, which results in multiple data subsets that are later joined during preprocessing by averaging the values. However, this task is rare compared to the three change types introduced above.

3 User Tasks

From a problem-oriented perspective, the main questions related to comparison tasks over time are: (1) Did changes happen, and—if so—where (existence and temporal location)? (2) How much did change (amount of change)? (3) What did change (details about different types of change)? (4) Where did the changes occur (details about type and location of changes)? Based on this and a series of discussions with domain experts in biomedical data analysis and our own experiences as data scientists who work extensively with tables, we identified a number of user tasks that an effective table comparison technique needs to address. We use these tasks throughout the paper to discuss existing tools and to introduce our own solution.

T1 **Identify the types of changes**. Users should be able to locate the four types of changes and identify them *row-wise*, *column-wise*, and *cell-wise*.

T2 **Compare multiple table versions over time**. Users should gain an overview of aggregated differences between multiple table versions over time and discover how many changes of which type occurred between two consecutive table versions.

T3 **Compare one table at two particular time points**. Users want to investigate the detailed changes of all types between different versions of the same table at two points in time.

T4 **Present raw data tables and meta-data**. Users should have the possibility to see the changes in the context of the raw data tables, including additional information such as structure, size, name, and date of creation.

4 Related Work

Gleicher *et al.* [12] divided the design space for comparative visualization into the three basic categories: *juxtaposition*, *superposition*, and *explicit representation*. These fundamental techniques can be used individually or in combination.

Juxtaposition

This design presents the objects to be compared separately or next to each other in either time or space. If juxtaposition is used properly, it can help users to discover repeated patterns and differences between compared objects [29].

Superposition

Using this method, multiple objects are presented in the same coordinate system or spatial substrate. The advantage of superposition is that proximity explicitly encodes similarity. However, the scalability of this approach is limited, as three or more objects in the same visualization already need interaction techniques to clarify differences.

Explicit Representation (Encoding)

This design approach computes the relationships in terms of differences between objects and encodes them visually. The advantage of this approach is that the viewer does not need to make a mental comparison or find the difference, as it has already been calculated. This approach requires a clear definition of the relationships between the compared objects in order to compute and explicitly show the resulting differences.

A few approaches exist that already combine successfully all three basic categories defined by Gleicher *et al.* [12]. Tominski *et al.*[28] presented interaction concepts for comparing tabular data that are inspired by natural behavior, although not all interaction techniques proposed are suitable for large datasets. Their side-by-side comparison, however, is also applicable to TACO. Further techniques exist that allow changes to be visualized and facilitate comparison tasks, such as interaction, analytical and statistical calculations, and animation [12].

To investigate the state-of-the-art, we start with discussing comparative visualization techniques mainly from scientific literature that aid comparison of various data types. In addition, we examine common diff tools and libraries for investigating differences in tabular data.

[1] http://cancergenome.nih.gov/

4.1 Comparative Visualization of Tabular Data

While much work exists on the comparison of graphs [3, 5, 7, 30], time series data [26], or image data [24], techniques designed for comparing tabular data are rare.

However, graph comparison techniques that employ adjacency matrices as a base representation (e.g., [3, 5, 7, 30]) are similar to our work in the sense that matrices can be interpreted as specific type of tables. *Matrix Wave* [30], for example, focuses on finding a path between multiple matrices rather than finding common patterns. Superposition is used to combine the original data and the explicit encoding of the difference in one visualization. The proposed method works well for finding paths and relations between time sequence data. However, it does not scale to large tables and tables that have many more columns than rows, or vice versa.

Another common approach is to visualize the datasets (or data subsets) as heatmaps and show them next to each other, i.e., using juxtaposition. The pioneer work on *VisDB* [17] for comparing database query results is an early example of a pixel-based approach that juxtaposes multiple heatmaps. *iHAT* [15] is a heatmap-based approach in which the user can aggregate rows and columns interactively. Techniques such as *Matchmaker* [19], *VisBricks* [18], and *StratomeX* [20] arrange the tabular subsets in an axes-based layout and draw explicit connections between the table representations to visualize relationships between items. While such an approach works well for comparing structural and reorder changes, investigating content and merge/split changes (T1) is difficult because the user must make the comparison mentally, which causes high cognitive load. Adding interactions, such as details-on-demand or linking-and-brushing, can help to alleviate the problem but the comparison still remains tedious and is therefore not feasible.

An additional limitation is that most visualizations lack the ability to perform row-wise, column-wise, and cell-wise comparison of tables simultaneously. For juxtaposition-based approaches this is mostly due to the fact that explicit relationship encoding only allows for being visually connected in one direction.

Apart from the mentioned approaches, also superposition methods have been applied [8, 16]. They visually represent matrices as heatmaps and project pairwise comparison vectors between matrices onto a low-dimensional space, for example, using MDS. However, these approaches do not cover the temporal aspects of data changes (T2).

4.2 Diff Tools

The primary purpose of difference or data comparison tools is to calculate and display the similarities or differences between datasets of various types. Most of the available tools are limited to particular file formats, such as *ExcelCompare*[2] to Microsoft Excel Spreadsheets, *Daff*[3] to CSV files, and *AQT*[4] to relational database tables. *DiffKit*[5] is one of the few exceptions that supports all of the file formats mentioned. All these tools have in common that they take two datasets as input, compute the differences between them based on a number of metrics, and finally present the result textually or visually. The final outputs of these approaches are usually textual, list-based representations of the differences or simple highlighting approaches that color-code the changes at the row, column, and cell levels. However, neither lists nor highlighting approaches scale to large tables with hundreds or even thousands of rows or columns or to a large number of changes. Furthermore, most of the existing tools focus on content and structural changes only and do not handle reorder and merge/split operations, thus failing to fulfill Task T1. Moreover, they only support pairwise comparison between two tables (T3), but are not able to provide an overview of multiple table versions over time (T2).

4.3 Summary

In the previous paragraphs, we have presented a number of different visual comparison methods. Examples for all three basic comparison categories of juxtaposition, superposition, and explicit representation have been found. However, to the best of our knowledge, there is no approach available that is able to address all necessary change types presented in Section 2 and all the tasks formulated in Section 3. To fill this gap, we developed TACO to facilitate effective visual comparison of potentially large tabular datasets over time.

[2]`https://github.com/na-ka-na/ExcelCompare/`

[3]`http://paulfitz.github.io/daff/`

[4]`http://querytool.com/`

[5]`http://www.diffkit.org/`

5 Change Calculation

A review of existing tools and libraries showed that none of them satisfies our need for calculating the difference between tables with respect to the four change types. To address this important prerequisite that constitutes the basis of our comparative visualization technique, we developed a method for calculating differences based on the identifiers of table rows and columns.

The result of a pairwise table comparison (Table A vs. Table B) is a union table containing all rows and columns that are present in either one of the input tables. The union table is then the basis for the visual representations in TACO. Figure 2 illustrates how changes of all four types are reflected in the union table. We detect structural (add/remove), merge and split, and reorder changes by matching row and column IDs from both input tables and then flag them accordingly in the union table. To cover content changes as well, we subtract all cells that belong to rows and columns that are present in both input tables (i.e., the intersection), which yields cell-based difference values. Finally, we handle reorder changes by storing the distance in position for each column and row with respect to the two input tables, ignoring added and removed rows and columns.

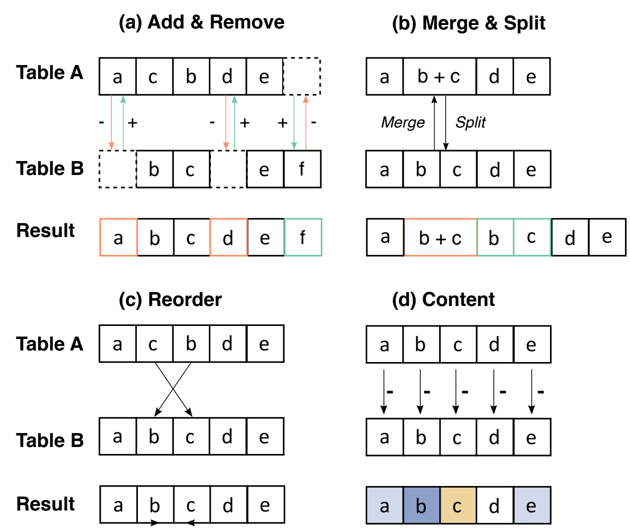

Fig. 2: *Effects of changes in two input tables on the union table. A and B are combined into a union difference table that is used as a basis for all change visualizations in TACO.*

6 Visualization Concept

As presented in Section 3, our design is guided by a set of representative user tasks. The design is additionally based on a series of discussions with domain experts and grounded in common visualization theory and concepts. In order to address user needs, we developed a multi-level overview+detail concept. TACO allows users to gradually add more focused and detailed views from top to bottom along three levels of detail (see Figure 3). It starts from an aggregated, time-oriented overview along a timeline (Figure 3.1). After selecting two table versions of interest, an intermediate level is revealed that provides meta-information on the selected data tables and the distribution of individual change types (Figure 3.2). Finally, an in-depth comparison view based on heatmaps is provided that allows changes to be investigated down to the level of individual cell values (Figure 3.3). Below, we discuss the individual representation levels in more detail, explain design rationales, and introduce the overall interaction concept.

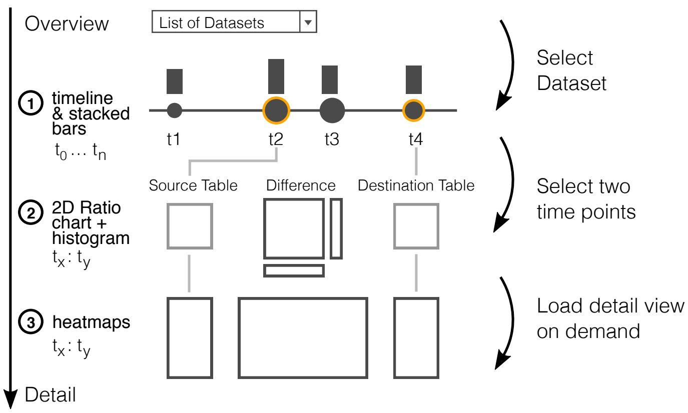

Fig. 3: *TACO allows users to compare changes in tables at multiple levels of detail. (1) Aggregated changes between two consecutive time points are represented as stacked bar charts that are arranged along a timeline. The overall bar height reflects the size of the calculated diff table. (2) Selecting two particular time points allows users to compare the changes and distribution in more detail using a 2D ratio chart and additional histograms for rows and columns. (3) The most detailed level includes heatmap representations of the two selected raw data table versions (left and right) and a diff heatmap that provides details on the location and amount of change at the cell level (center).*

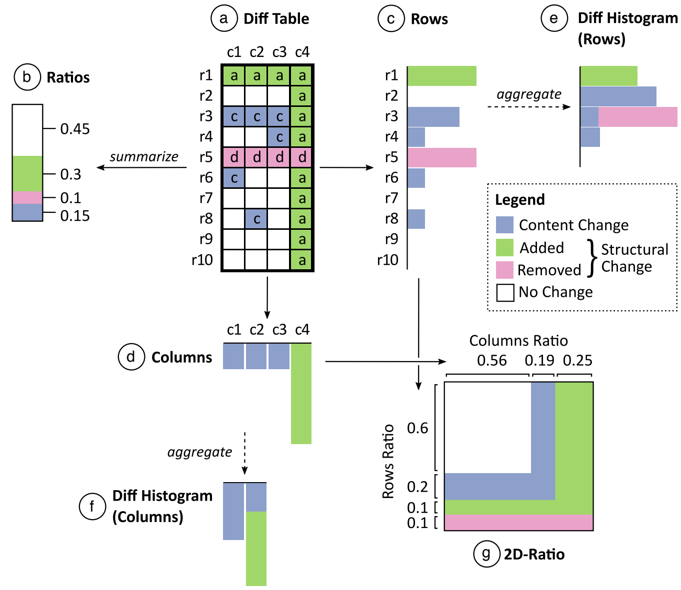

Fig. 4: *The difference between two tables is visualized as (a) a difference table. Changes are summarized on a per cell basis and visualized as a (b) bar plot. The diff table can be aggregated for (c) row and (d) column directions separately. (e,f) Further aggregation for one direction is achieved by binning, and shown as a histogram. Summarizing changes for rows and columns results in a (g) 2D ratio chart.*

6.1 Multi-Level Change Aggregation

As formulated in Task T2, a fundamental requirement is to examine the development of a tabular dataset over time. Summarization helps in finding patterns but comes at the price of losing details. Aggregation is needed to allow table comparison over time without overwhelming the user with too many details. In order to accomplish this, we first perform pairwise table comparisons and then summarize the different change types. A difference ratio between two tables is calculated by the ratio of the changed cells to all cells in the diff table, which results in a value between 0 and 1, where 0 equals no changes and 1 means that all cells are changed. Changes of different types can also be summarized by counting the cells that are affected by a specific change type and normalizing that number by the total number of cells in the diff table.

On the detail level, each pairwise comparison results in a difference table (diff table) in which change types are highlighted in color at the cell level (see Figure 4(a)). We apply different levels of aggregation operations to the diff table to represent the differences in a summarized way. First, aggregation is applied based on either dimension of the table individually, which means that the numbers of affected cells either per row or per column in the diff table are counted and shown as histograms, as illustrated in Figures 4(c) and 4(d), respectively. Hence, we retain information about the table dimensions and (approximate) locations of the changes. To accomplish this single-dimension aggregation, we consider the changes in only one dimension (e.g., row) while ignoring the changes in the other (e.g., column). Second, further aggregation can be achieved by aggregating the histogram data evenly into bins (see Figure 4(e,f)). The number of bins is determined by the histogram height (for rows) or width (for columns) and can be aggregated to a single pixel if necessary. This makes the representation more compact at the cost of less precision in terms of change location. Third, further summarization is performed simultaneously on both table dimensions to provide a more compact overview of relations of the different change types. In doing so, we lose location information completely, but retain change dimension information, using a 2D ratio chart as illustrated in Figure 4(g). Fourth, the most compact overall summary is achieved via a stacked bar chart that summarizes the ratios of change types globally at the cell level without information about rows or columns (see Figure 4(b)).

6.2 Visualization Components

To support Tasks T2 and T3, the TACO concept uses multiple levels, allowing the user to analyze the data at three levels of detail, as shown in Figure 3. This concept is based on Shneiderman´s *Visual Information Seeking Mantra* [25]: *"Overview first, zoom and filter, then details-on-demand"*. Zooming means either *zooming-in* and *zooming-out* or a shift of the user cognitive focus from one point in the view to another [11]. The second meaning is more relevant to what we suggest in this work.

Additionally, the user can selectively show or hide different change types using the control bar in the header (see Figure 1; Task T1). The control bar is visible at all detail levels of the interface and consists of a series of toggle buttons for the different change types. The buttons for toggling the display of the different change types are mainly interaction elements but also act as a legend of the colors used. We use a consistent color scheme throughout the whole interface, including the visual encoding of changes in the diff heatmap (6). The following colors are used to distinguish between change types: no change = (white), content = (blue), added = (green), removed = (pink), reorder = (purple). To avoid problems with red and green hues for colorblind people, we use colorblind-safe qualitative colors from Colorbrewer[6].

In the following sections we introduce the visual encoding and design decisions for each of the three detail levels.

6.3 Level 1: Change Overview Timeline

In the overview level illustrated in Figure 3.1, the user can compare one table version to the consecutive one. We use a timeline to visualize the changes over time (addressing T2), in which the user can see the temporal progression of the changes in terms of the aggregated amount of changes and when these changes occurred in time. A particular table version is shown as a labeled tick, and labels on the timeline are spaced according to the time of change.

In early design iterations, we considered using a projection-based representation as overview that does not take the time aspect into account but allows multiple tables to be compared simultaneously. However, in discussions with domain experts we received the feedback that showing incremental changes over time (Task T2) is more important. Further, we decided to use stacked bar charts for summarizing the changes for each table version with respect to the previous time point in a compact manner. Using vertical stacked bar charts rather than alternatives such as star plots or horizontal bar charts allows the user

[6]http://colorbrewer2.org/

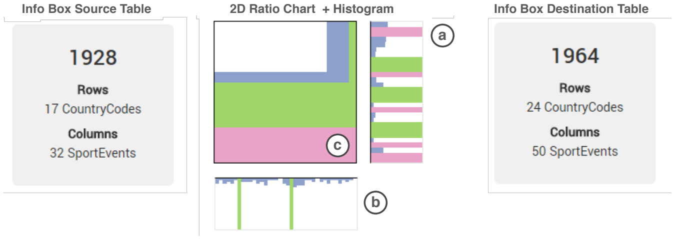

Fig. 5: *Aggregated pairwise comparison visualization (level 2) with info boxes showing meta-information describing the source and destination tables. Difference histogram for (a) rows and (b) columns; (c) 2D ratio chart presenting aggregated ratios of change types in both row and column direction.*

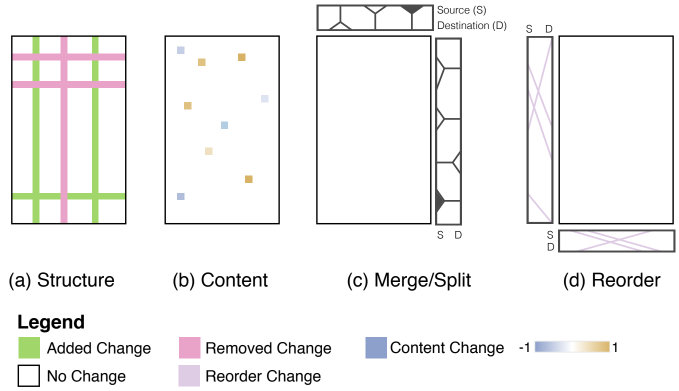

Fig. 6: *The four possible change types and their visual encoding in the diff heatmap.*

to compare the total number of changes for the different change types between multiple table versions over time (Task T2).

Reorder changes are disabled by default in the stacked bar chart visualization. This avoids a skewed representation, because reordered columns/rows can also have associated content changes. Hence, the sum of counted cells will exceed the number of total cells of the table. One possible way to avoid misinterpretation is to visualize reorder and content changes and introduce an additional color for the overlapping parts in the stacked bar chart. However, this may confuse the user and makes interpretation more difficult [14]. To address this problem, we allow the user to see reorder changes on demand as a segment superimposed on the content changes in the stacked bar chart.

Although each stacked bar represents the changes between two table versions based on the previous time point, we decided to place them on top of the timeline ticks to visually emphasize the points in time when a new table version was created and avoid the misleading impression that some change actually happened in the period between the ticks.

Once the user decides to investigate a particular version in more detail, it can be selected as a source table, and the subsequent selection sets the destination table by clicking the time point on the timeline.

6.4 Level 2: Aggregated Pairwise Comparison

After selecting two time points on the timeline, the differences between the associated tables are shown in more detail in an aggregated view (see Figure 3.2).

As illustrated in Figure 5, we position two info boxes to the left and right of the aggregated diff visualization in the center of the interface to fulfill Task T4. These boxes contain meta-information about the selected tables such as date, time, name, and size of the table version. Further, we show the following three aggregated diff visualizations, addressing Task T3:

- A *row diff histogram* presenting an aggregation of the difference in all rows into b_r bins. Figure 5(a) shows an example with 30 bins.
- A *column diff histogram* presenting an aggregation of the difference in all columns into b_c bins. Figure 5(b) shows an example with 30 bins.
- A *2D ratio chart* presenting aggregated ratios of the change types in both row and column dimension. Figure 5(c) shows an example with three types of changes in both rows and columns.

Each bin in a diff histogram is represented as a stacked bar where each part represents a type of change. When the user hovers over one bin in a diff histogram, percentages of the change types are shown in a tooltip. Upon user selection, the third level of detail is loaded in the form of a detailed pairwise comparison.

6.5 Level 3: Detailed Pairwise Comparison

On the one hand, the two selected raw data table versions are visualized side-by-side as heatmaps (see Figure 1 bottom left and right). A gray-scale color palette is used to represent cell values, as required for Task T4. This supports a juxtaposition comparison [12]. On the other hand, differences are shown directly as a color-coded diff heatmap in the center between the selected table versions. This enables an explicit representation (encoding) [12] which frees the user from performing mental comparisons between the two table versions. Regardless of which table version is chosen first on the timeline, the one shown on the left is always the source table (older one), and the destination table (more recent one) is the one on the right. Note that the source and target encodings of the raw data may be of limited use, in particular, if no meaningful order that might make patterns in the data visible has been applied to the input dataset.

Textual labels are added for both rows and columns in the raw table heatmaps. These facilitate relation to the original table's structure and content. Using a heatmap to represent a table has the advantage of better scalability to larger datasets. However, larger datasets turn a heatmap into a pixel-based visualization technique, where one or even multiple rows or columns are represented by a single pixel line. Textual labels are therefore available only for smaller table sizes and will be omitted for larger ones.

The diff heatmap visualizes the common (union) parts in both tables compared and changes happening between table versions using different visual attributes that represent the possible change types. Hovering over a cell in the diff heatmap shows the corresponding row ID, column ID, and the normalized change values. Further, interactive view coordination is applied to connect source, diff, and destination heatmaps. This means that, when a table cell is selected in each of the heatmaps, the corresponding cells are highlighted in all other views.

The following visual encoding concepts are used to represent different change types: Structure and content changes are indicated using colors in the diff heatmap, as illustrated in Figure 6(a) and (b). Merge changes are encoded as Y-connectors (Figure 6(c)) and reorder changes as slope graphs in a separate area between the heatmap and the diff heatmap (Figure 6(d)).

In the following paragraphs we discuss the individual visual encoding concepts for the diff heatmap.

Visualizing Structural Change

Structural changes are encoded as colored horizontal or vertical lines for rows and columns, respectively. Color hues indicate the type of structural change (see Figure 6(a)). The colors green and red are commonly used in the literature to indicate addition and removal operations [6]. Users associate green color with a positive value, growth, etc., which makes the color suitable for representing add operations. In contrast, red—in our case pink, since colorblind-safe colors are used (see 6.2)—is perceived as a negative value, which can be associated with losing information from deletions.

Visualizing Content Change

We use normalized quantitative differences along table columns to represent value changes within a table cell and apply a diverging color scale to illustrate the extent of the value difference. Starting from white, which represents no change, positive value changes are shown using

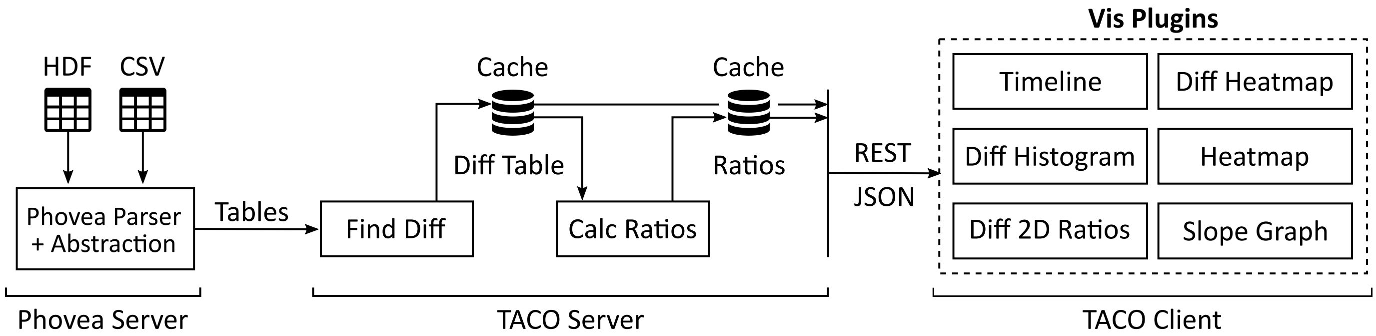

Fig. 7: *Client-server architecture of the TACO system illustrating the interplay between the different components. Input datasets in the HDF or CSV file format are prepared as Phovea tables. The comparison process between two tables results in a difference table, which is cached and used as input to the ratio calculation. Both the ratios and the difference table are encoded in the JSON format and transferred to the client, where several plugins visualize the data.*

different brightness levels of brown on the one, and different brightness levels of blue on the other side to represent negative value changes. Although color is not the best possible choice to represent quantitative differences, we decided on this design trade-off because other variables such as size or length cannot be encoded in a visualization of a diff table while satisfying scalability and consistency requirements. Figure 6(b) shows an example diff heatmap with brown cells encoding positive value changes and light blue cells encoding negative value changes. The user can emphasize cells with low value changes by adjusting the color scale using an interactive range slider.

Visualizing Reorder Change

Slope graphs that connect the old position of a row in the source table to the new position in the destination table are used to encode reorder changes. The slope graphs exhibit similarities to the encoding of merge/split changes, but introducing more colors would increase visual clutter and make interpretation of color harder [14]. We visualize reorder changes separately in two boxes positioned on either the left or right side of the raw data heatmap (depending on whether it is the source or destination table) and below (see Figure 6(d)) to overcome the problem of similarity. This box-based layout makes it easier to spot reordering patterns and outliers. As with merge/split changes, the edge on the left represents the source table and that on the right the destination table. We integrate the option to highlight a particular reorder change by hovering over the connection line to make that change stand out. This encoding is applied to both rows and columns.

A toggle button above the diff heatmap allows the user to hide the diff heatmap in the center on demand, making it easier to follow the connection lines between the source and destination heatmaps.

Visualizing Merge or Split Change

We assume that a row merge operation is a combination of a removal of two or more rows from the first table and an addition of one new row at the same relative position in the second table.

To encode the merge/split change, we use a Y-connector (see Figure 6(c)). The visual encoding concept is assembled into a box, positioned to the right and below the diff heatmap. The left edge of the box represents the source table and the right edge the destination table. The merged rows from the source table are connected with a Y-connector that ends in the destination table. If rows are one below the other, the area between the Y-connector is filled. This encoding is applied to both rows and columns analogously and split changes are represented inversely. In the case of multiple merges, we use interaction (mouse over) to highlight the matching merge and split rows/columns.

Note that merge/split changes are rare and therefore not present in our two usage scenarios.

7 Implementation

In order to cope with large tables while providing browser-based access to users, we implemented TACO based on a client-server architecture, built using the Phovea Framework[7] [13] which is an open-source visual analysis platform that is developed as part of the Caleydo project[8]. The server component is implemented in Python and supports data management, manipulation, and change calculation. The client is implemented in TypeScript and uses D3.js[9] for rendering. Both components exchange data via a RESTful interface. On the server side, we use the Phovea server component to load and access the tabular datasets (mainly tables stored in CSV format or *Hierarchical Data Format (HDF)*[10] files), as illustrated in Figure 7. This gives us access to the actual data inside the tables and the corresponding row and column identifiers. Tabular data and identifiers are processed and compared on the TACO server. The results of the comparison are cached to avoid re-computation of the same datasets and sent as a JSON file to the client side using the *Flask*[11] *RESTful* interface.

On the client side, the TACO client extends the Phovea framework by adding a specialized TACO visualization component that visualizes the results of the table comparison from the server. To show the original tables as heatmaps, we use an optimized canvas version for heatmap visualizations from Phovea that minimizes memory and processing load on the client side. Further, for scalability reasons, the most detailed pairwise comparison view is not loaded and rendered automatically, but only on demand by using the 'Load Detail View' button in the interface. This avoids potentially longer waiting times if large tables need to be transferred from the server.

Note that in our current TACO prototype reorder changes are visualized to the left and right of the diff table. In further improvements we will adapt the layout to also encode reorder changes of the columns.

The prototype of TACO with preloaded data from both usage scenarios is available at `https://taco.caleydoapp.org`. The source code is hosted on Github[12].

8 Usage Scenarios

To demonstrate how TACO can be applied for interactive analysis of tabular data, we present two usage scenarios. For the first usage scenario, we chose the Summer Olympic Games dataset because it is commonly known, and many of the changes reflecting historical knowledge become immediately visible in TACO. The second usage scenario demonstrates how our technique is applied to cancer genomics data, for which we developed TACO in the first place.

Since none of the existing tools is able to encode the different change types in a scalable way, we decided against a formal comparative evaluation, which would just confirm obvious limitations of other approaches. However, we do believe that strong usage scenarios are sufficient to demonstrate the efficacy of our technique.

[7] `http://phovea.caleydo.org/`
[8] `http://caleydo.org/`
[9] `http://d3js.org/`
[10] `https://hdfgroup.org/`
[11] `http://flask.pocoo.org/`
[12] `https://github.com/Caleydo/taco/`

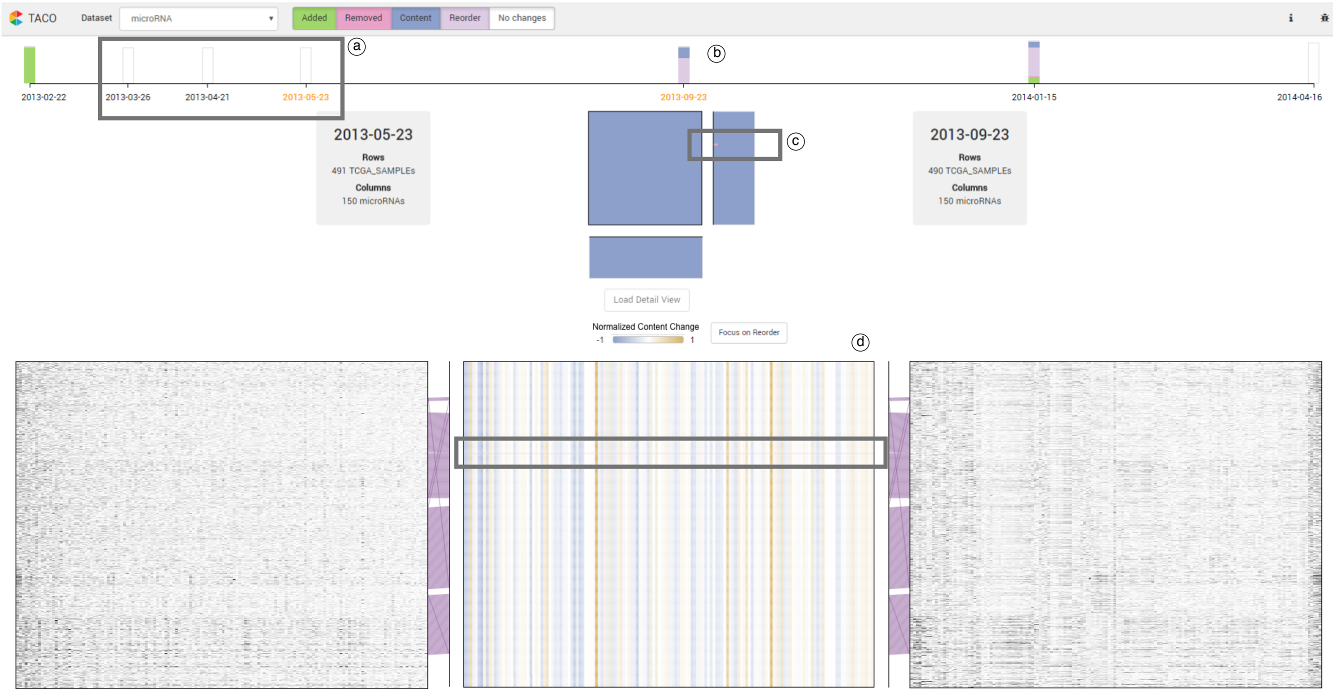

Fig. 8: *TACO showing the difference between the microRNA datasets from 2013-05-23 and 2013-09-23. The stacked bars along the timeline and the 2D ratio charts show that one patient was eliminated from the dataset because of quality issues (b), as indicated by the pink horizontal line in the 2D ratio chart (c) and in the diff heatmap (d). The mostly blue 2D ratio chart indicates that the values in all cells were changed because the removal of the patient data triggered a re-normalization applied to the whole dataset. Link to TACO state shown in this figure:* `http://vistories.org/taco-tcga-1`

8.1 Summer Olympics

In this usage scenario we demonstrate how TACO can help users to explore the four types of changes over the history of the Summer Olympic Games, as shown in Figure 1. The dataset published by the *IOC Olympic Studies Centre*[13] contains the medal tables (aggregated counts of gold, silver, and bronze medals) from all summer games between 1896 in Athens and 2012 in London. In 1896 the multi-sports event started with only 8 countries that competed in 7 disciplines. Over time, both the disciplines and the participating countries evolved to 45 countries and 89 disciplines, causing structural changes in the form of additions and removals. In addition to the changes in nations, the ranked medal table (ordered by the total number of medals per country) itself varied, which resulted in content and reorder changes.

The user starts by investigating the aggregated changes along the timeline (task T2). She recognizes that the stacked change bars increase in size over time, indicating that the table itself becomes larger in both directions due to a continually increasing number of countries and disciplines. Furthermore, gaps and deviations from the usual 4-year pattern caused by wars are also immediately visible.

To further investigate the changes caused by World War II, she selects the years 1936 and 1948, which results in the appearance of the 2D ratio chart below the timeline (task T3). According to the 2D ratio chart, only a few disciplines got added as columns, while the portion of added and removed rows representing the countries is much larger. To find out more, the user opens the detail view, which adds the source and destination tables together with the diff heatmap in the center, as shown in Figure 1 (task T4). By exploring the detailed changes, the user is able to confirm structural changes resulting from the outcome of the war. For instance, Germany (GER), which was ranked second in the medal table from 1936, did not participate in 1948 (but returned in 1952). Other nations, such as France (FRA), Belgium (BEL), and Italy (ITA), re-joined after skipping the games in 1936. In terms of disciplines, *200m Women* and *500m Singles Women* were added, while none were removed. The largest positive value change, indicated by the dark blue cell, is caused by the fact that Finland (FIN) was able to win all three *10,000m Men* medals in 1936, while they were not able to repeat this success in 1948. This is also reflected in the reorder changes, where Finland was ranked 6$^{\text{th}}$ in 1936 and last in 1948.

The changes discussed above are only example findings users could possibly make. It demonstrates how a set of domain tasks is addressed by the interactive visual exploration interface and what such a user session could look like. Next, we turn to a more complex scenario from the biomedical domain, which motivated the development of TACO.

8.2 Cancer Genomics

Large initiatives such as *The Cancer Genome Atlas (TCGA)* project collect comprehensive data from hundreds of patients for different tumor types. The goal of such initiatives is to generate and confirm hypotheses about tumor subtypes and their functional effects based on multiple molecular datasets, such as mRNA expression, microRNA expression, protein expression, copy number status, and gene mutation tables. The data is processed by large workflow systems that consist of dozens of analytic script-based tools performing tasks such as normalization, clustering, and significance tests. Running such a workflow results in a diverse set of derived data such as clustering results, lists of statistical scores, reports, and plots. Because of the size of the data and the complexity of the workflow, such pipelines can only be run in certain intervals, for example, once a month. However, changes in the workflow, its parameters, and the tabular input tables, make it difficult to determine what caused changes in the result, for instance, splitting of a cluster between runs. In our previous work on AVOCADO [27], we proposed a visual analysis approach that allows analysts to explore complex processing workflows that change over time. With TACO, we aim to give analysts a tool that also enables them to investigate changes in the tabular input data over time.

The presented usage scenario is based on cancer genomics datasets that are publicly available for download via the TCGA Data Portal[14]. We loaded different molecular datasets from cancer patients with *Glioblastoma Multiforme*, a very aggressive and common form of brain tumor, into the system. The primary goal of the analyst is to

[13] Olympic World Library `https://library.olympic.org/`

[14] `https://tcga-data.nci.nih.gov/`

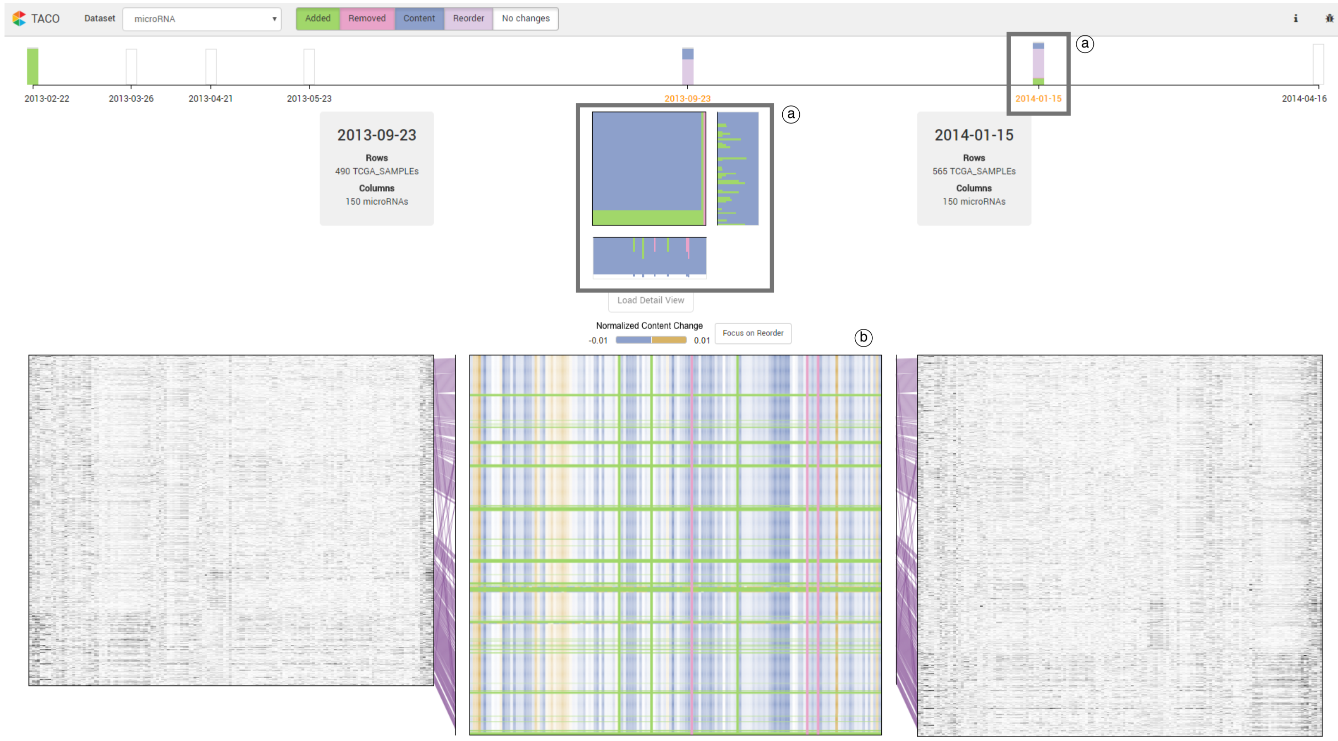

Fig. 9: *Selecting 2013-09-23 and 2014-01-15 as time points for the microRNA table comparison reveals that the re-normalization of the dataset again caused the content of the whole table to change. However, in contrast to the comparison shown in Figure 8, the content changes are subtle, and many more removals and additions become visible (a). Adjusting the color scale makes the content change more salient. While the additions of columns is caused by newly added patient data, the change in columns shows that some microRNAs got replaced by others because the new set of patients altered the ranking of the mostly significantly expressed microRNAs (b). Link to TACO state shown in this figure:* `http://vistories.org/taco-tcga-2`

find out if the tables change and if this is the case, how they changed. The usage scenario was created based on multiple feedback sessions with a bioinformatician who was strongly involved in the automatic processing of TCGA datasets.

The analyst starts by selecting the microRNA dataset, which contains 150 microRNAs (small non-coding RNA molecules that have been found to play a role in biological processes) for initially 491 patients. However, since increasing amounts of data were collected in the course of the TCGA project, the number of patients also increased over time. The 150 microRNAs included are those most significantly differentially expressed across all patients. The matrices are clustered using a *consensus non-negative matrix factorization* (cNMF) [10] for the patients and a *tree clustering* [9] for the microRNAs.

Following the multi-level detail concept of TACO, the analyst starts by investigating the aggregated changes encoded as stacked bar charts along the timeline (see Figure 8, task T1). A white stacked bar (a) shown for the first three runs indicates that the dataset stayed the same in terms of content and structure (task T2). However, the analyst then notices that between the runs from *2013-05-23* and *2013-09-23* all values changed slightly, as indicated by a full blue and purple bar (b). When inspecting the 2D ratio chart with the attached histograms, the analyst realizes that, in addition to the content changes, a single sample was excluded from the dataset, which is indicated by a pink horizontal line (c). By looking into the data acquisition protocols, he is able to confirm that the data from this patient was indeed retracted because of quality issues with the tissue sample. By taking a closer look at the preprocessing scripts that are applied to the table, he also finds out that with every addition or removal, the whole dataset is re-normalized, which explains the content changes. This also explains the stringy patterns where the same number change is applied uniformly to all samples in a microRNA column and the rows that are shifted due to a slightly different clustering result (steep lines in reorder slope graphs).

The analyst continues with an inspection of the differences between runs *2013-09-23* and *2014-01-15*, as shown in Figure 9 (task T3). While the overview again shows a large portion of content and reorder changes, more additions and removals of rows and columns also become visible (a). The analyst expected a large number of additions caused by newly included patients, which he is able to visually confirm. However, the structural changes in the microRNA columns (b) are surprising, which makes the analyst dig deeper by opening the most detailed difference heatmap visualization (task T4). After consulting with colleagues, he concludes that the changed set of microRNAs is caused by the new set of patients, which in turn altered the ranking of the most significantly expressed microRNAs. These ranking changes are shown as line patterns in the reorder slope graphs.

During the development of TACO, we continuously collected feedback by discussing different versions of the prototype with our collaborator. In addition to providing valuable suggestions for improvement that we integrated in multiple iterations, he repeatedly mentioned the added value of TACO for quality control purposes. Without a visual tool like TACO, analysts must manually apply traditional diff tools with two time points as input and look at the results. Doing this for many input tables that change in the course of large-scale projects is tedious. Further, the evolution of datasets can be investigated only indirectly by running the diff tool multiple times with two table versions at different time steps as input and then manually comparing the results. Finally, our collaborators mentioned that they intend to use TACO to show the evolution of long-running cancer genomics projects, such as TCGA.

9 Discussion and Limitations

Although we aimed to cover a broad range of user tasks and to implement a fully functional prototype, some limitations are inherent to the current concept and pose challenges for further improvement.

Heterogeneous Tables The presented concept of encoding the differences in a *diff table* with representations of higher level aggregates can be used to compare not only homogeneous but also heterogeneous tables. The current version of TACO compares only homogeneous tables (matrices), that is, tables that contain the same data type, range, and semantics in all columns and rows. However, in a heterogeneous table, every column (or row) can have a different data type and should therefore be visualized differently. We believe that TACO is able to cover this conceptually, but this will require a special normalization approach and visual encoding to effectively reflect the heterogeneity of the input tables.

Single Item Level Drill-Down At the most detailed level (Figure 3), the raw data and difference heatmap provide insights into changes between two particular table versions, for instance, where the change occurred (cell, rows, and columns) and which type of change was detected (added, removed, reordered, or content changes of cells, rows, and columns). A drill-down to the item level is partly integrated into the interface of the prototype. For smaller datasets, we present labels attached to the heatmaps that describe rows and columns (see Figure 1). However, to also support single item drill-down operations for larger tables, such as those demonstrated in the biomedical usage scenario, a possible solution would be to introduce a focus+context approach that increases the number of selected rows and columns while preserving the context.

Heatmap Limitations The raw data table heatmaps we present at the highest detail level in TACO have some limitations relating to aspect ratio and information density. The larger the table, the less detailed information the user can obtain form the visualized heatmap, except in cases in which the order of rows and columns provides a meaningful structure and users can see patterns in the data. A meaningful order of rows and columns can be achieved either by clustering the table or by sorting the rows/columns by another attribute, such as time or some hierarchy (e.g., product classification, ICD-10 codes). If this is not the case, the heatmap representation could be irrelevant to the user and therefore hidden from the interface.

Furthermore, we identified some restrictions of the concept when tables with a large number of rows and only few columns are visualized, or vice versa. In these cases we can either fill the available space by distorting the table or add scrollbars. This limitation could be resolved by integrating a lens to zoom into a particular area of the visualization.

Besides unbalanced aspect ratios, the number of available pixels on the screen is an inherent scalability limit of pixel-based representations. Going beyond this limit either requires scrolling, with unfavorable effects for gaining an overview, or makes aggregation of rows and/or columns necessary. This could lead to situations where different change types would need to be merged to a single pixel in the diff heatmap. An alternative approach would be color weaving [2], where pixels within a defined area are permuted randomly. However, this will require further research and evaluation.

Merge and Split Changes We have developed a visual encoding concept for merge changes as described in Section 6. Due to the lack of merge or split changes in the Cancer Genomics and the Olympic Medals datasets, we cannot demonstrate it in this version of the prototype.

Scalability and Performance TACO scales from a few dozen cells in the *Summer Olympic Games* dataset (8 rows $\times$ 7 columns for the total medals matrix of 1,896) to millions of cells in the *TCGA* dataset (571 rows $\times$ 24,174 columns of the copy number matrix from 2014-04-16). We improve the performance of the change computation by requiring unique identifiers for rows and columns and caching of the computation results.

User Tasks In Section 3, we introduced a number of user tasks for our problem domain. In prior work on task taxonomies, comparison tasks were in most cases included as leaf nodes in the corresponding task spaces [4, 22]. Hence, details about the nature of the comparisons have not yet been discussed at a more systematic level. Andrienko & Andrienko [4] differentiated between elementary and synpotic, direct and indirect, and comparison and relation seeking tasks depending on whether single or sets of elements, referrers or attributes, and values or the type of relationship are compared to each other. However, aspects such as structure vs. content are not expressed. Moreover, Gleicher et al. [12] mentioned that a complete taxonomy of comparison tasks remains outstanding. As a first step towards filling this gap, we propose the following three dimensions that comprise a more general task space for comparing complex data elements over time: (1) number of elements to compare: two vs. many; (2) high-level type of change: structure vs. content vs. reorder vs. merge/split vs. meta-data; (3) time: linear vs. branching. The first dimension refers to how many elements are compared, that is, whether pairwise comparisons are performed or more than two tables are compared at once. The second dimension focuses on the possible types of changes, inspired by research into versioning of relational databases [23]. Finally, the third dimension deals with the temporal model applied [1], that is, whether changes appear in a linear succession of events over time or if branches can also be modeled to allow for parallel branches of derived tables. As can be seen from the task space described, TACO supports a majority—but not all—of the design aspects. First, it focuses on pairwise table comparisons and does not support multi-way comparison of more than two table instances at once. Second, in terms of change types, meta-data changes are not covered. Third, branching histories of table changes are not supported in the timeline overview. All three of these would fit into the overall concept of TACO, but require further consideration in terms of change calculation, visual representation, and interaction. These need to be addressed as part of future work.

Detect Semantics in Changes Certain data transformations affect a table globally in a uniform manner, but do not change it completely. A concrete example of such a transformation is the normalization of data values. In its current form, TACO detects changes based on value comparisons at the cell level, which might lead to the false impression that the table has changed completely. To mitigate this, global data transformations or specific change semantics, such as normalizations, could be detected automatically and represented differently.

10 CONCLUSION

Investigating changes between multiple versions of a table, for example, to detect modifications in monthly payroll tables or to observe differences in multiple biological experiments, is a challenging task. In this work, we presented a visual comparison tool based on calculating pairwise differences between tabular datasets along a set of change types. We have introduced an interactive visualization concept that allows their differences over time to be investigated across multiple levels of detail. TACO provides two of three possible types of visual comparison approaches: juxtaposition and explicit representation [12]. To validate the effectiveness of our visualization tool, we applied it to two usage scenarios.

A number of future challenges have already been discussed in the previous section. In addition to addressing these limitations, planned future work includes conducting a long-term case study to collect empirical evidence on the application of TACO as well as benefits and potential difficulties to be addressed. Furthermore, as mentioned in the biomedical usage scenario, we plan to combine TACO with our previous work on AVOCADO [27]. In combination, the two solutions will provide domain experts with an effective tool that helps to understand which changes in the input tables, the processing workflow, and its parametrization caused certain changes in the output of such complex pipelines. This kind of provenance and causality analysis is also essential for making the results of data-driven sciences more reproducible and, in the long term, also more sustainable.

ACKNOWLEDGMENTS

We want to thank Samuel Gratzl for his contributions made during all phases of the project as well as to the implementation of the prototype system. We would also like to thank Nils Gehlenborg for giving domain-specific input on the biomedical usage scenario and for providing feedback on the prototype implementation and its potential value to domain experts. This work was funded by the Austrian Science Fund as part of the *VisOnFire* project (FWF P27975-NBL), the State of Upper Austria (FFG 851460), and the Austrian Ministry for Transport, Innovation and Technology (BMVIT) under the ICT of the future program via the VALiD project (FFG 845598) .

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human–Computer Interaction Series. Springer Verlag, London, UK, 2011.

[2] D. Albers, M. Correll, and M. Gleicher. Task-Driven Evaluation of Aggregation in Time Series Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*, pages 551–560. ACM Press, 2014.

[3] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted Graph Comparison Techniques for Brain Connectivity Analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pages 483–492. ACM, 2013.

[4] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data*. Springer, Berlin, Germany, 2006.

[5] D. Archambault. Structural Differences Between Two Graphs Through Hierarchies. In *Proceedings of the IEEE Conference on Graphics Interface (GI '09)*, pages 87–94. Canadian Information Processing Society, 2009.

[6] L. Bartram, A. Patra, and M. Stone. Affective Color in Visualization. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '17)*, pages 1364–1374. ACM, 2017.

[7] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The State of the Art in Visualizing Dynamic Graphs. In *Proceedings of the Eurographics Conference on Visualization (EuroVis '14) – State of The Art Reports*, 2014.

[8] M. Behrisch, J. Davey, F. Fischer, O. Thonnard, T. Schreck, D. Keim, and J. Kohlhammer. Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom. In *Computer Graphics Forum*, volume 33, pages 411–420. Wiley Online Library, 2014.

[9] Broad Institute of MIT and Harvard. Broad Institute TCGA Genome Data Analysis Center: Clustering of miR expression: consensus hierarchical, 2016. doi:10.7908/C1M32V4K.

[10] Broad Institute of MIT and Harvard. Broad Institute TCGA Genome Data Analysis Center: Clustering of miR expression: consensus NMF, 2016. doi:10.7908/C1M32V4K.

[11] B. Craft and P. Cairns. Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra? In *Proceedings of the International Conference on Information Visualization (IV '05)*, pages 110–118. IEEE Computer Society, 2005.

[12] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual Comparison for Information Visualization. *Information Visualization*, 10(4):289 –309, 2011.

[13] S. Gratzl, N. Gehlenborg, A. Lex, H. Strobelt, C. Partl, and M. Streit. Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*. IEEE, 2015.

[14] S. Haroz and D. Whitney. How Capacity Limits of Attention Influence Information Visualization Effectiveness. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2402–2410, 2012.

[15] J. Heinrich, C. Vehlow, F. Battke, G. Jäger, D. Weiskopf, and K. Nieselt. iHat: Interactive Hierarchical Aggregation Table for Genetic Association Data. *BMC Bioinformatics*, 13(8):1, 2012.

[16] R. Hourieh, H. Stitz, N. Gehlenborg, and M. Streit. TaCo: Comparative Visualization of Large Tabular Data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '16)*, 2016.

[17] D. A. Keim and H.-P. Kriegel. VisDB: Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, 1994.

[18] 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.

[19] 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.

[20] 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.

[21] I. Liiv. Seriation and Matrix Reordering Methods: An Historical Overview. *Statistical Analysis and Data Mining*, 3(2):70–91, 2010.

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

[23] J. F. Roddick, N. G. Craske, and T. J. Richards. A taxonomy for schema versioning based on the relational and Entity Relationship Models. In *Entity-Relationship Approach — ER '93*, pages 137–148. Springer, Berlin, Heidelberg, Dec. 1993.

[24] J. Schmidt, M. E. Gröller, and S. Bruckner. VAICo: Visual Analysis for Image Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, 2013.

[25] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages (VL '96)*, pages 336–343, 1996.

[26] H. Song, B. Lee, B. Kim, and J. Seo. DiffMatrix: Matrix-based Interactive Visualization for Comparing Temporal Trends. *Proceedings of the Eurographics Conference on Visualization (EuroVis '12) – Short Papers Track*, 2012.

[27] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg. AVOCADO: Visualization of Workflow–Derived Data Provenance for Reproducible Biomedical Research. *Computer Graphics Forum*, 35(3):481–490, 2016.

[28] C. Tominski, C. Forsell, and J. Johansson. Interaction Support for Visual Comparison Inspired by Natural Behavior. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2719 –2728, 2012.

[29] E. Tufte. *Envisioning information*. Graphics Press, 5th edition, 1995.

[30] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual Comparison of Event Sequence Data. In *Proceedings of the Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, pages 259–268. ACM Press, 2015.