Comparative Analysis of Multidimensional, Quantitative Data

Alexander Lex, *Student Member, IEEE*, Marc Streit, *Student Member, IEEE*,
Christian Partl, Karl Kashofer, and Dieter Schmalstieg, *Member, IEEE*
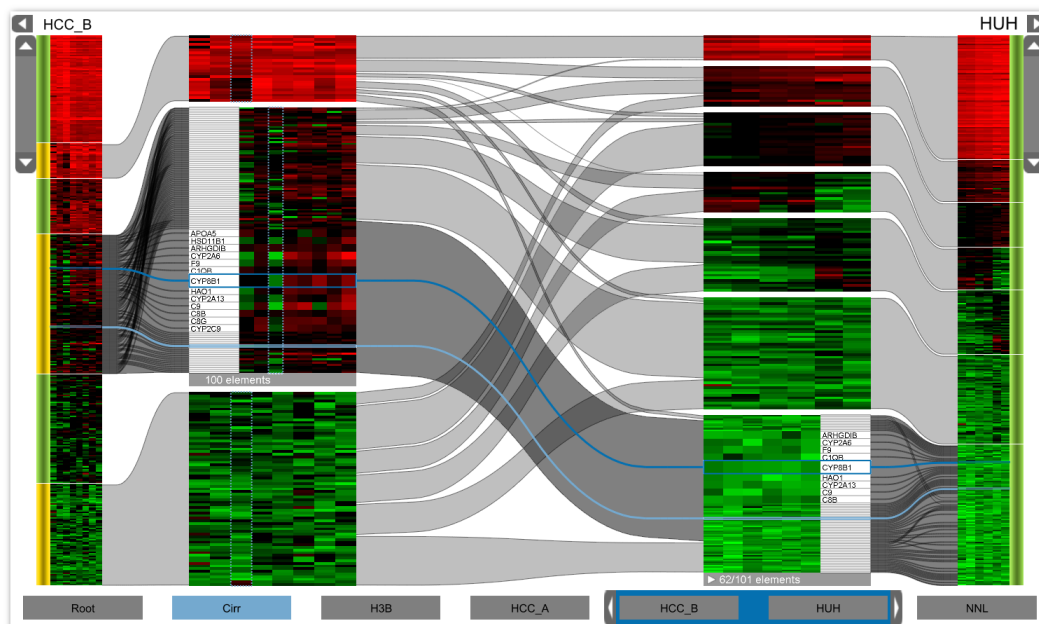
Fig. 1. The Caleydo Matchmaker technique allows users to split multidimensional datasets into subgroups, cluster them separately and analyze relations between the resulting clusters.

Abstract—When analyzing multidimensional, quantitative data, the comparison of two or more groups of dimensions is a common task. Typical sources of such data are experiments in biology, physics or engineering, which are conducted in different configurations and use replicates to ensure statistically significant results. One common way to analyze this data is to filter it using statistical methods and then run clustering algorithms to group similar values. The clustering results can be visualized using heat maps, which show differences between groups as changes in color. However, in cases where groups of dimensions have an a priori meaning, it is not desirable to cluster all dimensions combined, since a clustering algorithm can fragment continuous blocks of records. Furthermore, identifying relevant elements in heat maps becomes more difficult as the number of dimensions increases. To aid in such situations, we have developed Matchmaker, a visualization technique that allows researchers to arbitrarily arrange and compare multiple groups of dimensions at the same time. We create separate groups of dimensions which can be clustered individually, and place them in an arrangement of heat maps reminiscent of parallel coordinates. To identify relations, we render bundled curves and ribbons between related records in different groups. We then allow interactive drill-downs using enlarged detail views of the data, which enable in-depth comparisons of clusters between groups. To reduce visual clutter, we minimize crossings between the views. This paper concludes with two case studies. The first demonstrates the value of our technique for the comparison of clustering algorithms. In the second, biologists use our system to investigate why certain strains of mice develop liver disease while others remain healthy, informally showing the efficacy of our system when analyzing multidimensional data containing distinct groups of dimensions.

Index Terms—Multidimensional data, cluster comparison, bioinformatics visualization.

---

1 INTRODUCTION

While a lot of research has been conducted on multidimensional data analysis, most approaches try to either visualize the data as a whole

---

- *Alexander Lex, Marc Streit, Christian Partl and Dieter Schmalstieg are with the Institute for Computer Graphics and Vision at Graz University of Technology, E-mail: {lex|streit|partl|schmalstieg}@icg.tugraz.at.*
- *Karl Kashofer is with the Medical University of Graz, E-mail: karl.kashofer@medunigraz.at.*

using techniques such as parallel coordinates, or to extract the most relevant aspects using dimensionality reduction. In many cases, however, multidimensional datasets have an additional property which is often not employed to gain insights. Metadata, either explicitly encoded or just informally known by the user, allows us to split data into smaller batches, analyze and process it separately and then compare the batches with each other.

The Caleydo visualization framework [19] developed at our institute addresses the field of biomolecular data visualization. When discussing analysis methods with our collaborators, we discovered that most of their data has inherent groupings. For example, they want to compare measurements from different genotypes of a species, or from patients suffering from diverse forms of cancer. In biomolecular data

analysis, clustering is used to group records into meaningful subsets. However, clustering, especially of many dimensions, can conceal important relations. Figure 2(a) illustrates one such case. The samples 1 and 2 in this parallel coordinates plot will probably not end up in the same cluster when all dimensions are clustered at the same time. In many cases this is desirable. However, if we know that the first three axes are from experimental conditions different from the last two, the information that these records behave differently under these conditions may exactly be what we want to know. Our work aims to make this difference explicit to the user, by clustering those groups of dimensions (*i.e.*, subspaces of the dimensions) separately and showing the differences and similarities between them.
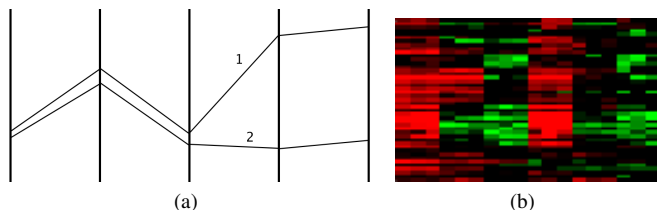

(a)   (b)

Fig. 2. (a) An example of data records which would be assigned to different clusters depending whether the dimensions were clustered combined or separately. (b) Scrambled, inhomogeneous cluster where no clear biological function can be assigned.

For a biologist, the optimal clustering lets him designate a clear biological meaning to a cluster. This is impossible for a cluster as shown in Figure 2(b), since the cluster is too inhomogeneous. The biologist's goal is to find all records that, for example, increase over time in one group, and then explore how these behave in another. This is not possible when all groups are clustered at the same time.

A related problem is the need to compare the results of clustering algorithms. Different algorithms, parameters and similarity measures can have a profound impact on the result [25, 26]. Quality metrics for clustering algorithms are hard to find. Usually, the quality is assessed manually through interpretation by the user. A method that clearly visualizes the differences between two algorithms could support the process of judging the quality of a clustering result significantly.

To solve these problems, we developed Matchmaker. Our primary contribution is a comprehensive Focus+Context technique employing details-on-demand and drill-down capabilities for comparing multiple, separately clustered groups of dimensions. As a secondary contribution, we introduce an order preserving curve bundling strategy which minimizes crossings between clusters.

Clustering is actually only one method to group records. Alternatively, other strategies could be applied, for example, grouping of genes based on their source chromosome. For the sake of clarity, we will refer to groups of records as clusters, independent of how they are grouped in order to distinguish between groups of dimensions and groups of records.

The paper is structured as follows: in the next Section we discuss related work. The system including the Matchmaker technique is introduced in Section 3. Section 4 contains two case studies, the first for clustering algorithm comparison, the second describing how a biologist uses our system for biomolecular data analysis. Section 5 concludes the paper.

2 Related Work

Comparing data is typically achieved using coordinated and multiple views [23] where at least one view is shown for each data set (*e.g.*, [29]). Exceptions are for example Graham and Kennedy [6] who use a directed acyclic graph to explore multiple trees in the same view, or Hong *et al.* [12] who use a union of two trees and highlight differences (similar in effect to image differencing used in image processing). Tu and Shen [33] visualize changes of hierarchical data using tree maps. Their approach also uses a union tree which is visualized with contrast tree maps. Changes of attributes are shown by blending the attribute

colors in one square from the attribute value in the first tree to the value in the second tree. For multiple attributes they use bars in the tree map squares to show the ratio between the two datasets. Changes in structure are visualized with special colors.

For data with many differences, however, such representations are ill suited, since, for example, integrating two trees into one view is difficult to understand. Consequently, we chose to employ an approach utilizing coordinated and multiple views.

However, the human visual system is not very good at comparing elements in different views. Therefore, it is advantageous that relations are shown explicitly to a user. Three basic ways to show relations between elements exist: color coding, drawing lines and blinking [25]. While blinking is the strongest attractor, it is also considered disturbing by many users and can hardly be used for more than one or two items. Color, as the second possibility to show relations, has been widely used in the literature. Van Long and Linsen [20] use colored brushing to show relations between a cluster tree and the concrete values in a parallel coordinates browser. Graham and Kennedy [5] show multiple trees and visualize their relations by interactive linking and brushing. They use orthogonal stretching [24] to focus on an area of interest. Munzner *et al.* [22] developed TreeJuxtaposer, a system that supports comparing large trees. They introduce the concept of guaranteed visibility, where elements identified as relevant are guaranteed to be visible, as well as structural comparison – *i.e.*, to find for all nodes of a tree their most similar nodes in every other tree. This allows them to use structural brushing, which highlights corresponding areas in all trees. TreeJuxtaposer also employs orthogonal stretching for areas of interest.

Using color alone, however, has several drawbacks. Healy [9] found out in a study that more than seven colors lead to reduced performance in accurately and rapidly detecting the colors. As a consequence, using color to encode differences does not scale well. In applications such as Seo and Shneiderman's Hierarchical Cluster Explorer (HCE) [25] or our Matchmaker, color is already heavily employed to show other data attributes. Consequently, we have chosen curves and ribbons to show relations. Lines and curves have the unique ability to scale to many elements, especially if measures such as bundling to avoid visual clutter have been taken.

HCE [25] is probably the work most closely related to ours. It supports comparing the effects of two different clustering algorithms on the same dataset. It renders two heat maps side by side and draws straight connection lines between the related items. While Seo and Shneiderman state that this basic implementation was already very helpful for their users, they also identify the problem that simply criss-crossing lines can cause confusion for the users. Furthermore, they show their method only for very small datasets (<50 records and 6 dimensions).

Holten [10] proposes a hierarchical edge bundling method which adds adjacency information of hierarchically structured data on top of an existing visualization. Holten and van Wijk [11] extended the concept of hierarchical edge bundling for the comparison of two hierarchies, which is similar to our problem. Their method works well if the leaves in one hierarchy can be resorted to minimize crossings of inter-tree edges. However, in our case the order of the records cannot be changed since the order encodes information.

Meyer *et al.*'s Mizbee [21] and Krzywinski *et al.*'s Circos [17] use bundled curves respectively ribbons to show relations and differences between genomes arranged in a circular layout. Mizbee uses two circles of chromosomes, one for each species' genome. The selected chromosome of the outer circle is copied to the inner ring, and curves are drawn between the location of conserved regions in this one chromosome and all other chromosomes in the target species. Consequently, only relations of one source chromosome to the target's chromosome are shown at a time. Additionally, an enlarged rectangular detail of the source chromosome, which uses color coding to convey the same information, and a view of one block's details compared to the block in the target species, is provided. In contrast to MizBee, Matchmaker can compare many datasets at the same time in an overview, while MizBee can compare only one chromosome of a source to a tar-

get. In addition, Matchmaker integrates detail views, which are shown separately from the circular layout in MizBee, into the overview and thereby makes the relation between overview and detail more obvious.

Circos [17] can place several datasets in concentric rings and show position changes with curves connecting the rings. However, this method does not scale to many changes in position, which is why alternatively chromosomes from different samples can be arranged on a single circle. Using a multi-circular layout for analysis of clusters in multidimensional datasets would not be possible due to the many changes in sequence. Furthermore, analyzing inter-relations between more than one dataset (or in our case groups of dimensions) with a circular layout would result in heavy over-plotting.

Telea and Auber [32] describe Code Flow, a system for comparing different versions of source code on the code level. They do this by rendering icicle plots along vertical axes, where each axis represents the version of the software system under investigation. They then draw spline tubes between corresponding fragments in the different versions. The tubes are opaque in the middle and translucent at the borders, to allow a clear separation of the tubes. To draw the user's attention to changes, they use color for the bands that changed within the range of currently inspected versions, while others are rendered gray. Telea and Auber's application domain is vastly different from ours: source code evolves gradually and thereby makes bundling or similar measures obsolete. In addition, Code Flow does not provide drill-down methods which preserve the context to the whole dataset.

Parallel sets by Kosara *et al.* [16] uses axes with boxes of categories where the size of the box is proportional to the frequency of elements in the category. Relations between the categories are shown by connecting them with parallelograms. One could consider the clusters in our datasets as categories and use parallel sets to visualize the relations. However, parallel sets do not show individual elements and consequently do not use any details-on-demand techniques.

A visualization of clustering stability between several algorithms was developed by Sharko *et al.* [26]. They use a cluster stability matrix, which shows the number of times two genes appear in the same cluster when running different algorithms. To visualize the stability matrices they use heat maps. While they employ an indirect approach of calculating and visualizing a metric, we directly show relations between clustering results. They thus only provide information on how different results are, but not on the exact differences.

Automatic dimension reduction, using, for example, principal component analysis (PCA) [14], is not desirable in our case, because we assume that the input data is part of a well designed experiment where the user has a priori knowledge of the dimensions' semantics and may already have hypotheses about their relations.

3 OVERVIEW OF MATCHMAKER

The comparative analysis workflow is basically a two-tier approach. First, the user needs to group the dimensions and cluster their records separately. This is described formally in Section 3.1 and from a user's perspective in 3.2. Second, the interactive analysis is conducted using Matchmaker. In Section 3.3, we introduce the technique including a discussion of the applied curve bundling strategy for the inter-group connections and the overview and detail approach. We conclude this section with a reflection of the technique's scalability (3.4) and some words about its implementation in the Caleydo framework (3.5).

3.1 Formalizing the problem

Formally, the tabular input data is a matrix $M = \{v_{ij} | 1 \leq i \leq n, 1 \leq j \leq m\}$ where the columns $D = \{d_1, ..., d_m\}$ are the dimensions and the rows $R = \{r_1, ..., r_n\}$ represent the data records (see Figure 3(a)). Each matrix cell v_{ij} is a value in row r_i of the dimension d_j. We introduce a user-driven grouping of dimensions $G = \{g_1, ..., g_n | g \in \mathscr{P}(D)\}$ where each dimension can be assigned to multiple groups (in Figure 3(b) for example, d_2 is assigned to g_1 and g_2). For each g_i in G we create a set $C_i = \{c_1, ..., c_n\}$ which contains the rows (restricted to dimensions in g_i) automatically determined by a clustering function (Figure 3(c)), where r_j can only be part of one cluster. The comparison is then performed on the grouped and clustered data. The records
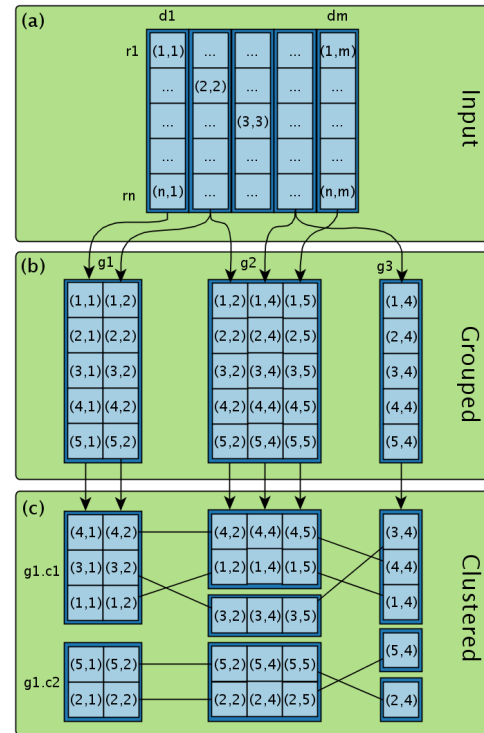

Fig. 3. Abstract comparative analysis workflow. The dimensions of the tabular input data (a) are arbitrarily grouped together (g_1, g_2, g_3 in (b)). The groups are then clustered (*e.g.*, resulting in c_1 and c_2 for group g_1) in (c)) and finally are ready for the analysis.

$r_1, ..., r_n$ are connected among all groups, enabling the user to detect whether records remain in one cluster or if clustered records are pulled apart.

3.2 Data preparations

The grouper view (*cf.* Figure 4) supports the creation of a hierarchically grouped dimension set in a nested representation. After loading the input data from a database or CSV file, the grouper presents the dimensions as a nested tree, where the dimensions are children of the root. The dimensions can be combined to groups, duplicated, removed and resorted.

After concluding the group definition, a common first task is the reduction of data records based on statistical tests. A reduction to data records that show statistically interesting behavior is often desired. In
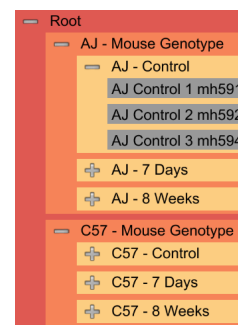

Fig. 4. The grouper view supports creating a hierarchical sorting of data dimensions in a nested tree representation. The user can trigger statistics on individual groups as well as between groups. Finally, the user selects a set of groups from arbitrary depths in the hierarchy for the comparative analysis.
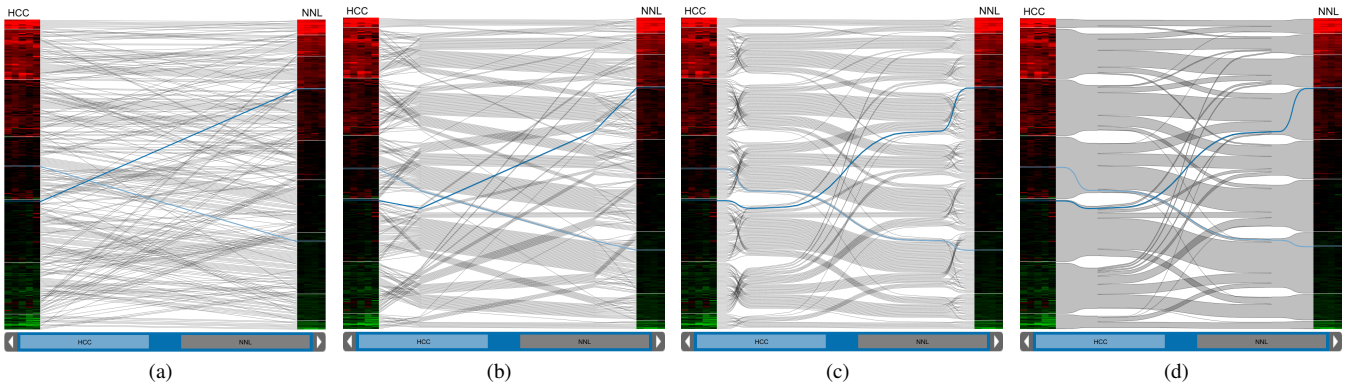
Fig. 5. The comparison of two datasets with different curve styles for connecting the records. The straight line rendering in (a), where we connect the records directly, produces a cluttered image, even for this relatively small dataset of about 400 records. In (b) we still use straight lines, but apply the resorting strategy with added control points on a per-cluster basis (see Figure 6), resulting in a much clearer representation with identifiable cluster relations. In (c) we replaced the lines with spline curves. The curves are abstracted to ribbons in (d).

molecular biology tests are often designed with a goal or specific question in mind, for example to find genes which change expression over time in one condition while remaining constant in another one. Also, statistics can be used to remove records which are not consistent over a series of replicates and therefore do not meet the quality criteria of a study. Ultimately, data reduction is necessary to bring datasets of larger scale to a size suitable for exploration with the Matchmaker technique.

For this purpose, we integrated the R software environment [31]. By using R, we can utilize a rich set of statistical methods, producing validated results with the best possible performance. Currently, we use t-testing and reduction based on fold-change. However, other functionality can be easily added by an end-user savvy in R. Filtering based on statistics is applied on groups defined in the grouper view. After this data preprocessing, we have a subset of the raw data which is then used as input for the comparative analysis.

When the user triggers loading of the selected groups in the comparison view, a clustering algorithm is run on each of the groups separately. The clustering results in a classification of the data records according to an attribute. The Caleydo framework provides partitional (e.g., k-means, affinity propagation [4]) and hierarchical clustering algorithms (e.g., Eisen et al.'s tree clustering algorithm [3]) as well as interfaces to Weka [7] and R [31] to utilize external cluster implementations. The nature of hierarchical clustering algorithms requires a cut-off along the dendrogram to determine the actual clusters. Caleydo uses a default value for the cut-off, which can be modified using a slider on a dendrogram in the Matchmaker detail view (see Section 3.3).

3.3 Visualization Technique

The Caleydo Matchmaker technique allows a visual comparison of multiple groups of clustered data. Since there is no inherent order of clusters and records in the clusters, we sort both clusters and records within the clusters according to their mean value. Consequently, we introduce meaning to the position of the records – which is important since position is the most powerful visual variable available [1]. In addition, having introduced a specific ordering, we can use the parallel coordinates metaphor [13]. We arrange the groups side by side, where each group is equivalent to an axis in a parallel coordinates plot. By connecting the identical records between groups, we complete the parallel coordinates metaphor. However, instead of using simple lines as axes, we show the groups as heat maps. This allows us to encode:

- the magnitude of the concrete values in the heat map using color,

- the average magnitude of a cluster via position,

- the average magnitude of a record relative to others in the same cluster via position in the cluster and

- the relations between clusters and records via connection curves.

Since we aim to visualize amounts of data on a scale where a single pixel has to represent more than one value, we face the problem of level of detail (LOD) culling. Fortunately, the clustering automatically aggregates data, so that even if LOD culling occurs, the global trends are still visible. However, our requirements make it necessary to be able to explore the magnitude and the relations of individual data records. Consequently, following Shneiderman's mantra – overview first, zoom and filter, details-on-demand [27] – Matchmaker provides an overview and interactive embedded detail views for individual clusters (cf. Figure 1). In both overview and detail mode relations are shown using curves or ribbons. A naive approach for connecting records, however, results in visual clutter rendering the visualization unusable. Therefore, we have developed an edge bundling strategy suitable for our task.

Edge Bundling

The most primitive way to show the cluster distribution among the groups is to draw straight lines connecting the data records. This basic approach is for instance employed by Seo and Shneiderman [25] as an add-on feature of their HCE software. As discussed earlier, this method does not scale well. Even in small datasets it is hard to identify trends. Figure 5(a) shows the comparison of two heat maps with about 400 data records. While at the top the data records remain mostly
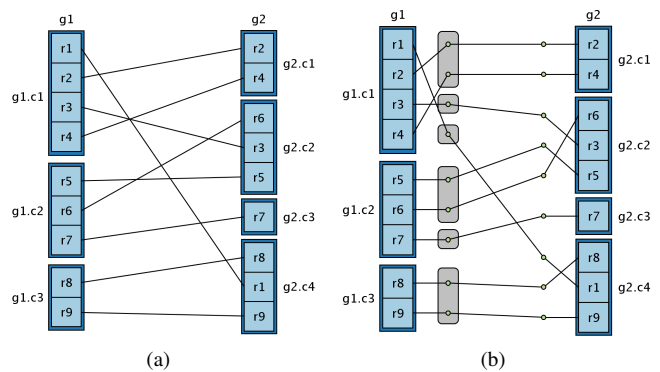

Fig. 6. (a) The naive approach using direct connections. (b) Our bundling strategy, where we introduce support points (green) through which the curves are routed. Support points are sorted based on the destination cluster of their connection. This technique minimizes crossings between the clusters at the cost of crossings between the support points and their cluster.

within the same cluster, everywhere else crossings between clusters can be observed. Using brushes to highlight records improves the situation slightly, but relies purely on interaction.

One could argue that the straight lines work reasonably well in parallel coordinates plots, especially if some basic clutter reduction methods such as using transparency are employed, which should work equally well in our analogous system. However, typically parallel coordinates plots are not used to show data that is evenly distributed along an axis (except for cases where they are used to show categorical data) in the way Matchmaker does.

One possibility to address this problem would be to sort the records within clusters, since, as stated before, the order within a particular cluster has no a priori meaning. Resorting the data records in the clusters by taking their position in the compared group into account can reduce the number of crossings significantly. However, since we want to employ position as a visual variable, and want to compare more than two groups simultaneously, this is not an option.

As a consequence, using methods that rely on sorting for crossing reduction, as for example Holton's method of hierarchical edge bundles [11] does, is not possible, even when a hierarchy behind the data is available (*e.g.*, when a hierarchical clustering algorithm was used to produce the clusters). We therefore developed a bundling strategy which:

- makes use of the grouping of records into clusters,

- makes use of the knowledge about the destination position of a record and

- minimizes crossings of bundles between clusters by accepting crossings of individual lines within clusters.

The basic idea is sketched in Figure 6(b). For every record in every group we introduce a support point (green in Figure 6(b)). Records within a cluster are connected to any of the support points within the cluster, but never to a support point from another cluster. The support points are ordered, so that the topmost support point is associated with the topmost cluster in the target group for which the source cluster in the source group has a record. Once this point has been designated, the next highest point of the source cluster is associated with the next-highest equivalent record. If there is another equivalence between the clusters, the target cluster's next point is used (as for the connection of r_4 in $g_1.c_1$ to its equivalent in $g_2.c_1$ in Figure 6(b)). Otherwise the next cluster is searched for equivalences. If there is one, the points are associated (as for example the connection of r_3 in $g_1.c_1$ to $g_2.c_2$). This is repeated for all clusters of the source group. The support points in the source cluster then are iteratively connected with the topmost free record in the source cluster that is connected to the topmost target cluster. This is done for the target clusters as well.

As a result, all records from a target cluster which connect to the same source cluster are assigned to control points that are adjacent in both the source and the target cluster. Therefore, all connections between two clusters from the source and the target group now run in parallel, minimizing the crossings between control points. This technique enables a user to easily identify trends as well as outliers. The main trends are shown as wide bands, connecting the clusters at low angles, if the compared data is somehow similar. Outliers are easily perceived as thinner bands at steep angles that cross several groups. As a trade-off, there are now many crossings between the clusters and their control points, making the precise association between records of two groups difficult for non-trivial cases. However, since this is not possible with any other bundling strategy either, and can be alleviated using interactive brushing, we believe the bundling strategy is a significant improvement.

Figure 5 (a)-(d) illustrates the different approaches for showing connections. Figure 5(a) uses straight lines and no bundling, in Figure 5(b) the bundling strategy was applied, again using straight lines. This
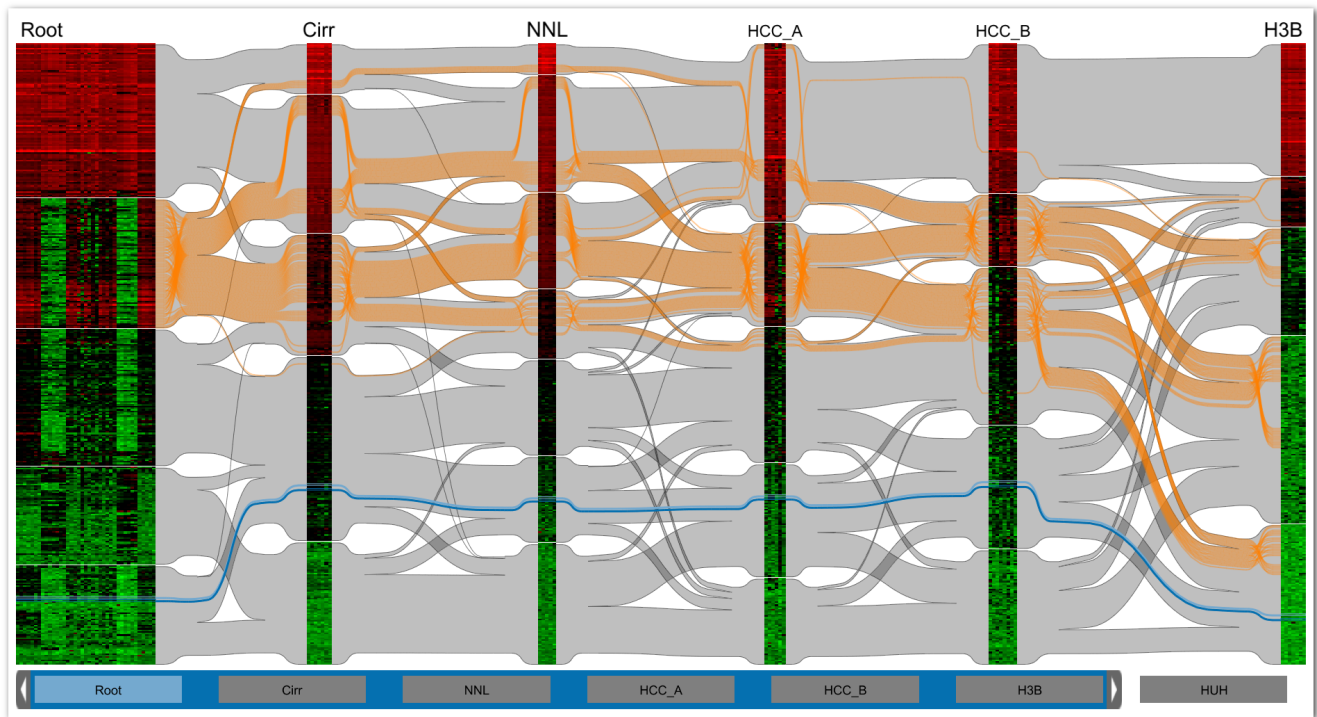

Fig. 7. The Matchmaker's overview displaying 39 different dimensions (78 in total) showing patient and cell line gene-expression data with 400 statistically filtered and clustered genes each. The dimensions are grouped according to diseases (*e.g.*, Cirr = Cirrhosis). The left heat map is the root group containing all experiments clustered together. Ribbons connect the experiments while abstracting the individual genes, showing the relations between clusters among the groups. While the genes in the Cirr group are clustered similarly to the combination of all dimensions in Root, many differences between clusters are evident between HCC_B and H3B. The orange overlay highlights all genes selected in the second cluster of the Root group, showing how this cluster spreads over the compared groups.
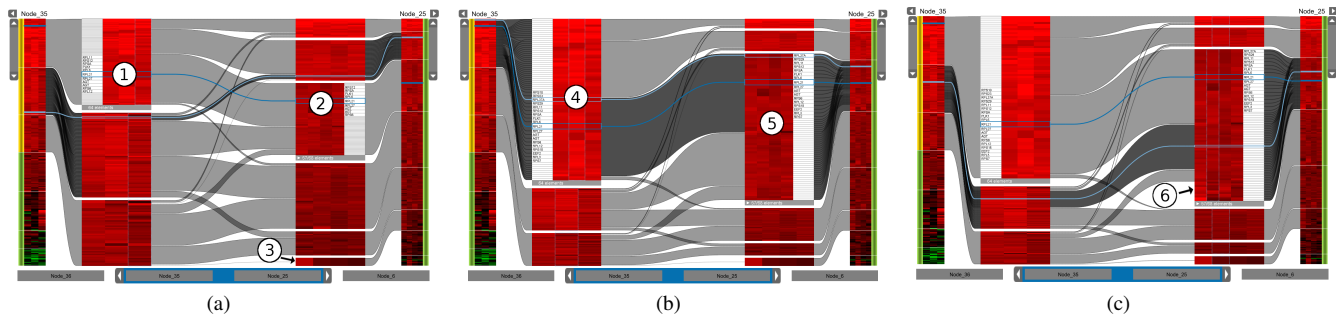
Fig. 8. Different states of the detail mode (also shown in Figure 1). (a) The detail view displaying three selected clusters in detail heat maps and one selected record for which orthogonal stretching is used to be able to show its caption (see (1) and (2)). Notice the single visible record in the detail heat map (3) out of the larger cluster to its right. The other elements are hidden because they do not occur in one of the selected clusters. (b) The detail view showing the same data with orthogonal stretching applied for the selected heat map ((4) and (5)). Notice that the heat maps with selected elements have more than twice the height as in (a). (c) The otherwise hidden elements (6).

makes the differences between the left and the right group easily recognizable. The exact nature of changes of the clusters is now obvious. A further visual improvement can be achieved by replacing the lines with spline curves, as shown in 5(c). While this visual representation is already very clear, due to the many parallel curves it suffers from Moiré patterns in some situations. Additionally, abstracting the individual connection lines using ribbons is an option (see Figure 5(d)). Matchmaker supports both, using individual curves and ribbons, and leaves it up to the user to choose. Ribbons have three advantages over individual curves: there are no Moiré patterns, they further reduce visual clutter and they improve rendering performance. This comes at the cost of hiding the associations of individual elements. To amend this, we employ a details-on-demand strategy: as soon as a user hovers over a ribbon the contained curves are rendered.

Overview Mode

The overview is the starting point after the user finishes the grouping and statistical preprocessing in the grouper view. In this mode we show several groups of dimensions and the connection curves or ribbons simultaneously. To minimize visual clutter, we have reduced the spacing between the control points in the overview mode, resulting in slimmer ribbons or bundled curves.

As discussed before, Matchmaker shares some basic properties with parallel coordinates. To alleviate the problems of static parallel coordinates such as following a set of lines across several axes or comparing between two particular axes [28] – which are equally relevant in Matchmaker – we provide the ability to brush records and rearrange axes, which are common in many parallel coordinates implementations. Figure 7 shows the overview with a brushed cluster. Notice that brushing for clusters or ribbons either via the highlight-on-hover feature or persistent brushing with colors is reflected in the whole overview, giving a good impression of the elements' distribution across all groups.

Interactive re-arranging is achieved by dragging the group's caption in the group bar to the desired position. The group bar always reflects the order of groups while the blue slider indicates which groups are visible. Individual re-clustering of a group, for example with different parameters, removing or duplicating of groups is achieved using the group bar's context menu.

In some cases, users may want to see only three or four groups to be able to inspect relations more clearly. This can be achieved by dragging the slider in the group bar at the bottom of the overview to include only the desired groups. The other groups' heat maps are hidden, but their caption remains visible in the group bar. The group bar therefore always helps the user to remember which other groups are currently available, albeit not visible.

Even though the overview is able to convey the main trends in the data, for a deeper understanding of the dataset a drill-down to the level of individual data records is necessary. To make this possible

Matchmaker uses a detail mode, which is activated by using the mouse wheel. We use animated transitions to switch between overview and detail mode, thus making the changes of the layout transparent to the user.

Detail Mode

Initially, the detail mode (see Figure 1), similar to the overview, presents the heat maps of two groups and the relations between them. However, several GUI elements were added: The cluster bar (at the outer sides of the heat maps in green respectively gold when selected) allows the user to pick individual clusters for detailed inspection. Multiple clusters can be selected by pressing the Ctrl-key while selecting the clusters. Furthermore, we provide a slider next to the cluster bar, which simplifies the selection of multiple clusters at the same time. Finally, buttons at the top corners allow the user to slide-in dendrograms, showing the relations between records as determined by a hierarchical clustering algorithm.

Figure 8 shows the detail view with three clusters selected for comparison in different states. Figure 8(a) shows the default spacing where every heat map has a height proportional to the number of elements it contains. For every record in one of the selected detail heat maps the clusters from the target group are selected. Records of the target group that are not in the selected source clusters are hidden, which is most evident in the single enlarged element at the bottom which belongs to a larger cluster (see (3) in Figure 8(a)). Hiding non-referenced records allows us to show the relevant, referenced records at maximum size. Hidden records are indicated by the caption in the gray tool-bar below the detail heat map, which is shown when a record is selected or the mouse is hovering over the heat map. In addition, the relation of size between the overview and the detail indicates hidden records.

The detail heat maps use orthogonal stretching for their records [24] to show currently selected and immediately surrounding records (see (1) and (2) in Figure 8(a)). Optionally, orthogonal stretching can also be employed for whole detail heat maps (cf. (4) and (5) in Figure 8(b)), enabling a more detailed analysis of the selected heat map. Compared to Figure 8(a), several more records have captions in Figure 8(b), since more space is available.

In some cases, hiding records might not be desirable, therefore it can be turned off. An example is shown in Figure 8(c), marked with (6). The previously hidden records at the bottom of the large heat map on the right are naturally not connected to records on the left. Showing or hiding can be triggered by clicking the button in the tool bar.

While individual records are rescaled to fit within the current size of the heat maps, we chose to define a minimum size for a detail heat map. This ensures that all heat maps in the detail view are usable and not reduced to only a couple of pixels. If the number of heat maps is too large to be shown simultaneously, some heat maps at the bottom are culled since they are out of the view frustum. They can be brought back into focus by reducing the number of selected heat maps.
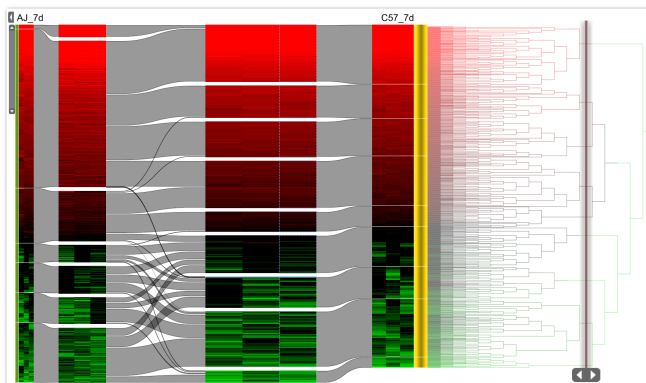
Fig. 9. Detail dendrogram with dynamic adjustment of hierarchy cut-off for cluster selection. Changes in cut-off are immediately reflected in both the overview and the detail heat maps.

Finally, the button at the top corners makes a dendrogram for dynamically adjusting the size of clusters appear (see Figure 9). To change the cluster size, the user can drag the cut-off bar up and down the hierarchy and the cluster borders are set where it is released.

3.4 Scalability

The proposed methods and the underlying implementation perform well for datasets with up to 100 dimensions and up to 2000 data records on standard hardware (an Intel Core Duo CPU with an NVIDIA GTX 8800 GPU and a 22 inch screen with a resolution of 1680x1050). The number of dimensions is not a hard limit, as they are managed in the grouper view where a subset of combined dimensions can be selected for the analysis. By default, the Matchmaker view can present up to 10 groups of which 6 can be rendered simultaneously, the rest is accessible using the group bar at the bottom. This was found to be a good compromise between the desire to show more data and the desire to avoid visual clutter. This can be changed in the settings to accommodate unconventional displays, for example. How many data records the Matchmaker can handle depends largely on the number of clusters and the similarity of the groups. Given the described hardware configuration, experiments showed that, for about 10 clusters, the technique can handle up to 3000 data records with acceptable visual clutter. However, since records cannot be resorted, a larger number of clusters or vastly different datasets result in a growing number of crossings. Our order-preserving bundling technique produces a readable overview comparison for up to 20 clusters for datasets with less than 2000 records. However, by using the detail mode for the cluster inspection, the user can analyze many more clusters.

3.5 Implementation

The Matchmaker technique is implemented in Java as a part of the Caleydo visualization framework[1] [19, 30]. For rendering, we use the Java binding for OpenGL (JOGL)[2]. We access R via rJava[3].

The images were produced using a real-life published dataset [15], except for Figures 2(b), 11(a) and 11(b) where we used the dataset discussed in Section 4.2. The data is a compound set of gene-expression experiments from patients with different diseases, on which we based the experiment grouping for the comparisons. The color coding for all heat maps is on a logarithmic scale. The color mapping from green to black to red is the standard for heat map visualizations of gene-expression data. We also provide alternative mappings suitable for red-green blind users. All other colors for both the visualization technique as well as the figures in this paper were taken from Color Brewer [2].

---

[1]http://www.caleydo.org

[2]http://kenai.com/projects/jogl

[3]http://www.rforge.net/rJava/ interface

4 CASE STUDIES

In the following, we will present two case studies of analysis conducted with the Matchmaker technique. The first shows how Matchmaker can visualize differences between clustering algorithms, the second explains a real-world use case for our technique in biomolecular data analysis.

4.1 Comparison of clustering algorithms

Usually, data analysis tools provide a wide range of clustering possibilities to the user. There are several types of clustering algorithms (*e.g.*, partitional vs. hierarchical, unsupervised vs. supervised) and other influential factors such as the choice of a distance measure or parameters. However, users are often not aware of the consequences of these factors, and cannot anticipate the results. Due to the flexible arrangement of dimension grouping in Matchmaker, the user can load the same data (sub)sets multiple times into the comparison view, showing each as an identical heat map. In turn, each of the underlying datasets can be clustered separately with either the same algorithm and varying parameters, or completely different algorithms. This way, Matchmaker enables a user to understand the impact of the cluster algorithms and its parameters applied to a concrete dataset. Consequently, the user can decide which clustering algorithm fits the data best. Figure 10 shows the clustering algorithm comparison scenario using data from [15]. Experiments (*i.e.*, dimensions) of two cell lines were grouped together and clustered multiple times using different algorithms: hierarchical clustering on the left, k-means clustering in the middle and affinity propagation on the right. All algorithms were parameterized so that they would choose a similar number of clusters and the same distance measure (Euclidean distance) was used. Figure 10 clearly shows that the k-means algorithm (used in (b)) assigned obviously differently expressed genes to the same cluster, while affinity propagation and the hierarchical clustering algorithm created separate clusters (highlighted in yellow and orange respectively in Figure 10). At the bottom of the heat maps k-means splits the group of genes, which both the tree clustering algorithm and affinity propagation assigned to one cluster, into three separate clusters, with no clear evidence of difference between the elements. This leads to the conclusion that the k-means algorithm is not a good choice for this data, while the two other algorithms achieve comprehensible – but still different – results.
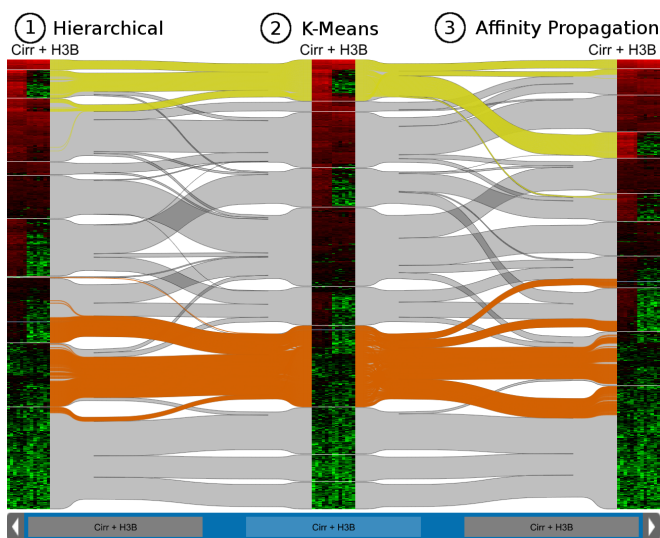

Fig. 10. A comparison of three clustering algorithms run with 1800 records: (1) hierarchical clustering, (2) k-means and (3) affinity propagation. The yellow and orange brushing show how the k-means algorithm assigned obviously different records to one cluster, while the other two worked as desired.
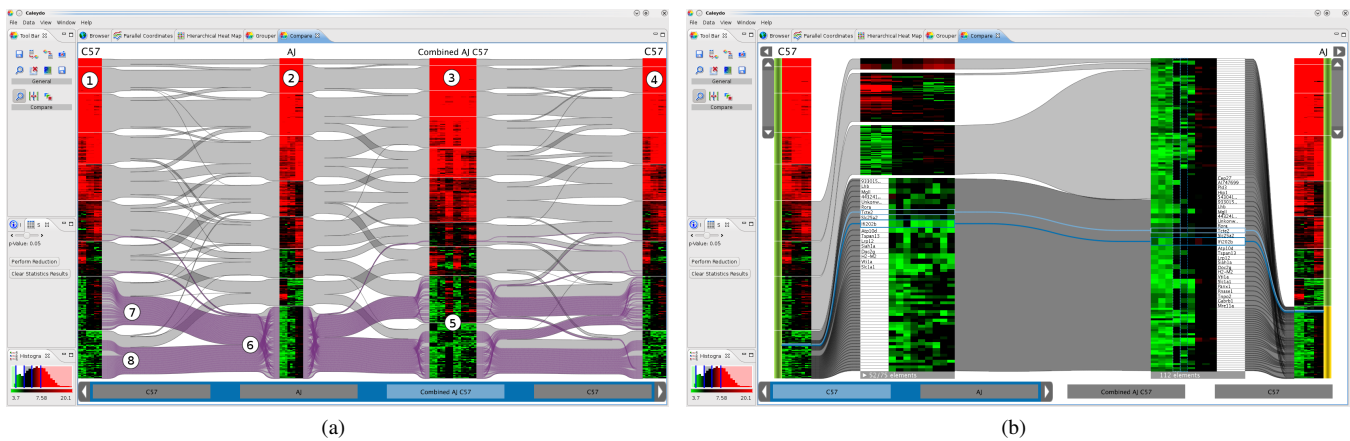
Fig. 11. Screenshots of the Caleydo Matchmaker taken during an analysis session by a biologist. (a) We see four groups (1-4). The first two, C57 and AJ are homogeneous. Each consisting of 9 experiments: control, 7 days of intoxication and 8 weeks of intoxication, with 3 replicates per category (from left to right). The third (Combined AJ and C57) contains all experiments from the first two groups. The fourth group is a copy of the first to enable better comparisons. Notice the inhomogeneous clusters for the combined group (5). Clustering the single columns yields more consistent results, allowing a biologist to assign meaning to a cluster. The biologist brushed the bottom cluster in AJ (6), identifying that the genes in this cluster are split into two groups in C57, one being similarly regulated over time to AJ (7), the other (8) containing genes not-deregulated (equally regulated) in C57 while upregulated (going up over time) in AJ (6). Since this difference may be important, he chose to explore this cluster in detail. (b) We see the deregulated cluster in AJ on the right, and the not-deregulated cluster for C57 containing the same genes on the left. By exploring the genes and using Caleydo's built-in features to find contextual information on genes, he was able to hypothesize that these genes are involved in apoptosis and thus alter the phenotype of the liver tissue by removal of cells damaged by oxidative stress.

4.2 Biomolecular analysis by a biologist

Our collaborators from the Medical University of Graz are studying why patients differ in their susceptibility to develop steatohepatitis (inflammation and fattiness of the liver) even when exposed to the same amount of steatohepatitis-inducing conditions like alcohol abuse, diabetes or obesity. The reason for this difference in susceptibility to steatohepatitis inducing agents has to be genetic, and the purpose of our partner's experiments are to define genetic regions or modifier genes which are differentially expressed in these two groups and are responsible for the different reaction to the same causative agent [18].

They use a mouse model of steatohepatitis induction, where animals develop steatohepatitis features, like ballooning of hepatocytes (break down of the cell's skeleton) and Mallory-Denk-Body formation (aggregates of misfolded proteins), after being fed with rodent chow supplemented with DDC (3,5-diethoxycarbonyl-1,4-dihydrocollidine) for 8 weeks [8]. Our collaborators identified two mouse strains, A/J (AJ) and C57Bl6/J (C57), which show distinct phenotypes upon DDC feeding. By histological analysis of liver tissue it is possible to determine that AJ mice develop steatohepatitic features, whereas C57 mice do not. To determine which genes are differentially deregulated in the two mouse strains, they performed an experiment where three groups of animals in each strain were fed with DDC for 8 weeks, 7 days or not at all (control). Gene expression data was generated from the liver tissue of these animals using whole genome microarrays (Applied Biosystems Inc). The analysis of this data involves finding deregulated genes (*i.e.*, changing expression over time) in the course of DDC feeding in AJ animals, the responder strain, but are not deregulated in the C57 animals, and vice versa. This analysis is difficult to perform with traditional tools, which do not treat the groups individually.

Using the Caleydo framework and the Matchmaker visualization technique, they were able to perform cluster analyses on the DDC feeding experiment in each mouse strain separately. Figure 11 shows two screenshots taken during an analysis session by a biologist. Figure 11(a) shows the regulation over time (control, 7 days and 8 weeks, with 3 experiment replicates each) for different mouse strains: on the left the C57 strain (1), next to it the AJ strain (2), then a group where both were combined (3) and a duplicate of the C57 strain (4). In the overview, we see that the bottom two clusters are very inhomogeneous (5). When following the highlights it becomes obvious that if the clustering is done on a single strain the genes present in

the highlighted cluster in AJ are being split up into two clusters in C57 (7 and 8 in Figure 11(a)). One of those clusters in C57 contains genes not-deregulated (equal over time) in C57 (8). The expert noted that these genes might be important in the different reaction of C57 to DDC intoxication. He then continued to analyze this cluster in more detail (Figure 11 (b)). While browsing the list of genes in this cluster he found several genes involved in the regulation of apoptosis (programmed cell death) which might cause cellular turnover in the liver and alter the phenotype by removing cells damaged by oxidative stress. The removal of these damaged cells, which are prone to ballooning and have Mallory-Denk bodies by apoptosis, could be a reason why these features of steatohepatitis are absent in C57.

The expert stated that for him the key advantage of clustering distinct groups (AJ and C57) separately is that he can quickly assign a biological meaning to a cluster (for example "up-regulated in AJ"). Matchmaker then enabled him to follow these genes in the other strain and see how they behave there. This is more difficult if the groups are clustered together, as the clustering algorithm tries to find a best match over both groups and thus makes the clusters inhomogeneous.

4.3 Discussion

When observing our users during the case studies we noticed that the process of data preparation (filtering, choosing and generating groups, running clustering algorithms on the groups) needs to be improved. While this was not the focus of our research for this paper, it is crucial for an adoption by end-users that this process is made intuitive.

For the Matchmaker interface itself, feedback on ease of use was positive throughout. However, we noticed significant differences of how easily users understand the benefits of the methods for the two use cases. When comparing clustering algorithms, the meaning of the groups and their relations are immediately obvious - one group corresponds to one clustering algorithm and all groups show the same data. However, for biomolecular analysis where meaningful sub-spaces of the data need to be created in order to benefit from the Matchmaker technique, a more thorough introduction was necessary. Only after the expert was instructed that clusters are now largely homogeneous, allowing him to easily identify how clusters change between groups, did he realize the benefits for his application.

5 CONCLUSION AND FUTURE WORK

In this paper we have presented Matchmaker, a visualization technique that makes it possible to split and individually combine a multidimensional dataset into several groups of dimensions, run clustering algorithms on these groups separately and then visually compare the results. This enables users to find patterns in the data which otherwise would be obscured, and to compare the effects of different clustering algorithms. In an informal case study we have shown that our technique is a valuable tool for biomolecular data analysis, especially combined with other features of Caleydo which help bring the raw data into their biological context. We also demonstrate how the technique can be used to evaluate the quality and properties of clustering algorithms or their parameters. We believe that this can be very helpful in choosing the right clustering algorithm for a wide audience.

In the future, we want to make our technique more scalable, to be able to visualize whole-genome data and also data generated by other high-throughput techniques like metabolomics, proteomics and automated analysis of tissue sections. This will also be very relevant when these individual datasets are combined in a holistic analysis for the upcoming field of systems biology. We plan to do so by introducing methods that hint at potentially interesting patterns while hiding uninteresting ones. Furthermore, an additional level of focus, as already used in Caleydo's hierarchical heat map [19], will help dealing with large-scale data. However, sensible routing of connection curves that still convey trends is difficult in such a hierarchical setup and therefore will be an interesting challenge.

REFERENCES

[1] J. Bertin and G. Jensch. *Graphische Semiologie: Diagramme, Netze, Karten.* de Gruyter, Berlin, first published 1967, german edition, 1974.

[2] C. A. Brewer. Colorbrewer. http://www.ColorBrewer.org, last accessed March 27, 2010, 2010.

[3] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Academy of Science USA*, 95(25):14863–14868, Dec. 1998.

[4] B. J. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, Jan. 2007.

[5] M. Graham and J. Kennedy. Combining linking & focusing techniques for a multiple hierarchy visualisation. In *Proceedings of the Fifth International Conference on Information Visualisation*, page 425. IEEE Computer Society, 2001.

[6] M. Graham and J. Kennedy. Exploring multiple trees through DAG representations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1294–1301, 2007.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[8] S. Hanada, P. Strnad, E. M. Brunt, and M. B. Omary. The genetic background modulates susceptibility to mouse liver Mallory-Denk body formation and liver injury. *Hepatology (Baltimore, Md.)*, 48(3):943–952, Sept. 2008. PMID: 18697208.

[9] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of the 7th conference on Visualization '96*, pages 263–ff., San Francisco, Ca., United States, 1996. IEEE Computer Society Press.

[10] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741748, 2006.

[11] D. Holten and J. J. van Wijk. Visual comparison of hierarchically organized data. *Computer Graphics Forum*, 27:759–766(8), 2008.

[12] J. Hong, J. D'Andries, M. Richman, and M. Westfall. Zoomology: comparing two large hierarchical trees. In *Posters Compendium of Information Visualization 2003 (Seattle, WA, USA)*, pages 120–121, 2003.

[13] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. of the First IEEE Conference on Visualization*, pages 361–378, San Francisco, CA, USA, 1990.

[14] I. Jolliffe. *Principal Component Analysis.* Springer, 2nd edition, Oct. 2002.

[15] K. Kashofer, M. M. Tschernatsch, H. J. Mischinger, F. Iberer, and K. Zatloukal. The disease relevance of human hepatocellular xenograft models: molecular characterization and review of the literature. *Cancer Letters*, 286(1):121–128, Dec. 2009. PMID: 19111389.

[16] 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.

[17] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009.

[18] C. Lackner, M. Gogg-Kamerer, K. Zatloukal, C. Stumptner, E. M. Brunt, and H. Denk. Ballooned hepatocytes in steatohepatitis: the value of keratin immunohistochemistry for diagnosis. *Journal of Hepatology*, 48(5):821–828, May 2008. PMID: 18329127.

[19] 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 *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, Taipei, Taiwan, 2010.

[20] T. V. Long and L. Linsen. MultiClusterTree: interactive visual exploration of hierarchical clusters in multidimensional multivariate data. *Computer Graphics Forum*, 28(3):823–830, 2009.

[21] M. Meyer, T. Munzner, and H. Pfister. MizBee: a multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):897–904, 2009.

[22] T. Munzner, F. Guimbretire, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In *ACM SIGGRAPH 2003 Papers*, pages 453–462, San Diego, California, 2003. ACM.

[23] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *International Conference on Coordinated and Multiple Views in Exploratory Visualization*, volume 0, pages 61–71, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[24] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proc. of the 6th annual ACM symposium on User interface software and technology*, pages 81–91, Atlanta, Ga., United States, 1993. ACM.

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

[26] J. Sharko, G. G. Grinstein, K. A. Marx, J. Zhou, C. Cheng, S. Odelberg, and H. Simon. Heat map visualizations allow comparison of multiple clustering results and evaluation of dataset quality: Application to microarray data. In *Proceedings of the 11th International Conference Information Visualization*, pages 521–526. IEEE Computer Society, 2007.

[27] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings on Visual Languages*. IEEE Computer Society, 1996.

[28] H. Siirtola and K. Rih. Discussion: Interacting with parallel coordinates. *Interact. Comput.*, 18(6):12781309, 2006.

[29] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional databases. *Commun. ACM*, 51(11):7584, 2008.

[30] M. Streit, A. Lex, M. Kalkusch, K. Zatloukal, and D. Schmalstieg. Caleydo: Connecting pathways and gene expression. *Bioinformatics*, 25(20):2760–2761, July 2009.

[31] R. D. C. Team. *R: A Language and Environment for Statistical Computing.* Vienna, Austria, 2009. ISBN 3-900051-07-0.

[32] A. Telea and D. Auber. Code flows: Visualizing structural evolution of source code. *Computer Graphics Forum*, 27(3):831–838, 2008.

[33] Y. Tu and H. Shen. Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1286–1293, 2007.