

Visual Comparison of Orderings and Rankings

Michael Behrisch¹, James Davey², Svenja Simon¹, Tobias Schreck¹, Daniel Keim¹ and Jörn Kohlhammer²

¹Universität Konstanz, Germany

²Fraunhofer IGD, Germany

Abstract

In many data analysis problems, sequentially ordered (or ranked) data occurs that needs to be understood and compared. Ranking information is essential in applications such as multimedia search where retrieval rankings need to be inspected; alignments of gene sequences in bio-molecular applications; or for a more abstract example, considering the permutations of rows and columns for purpose of matrix visualization. In each of these examples, often many different orderings of a given data set are possible. E.g., a search engine may produce, based on different user parameterizations, different rankings. A relevant problem then is to understand the commonalities and differences of a potentially large set of rankings. E.g., finding global or partial orderings in which different ranking or sorting algorithms agree can support the certainty in the respective ranking by the user.

We consider the problem of comparing sets of rankings with these questions in mind. We present an approach for a visual comparison of sets of rankings that effectively allows to spot commonalities and differences among rankings. The approach relies on a small-multiple view of glyphs each of which visually contrasts a pair of rankings. The glyph in turn is defined on a radial node-link representation which allows effective perception of agreements and differences in pairs of rankings. We apply our approach on different use cases and demonstrate its effectiveness in spotting patterns of similarity and differences in sets of rankings.

1. Introduction

Retrieval systems are omnipresent and indispensable components for information-centered work. However, different retrieval systems may provide deviating rankings, the joint consideration of which may be important. Further, in analysis domains such as bioinformatics or network security, decision making processes are based on sets of rankings. A central problem is that users are often not able to assess the quality and/or stability of a given ranking, since alternative rankings are often neither presented nor computed as a reference. This problem is inherently prominent whenever rankings are influenced by user-/system-determined parameter settings, such as the used similarity function, feature vector representation, or the underlying retrieval algorithm.

We are considering the problem of comparing large sets of rankings. We devise a solution to this problem inspired by Shneiderman's Visual Information Seeking Mantra [Shn96]. Specifically, we define three comparison levels of interest and corresponding visualization support as follows:

(1) The first comparison level refers to *overviewing of rankings*. In our case, this corresponds to all possible com-

binations of rankings. In this N:N comparison task, a goal is to identify consistent from contradictory results by visual means. A matrix representation is a straightforward tool. Correlating structures among the matrix cells can be identified, similar like in the Scatter Plot Matrix approach [CCKT83] for high-dimensional data.

(2) A row-wise or column-wise analysis can take place in the comparison matrix, corresponding to a *more detailed* comparison level (1:N comparison task). The goal is to comprehend, which comparison ranking has the most consensus (or disagreement) with respect to the default ranking.

(3) When the user is able to identify one interesting comparison view the task changes to a *detail-on-demand* view. Specifically, 1:1 comparison views among rankings can be selected by users.

We will next follow, after a discussion of related work, this structure to develop our rank-based visual analysis tool.

2. Related Work

Much work exists that studies visually analyzing and comparing sequential (ordering, ranking) data. The notion of se-

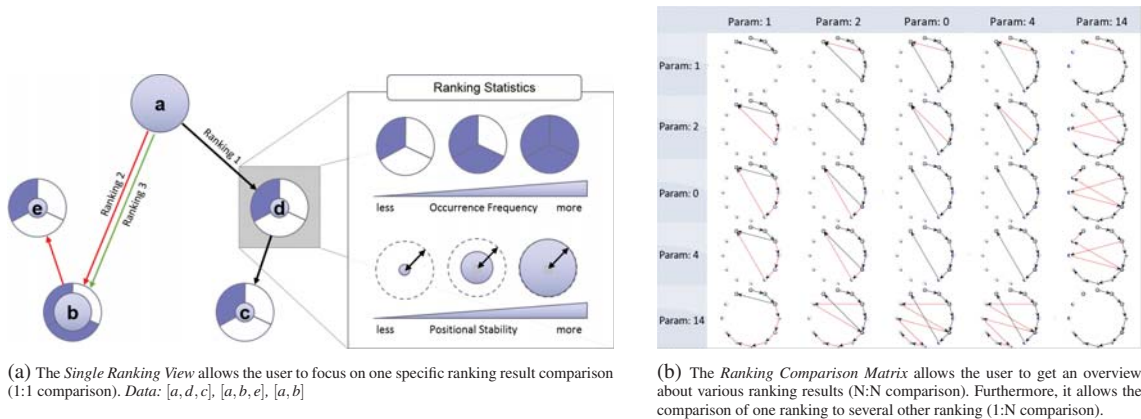


Figure 1: The primary views of the ordering and ranking comparison interface are subdivided into the *Ranking Comparison Matrix* (b) and *Single Ranking View* (a). A *Details View* complements the interface with meta information.

quential data per se is very broad and comprises many applications. The article of Gleicher et al. [GAW*11] surveys and structures the solution space for visual comparisons of different data types. Here, we can only discuss a selection of aspects here.

Generally, time series are an important instance of sequential data. Time series visualization is concerned with visual mappings for series of measurements, typically given by quantitative, equally-spaced consecutive values [AMST11]. The comparison of two or more sequential data sets is a key problem in many applications. In fact, many time series visualization techniques were designed for comparison tasks, such as dense pixel-based approaches for comparing large numbers of time series [KAK95]. The elements of a series or sequence can also be symbolic, as e.g., in DNA sequences. The analysis of sequences of values may include relationships among them. An example are sequences of email messages sharing reply/forward relationships [Ker03].

Techniques exist which allow to compare data which is inherently non-sequential, by finding a linear mapping of data elements, on which then sequence visualization can be applied. Examples include the TreeJuxtaposer [MGT*03] system, which compares pairs of hierarchies side-by-side by finding correspondences between tree nodes mapped in sequential order (e.g., by a dendrogram). Another example is given in [HvW08], where pairs of hierarchies are compared by linear (icicle) mappings with bundled connectors showing element relationships. A further example is the TimeArc-Trees [GBD09] approach for comparing sequences of directed graphs. It is based on a linear mapping of nodes, a sequence of which is shown with nodes aligned for comparability.

We here are interested to compare for differences in the positions of elements among sets of sequences. Our ap-

proach is inspired by the Scatter Plot Matrix technique [CCKT83], allowing to compare pairwise combinations of variables in high-dimensional data. Matrix structures have been exploited previously for comparison of relational data, e.g., in [BN11, GHS10, SM07]. Small-multiple views of graphs for comparison based on clustering and projection have been proposed in [vLGS09]. Our method is novel in that we combine a matrix approach with a custom glyph, based on a radial network layout, to compare the differences among pairs of sequences with permutations of its data elements.

3. Visual Sequence and Ranking Comparison

We next describe our ranking comparison visualization approach which follows the analysis steps outlined in Section 1. For presentation purposes, we start with a glyph for a detailed ranking comparison, based on which we form small-multiple views for comparing sets of rankings.

3.1. Comparison View Glyph (1:1 Comparison)

The comparison glyph serves to identify the consensus, respectively disagreement, between pairs of rankings. Figure 1a depicts its design. A clock-wise circular layout of nodes encodes one selected baseline ranking. The nodes are positioned according to their index position in the ranking. Additionally, black arcs represent the base ranking sequence. A second ranking to compare against is then overlaid by inserting red arcs into the base ranking glyph. We assume the element sets of the rankings to largely overlap, but there may be elements present in only one ranking. To ensure comparability, we position all nodes that occur in both ranking sets to the position given by the baseline ranking, whereas additional nodes from the comparison ranking(s) are inserted at the end of the baseline ranking. As a result, the structure of

the glyph arcs are a visual indicator for the degree of agreement between the two rankings. More rankings can be displayed on top of the base ranking, each resulting in distinctively colored edge sets (Figure 1a exemplifies a third ranking with green arcs). Considering Figure 1a, the rankings differ (a) in their retrieved result list size (the black-colored ranking comprises three items, the green-colored ranking two) and (b) in the ordering ($a - d - c$ versus $a - b - e$).

We visually encode additional information regarding ranking positions and occurrence frequency in the nodes of the glyph. (1) Most use-cases require to assess the amount of occurrences for one specific result item among all investigated ranking lists (e.g., found in every/none/some of the investigated ranking list) and (2) the user wants to investigate the stability regarding positional changes (e.g., found always on first position). Thus, we encode the agreement on the position for this specific item among all investigated ranking lists in the glyph. As Figure 1a depicts, for example node d and c are found only in one of the selected experiments. Hence, a pie-chart like metaphor represents this aspect. The more rankings are under investigation the smaller the portioning of the pie-chart. For demonstration purposes, we are adding the ranking result $[a, b]$ to the example above. As Figure 1a then depicts, node d was found in one of the three selected experiments.

In addition to that, the positional agreement is encoded by the diameter of an overlay on top of the pie-chart. For example, the item a is ranked by all three selected rankings on the same position. Hence, the diameter is 100% of the node's size and explicitly hides the double-encoded occurrence information. On the contrary, only two rankings disagree on the position of b , thus leading to a smaller (66%) overlay.

3.2. Ranking Matrix (1:N and N:N Comparison)

A matrix of ranking comparison glyphs facilitates the 1:N and N:N comparison tasks, similar to a Scatter Plot Matrix. The vertical axis spans the space of base rankings, over which all other rankings in the data set are overlaid each one along horizontal direction. Along each row, the same baseline ranking is compared against all other rankings, as Figure 1b illustrates.

The ranking matrix can be sorted according to specific criteria. The current implementation sorts similar ranking comparisons to the upper left corner by considering the amount of reoccurring items among the two selected rankings. Further sorting approaches, e.g., considering inter-comparison of edge crossings could be useful and we want to explore them in future work.

The visual task for the user is to assess the matrix's compliance (most rankings find a consensus or disagree) and conduct a visual pattern search for correlations. These correlations form visually groupings of similar ranking result sets, as for example Figure 3 depicts. Whenever the user wishes to investigate one ranking comparison in detail, one

matrix comparison can be visualized separately in a maximized comparison view panel.

4. Application Examples

We next discuss three different use cases to illustrate the applicability of our approach.

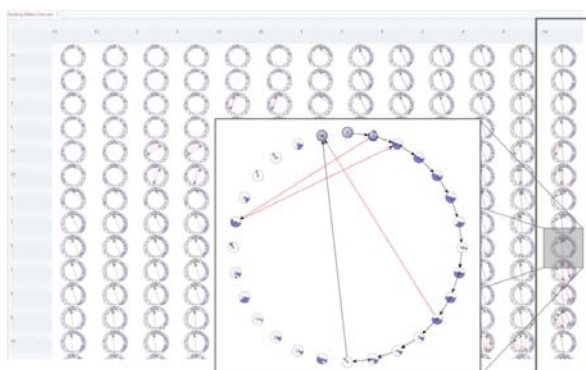
4.1. Sequence Data

In bio-molecular research, the BLAST [AMS*97] algorithm is a well-known approach to perform a similarity search against a database of genes with a given query sequence. The result is a list of genes ordered by similarity and limited by a significance threshold. A high similarity of a retrieved gene to a query suggests that the gene and the query have a common ancestor, and probably serve the same biologic function. BLAST requires a set of parameters to be provided. Typically, it is used with default settings, yet it is known to be sensitive to parameter changes. It is therefore of interest to compare result lists from different runs with varying parameter settings to the default settings. Result lists from different parameter settings can either include more or less genes, furthermore the order can be different.

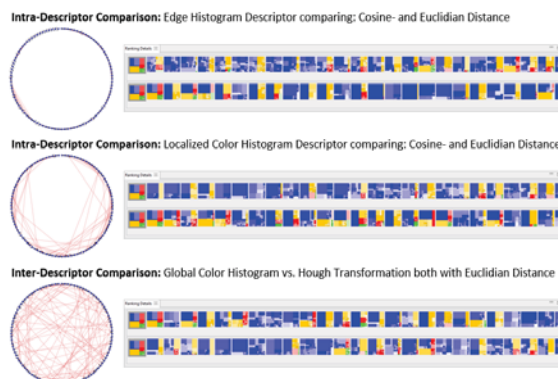
Figure 2a shows the comparison of the result ranking for the default settings compared (used as base) against rankings obtained by twelve alternative parameter settings. It is clearly recognizable that nearly half of the genes are not found with the default parameter settings and that the order of the found genes differs between parameter settings. However, some settings result in no positional changes and others show similar changes among each other, which can be explained by only small parameter variations between settings. By means of the pie-chart node representation, the single ranking view also allows to assess that genes which have not been found with the default settings are only found with a small number of parameter settings. Moreover, the positional accuracy encoding of node "1" depicts that this gene was always the most similar gene settings. Therefore, this result item can be considered the most stable.

4.2. Ranking of Data Views

The search for similar images is a prominent task in multimedia retrieval. It typically relies on image descriptors and according similarity functions, of which many different alternatives exist. We consider an example of comparing rankings of TreeMap views. Alternative rankings are given by using different descriptors and similarity functions. We consider a set of standard descriptors (including Global and Local Color Histogram, Local Edge Histogram, and Hough Transformation descriptor) and similarity functions (Euclidean, Cosine Distance, Dice coefficient). We consider a set of 100 artificially created TreeMap views ranked against a given query view and using different combinations of descriptors and similarity functions. Our question is, which combinations result in similar rankings.



(a) Visual comparison of gene sequence data in a biological data use case (1:N comparison).



(b) Visual comparison of image retrieval results obtained using different image descriptors and similarity functions (1:1 comparison).

Figure 2: The visual ordering comparison is demonstrated by three real-life use cases from different domains.

Figure 2b shows illustrative results for the comparison of rankings along different descriptors and similarity functions. One can visually depict that despite the very different definition of the rankings, some combinations provide similar rankings. E.g., in the intra-descriptor analysis (varying the similarity function, but not the descriptor) the Edge Histogram descriptor results in a similar ranking result when comparing Cosine- and Euclidean distance. In an inter-descriptor comparison (varying the feature descriptor, but not the similarity function) it becomes obvious that the Global Color Histogram descriptor delivers significantly diverging ranking results as the Hough Transformation.

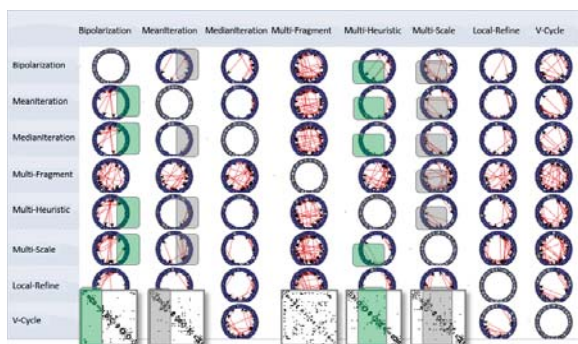


Figure 3: Visual comparison of 2D sortings in a matrix sorting use case (N:N comparison)

4.3. Matrix Sorting [2D]

Finally, we consider a more abstract use case. Effective matrix visualization typically requires appropriate matrix sorting. To date, many matrix sorting algorithms have been proposed, and we can apply our tool also to compare such sorting algorithms. We use matrix data from the Jordi Petit test

suite [Pet03], and a set of eight matrix sorting algorithms for illustrative purposes. In the N:N comparison of the matrix sortings depicted in Figure 3 we can assess the matrices' sorting conformity and conduct a visual pattern search for correlations. It stands out that a larger amount of edge crossings identifies the *Multi-Fragment* sorting algorithm as the most disagreed ranking result (also depicted by the matrix image on the bottom of the column). In a more detailed view, the *gray areas* become of interest: Here some of the algorithms disagree on a part of the ranking list. However, this is contrasted by the *green areas*, representing algorithms with a large consensus among the ranking results.

5. Conclusion and Future Work

We presented an approach for visual analysis of sets of rankings, important in areas such as multimedia retrieval, bioinformatics, and others. We defined a glyph for pairwise ranking comparison, which is used in a small-multiple layout for overviewing and also, for detail-on-demand. The approach allows to identify structures in a potentially, large space of alternative rankings. A straightforward next future work includes developing improved methods for sorting the glyph matrix for visual similarity of the arc-link structures. We also will explore how the approach can benefit further application areas. Considering the matrix sorting use case, one idea is to include user feedback to steer matrix sorting approaches, based on the comparative views of candidate sortings.

Acknowledgments

This work has been partly funded by the German Research Foundation (DFG) under the grant SPP 1395 (Information and Communication Theory in Molecular Biology, InKoM-Bio), project 'Finding new overlapping genes and their theory (FOG-Theory)'. Additionally, we thank Michael Hundt for valuable discussions that contributed to this work.

References

- [AMS*97] ALTSCHUL S. F., MADDEN T. L., SCHÄFFER A. A., ZHANG J., ZHANG Z., MILLER W., LIPMAN D. J.: Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res* 25, 17 (Sep 1997), 3389–3402. 3
- [AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011. 2
- [BN11] BRANDES U., NICK B.: Asymmetric relations in longitudinal social networks. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2283–2290. 2
- [CCKT83] CHAMBERS J. M., CLEVELAND W. S., KLEINER B., TUKEY P. A.: *Graphical Methods for Data Analysis*. Chapman and Hall, New York, 1983. 1, 2
- [GAW*11] GLEICHER M., ALBERS D., WALKER R., JUSUFI I., HANSEN C. D., ROBERTS J. C.: Visual comparison for information visualization. *Information Visualization* 10, 4 (oct 2011), 289–309. 2
- [GBD09] GREILICH M., BURCH M., DIEHL S.: Visualizing the evolution of compound digraphs with timearctrees. *Comput. Graph. Forum* 28, 3 (2009), 975–982. 2
- [GHS10] GLEICHER M., HATFIELD D., SHAFFER D.: Comparing epistemic frames: An exercise in visual comparison. In *Eurovis 2010 Poster Proceedings* (jun 2010). 2
- [HvW08] HOLTEN D., VAN WIJK J. J.: Visual comparison of hierarchically organized data. *Comput. Graph. Forum* 27, 3 (2008), 759–766. 2
- [KAK95] KEIM D. A., ANKERST M., KRIEGEL H.-P.: Recursive pattern: A technique for visualizing very large amounts of data. In *IEEE Visualization* (1995), pp. 279–286. 2
- [Ker03] KERR B.: Thread arcs: an email thread visualization. In *Proc. IEEE Symposium on Information Visualization* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 211–218. 2
- [MGT*03] MUNZNER T., GUIMBRETIÈRE F., TASIRAN S., ZHANG L., ZHOU Y.: Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.* 22, 3 (July 2003), 453–462. 2
- [Pet03] PETIT J.: Experiments on the minimum linear arrangement problem. *J. Exp. Algorithmics* 8 (Dec. 2003). 4
- [Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), VL '96, IEEE Computer Society, pp. 336–. 1
- [SM07] SHENY Z., MA K.-L.: Path visualization for adjacency matrices. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC conference on Visualization* (Aire-la-Ville, Switzerland, Switzerland, 2007), EUROVIS'07, Eurographics Association, pp. 83–90. 2
- [vLGS09] VON LANDESBERGER T., GÖRNER M., SCHRECK T.: Visual analysis of graphs with multiple connected components. In *Proc. IEEE Symposium on Visual Analytics Science and Technology* (2009), IEEE Computer Society, pp. 155–162. 2