Preprint Version - Accepted, to Appear:
IEEE Transactions on Visualization and Computer Graphics 24(1), January 2018, Proceedings InfoVis 2017.

Considerations for Visualizing Comparison

Michael Gleicher *Member, IEEE*

Abstract— Supporting comparison is a common and diverse challenge in visualization. Such support is difficult to design because solutions must address both the specifics of their scenario as well as the general issues of comparison. This paper aids designers by providing a strategy for considering those general issues. It presents four considerations that abstract comparison. These considerations identify issues and categorize solutions in a domain independent manner. The first considers how the common elements of comparison—a *target* set of items that are related and an *action* the user wants to perform on that relationship—are present in an analysis problem. The second considers why these elements lead to challenges because of their scale, in number of items, complexity of items, or complexity of relationship. The third considers what strategies address the identified scaling challenges, grouping solutions into three broad categories. The fourth considers which visual designs map to these strategies to provide solutions for a comparison analysis problem. In sequence, these considerations provide a process for developers to consider support for comparison in the design of visualization tools. Case studies show how these considerations can help in the design and evaluation of visualization solutions for comparison problems.

Index Terms—Information Visualization, Comparison, Taxonomies, Visualization Models, Task Analysis.

1 Introduction

Comparison is a common thread in data analysis and visualization tasks. It might involve looking for differences between two CT scans, looking for similarities between several fluid flows, finding trends in a set of social networks, or finding common patterns in a library of genetic sequences. Regardless of the data type or domain, analysis often requires understanding the relationships among multiple objects. Such comparisons are often challenging as they combine the issues of individual objects and their relationships. This paper provides a framework for considering comparison to aid in designing tools to help users with comparison tasks.

For example, consider the scenario of an instructor interpreting email traffic data for a class with project groups. Comparison is more than just finding differences. For example, the instructor may want to: compare traffic patterns between groups to identify differences and connect these differences to project performance; compare individual students to dissect how similar usage patterns lead to desirable outcomes; or compare overall patterns to their expectations to determine if assignments are working as planned. Existing frameworks for visualization design (e.g., [59]) can help designers address such specific scenarios. For example, data abstraction considers the email data as a weighted network, allowing the use of principles and designs from other network problems (such as biological networks). Task abstraction helps identify the user's needs to match them to a tool design. However, such frameworks rarely break tasks down beyond the broad "compare" (Sect. 2.2), and offer little guidance in how to map these tasks to comparison solutions.

This paper provides an abstract framework to aid in designing solutions for scenarios involving comparison. The framework consists of a series of four considerations that help understand comparison tasks, their challenges, and their potential solutions:

- *Identify the Comparative **Elements**:* Comparisons involve two elements, a set of *targets* (i.e., the set of items being compared) and an *action* performed on the relationships (e.g., similarities and differences) among these targets. In the class email examples, the targets include group traffic patterns, individual usage patterns, and the instructor's expectations. The actions on relationships include identifying them, connecting them to outcomes, and dissecting them to find explanations. Section 3 discusses the comparative elements. Identifying targets and actions in a comparison provides a basis for understanding the challenges a visualization should address.
- *Identify the Comparative **Challenges**:* Comparisons grow difficult for three categories of reasons: the number of items to compare, the size or complexity of the items, and the size or complexity of the relationships between items. In the example, the number of items (students or groups) is unlikely to grow very large, but the individual items (e.g., the traffic patterns within a group) and the relationships between them are complicated. Section 4 describes the categories of comparative challenges. Identifying the key challenges in a comparison helps in selecting a strategy to address it.
- *Identify a Comparative **Strategy**:* The comparative challenges all involve a scalability problem. Section 5 categorizes solutions to such problems into three broad strategies: scanning sequentially, selecting a subset, and summarization. Matching strategy to challenges and user needs is important. In the class email example, finding outliers in long lists may be best supported by a scanning strategy, whereas a summarization strategy may be more appropriate for handling very complex patterns within groups. Section 5 describes the broad categories and how they may be supported in comparison applications.
- *Identify a Comparative **Design**:* Prior papers [37, 77] suggest that visual designs for comparison fall into three categories: juxtaposition, superposition, and explicit encoding. The choice of a design must align with the other considerations. In the example, the complex relationships between group traffic patterns may make superposition designs inappropriate or a summarization strategy might lead naturally to an explicit encoding. Section 6 reviews the visual design categories and connects them to the other considerations.

Comparison is a common need for users, and visualization can often help. Visualization can support comparison without explicitly addressing it. For example, a tool designed to examine one object at a time can be used for comparison by relying on the viewer's memory, or objects can be compared using multiple displays in separate windows. However, comparison tasks are best supported by tools designed to address comparison. The four considerations provide a framework (summarized in Fig. 1) that can help understand why such explicit support for comparison is helpful and how it can be designed by exposing the challenges in comparison scenarios and matching these challenges to solutions.

The primary contribution of this paper is a framework that abstracts comparison tasks and the approaches that support them. The framework comprises a series of four considerations that may be used in

- *Michael Gleicher is with the University of Wisconsin–Madison. E-mail: gleicher@cs.wisc.edu.*

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

Authors pre-print.
DOI 10.1109/TVCG.2017.2744199.
Paper is accepted and will appear in October, 2017

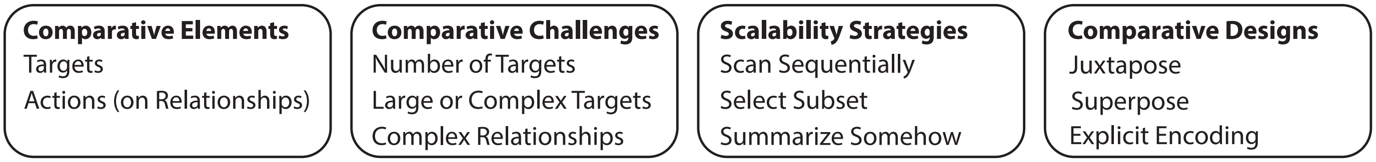


Fig. 1. Overview of the four considerations of comparison and the abstract categories they impose.

sequence to understand a comparison task and to help design visualization support for it. The considerations help identify how specific support for comparison can aid users with their tasks. The paper provides concise categorizations of the spaces[1] of the answers to the questions that can serve to organize problems and prior solutions. The paper also provides examples, in the form of case studies, that illustrate the value of this approach in designing comparative tools.

Visualization is only one of many approaches to help with comparison. These various analytics approaches fit together: statistical, computational and visual tools are often combined. This paper focuses on visualization for comparison, although its considerations may apply more broadly and can help articulate visualization's role in the broader toolbox. Similarly, this paper treats comparison as a special type of analysis, with unique challenges that are worthy of specific consideration. However, these considerations may apply more broadly. Indeed, one may take the viewpoint that much, if not most, of analysis can be viewed as comparison.

2 BACKGROUND

This paper uses the term comparison in its broad, common usage. The dictionary definition of comparison (specifically of its root verb compare) is "the examination of the similarities and/or differences between two, or among a set of, items[2]." This definition has two clear elements: a *target* "set of items" and an *action* (e.g., examination), performed on the relationship (e.g., similarities and differences) between them.

This paper avoids classifying systems as "comparison" and "non-comparison." It is difficult to define a clear and consistent binary criteria. Users often find ways to perform comparative tasks even if systems were not explicitly designed to support them. A system may serve comparative goals or provide lessons for the designers of future systems even if the authors have not written about their approach in terms of comparison, or explicitly thought about comparison in the design and development of their system. Sect. 3 discusses how we can view tasks in terms of having two elements, targets and actions, rather than defining an absolute standard of being a comparison.

2.1 Considering Visual Comparisons

There is a long historical tradition of creating visual designs that better support comparison. For example, Tufte's volumes (especially [78]) give many historical examples, and Playfair's initial "invention" of the line graph was to illustrate a comparison (see [21] or [16, Fig. 6]). Visualization tools can explicitly support comparison in many ways. For example, they may make it easier to address practical issues in viewing multiple objects side-by-side (e.g., [1]), provide interaction techniques to help examine such juxtaposed views (e.g., multi-view coordination [67] or guaranteed visibility view controls [60]), use visual designs to make such views more effective, or provide alternatives to juxtaposition. The diversity of comparative tools exhibits a diversity of comparative designs. This paper provides an abstract framework to help understand and organize this range.

Prior work surveys the range of visual solutions to comparison problems. Some surveys focus on particular visual designs, such as Tufte's chapter on small multiples [78]. Others focus on the range of solutions for a specific data type. For example, Graham and Kennedy [39] survey a range of visual mechanisms to compare trees. Several other surveys consider methods for comparing flow fields [64, 79, 80]. Gleicher et al. [37] presented a broad survey with over 100 different comparative visualization tools from information visualization domains, organized by their comparative visual designs (see Sect. 6). These surveys focus on the visualization solutions; in contrast, this paper focuses on understanding the comparison problems.

A number of papers consider user performance at comparative tasks using various visualizations. Sometimes these studies focus on comparing methods for specific tasks, such as Livingston et al.'s [53] evaluation of scalar field comparisons or Alper et al.'s [4] evaluation of brain connectivity graphs. Other papers explore design decisions and perceptual abilities for particular designs, such as work on comparing groups in scatterplots [38] or star glyphs [35]. Such evaluations are useful in understanding tradeoffs in specific designs, and perceptual abilities that designs may exploit. This paper takes a top-down view to understand the problems that designs may be trying to address.

Another important source of background is the rich literature on perceptual issues related to comparison. The perceptual and cognitive science communities have considered the problem of visual comparison for decades, see Farell [29] for a historical review. The interpretation of visualizations can be complicated by a variety of phenomena including "change blindness" (see Rensink [66] for a discussion of these phenomena, and Franconeri [32] for a discussion of relevant limitations in the mechanisms of perception). Some findings from perceptual science may have direct impact on the design of comparison methods. For example, translated copies of an object are easy to compare [51], but factors such as texture, orientation, scale, space, and time may complicate comparison [50].

2.2 Task Abstraction

The visualization literature has a significant interest in understanding and categorizing abstract notions of task; see Brehmer and Munzner [13] or Schulz et al. [73] for recent works with historical retrospectives on the task analysis literature. Most task characterizations include comparison within their scope. In surveying task typologies, Brehmer and Munzer [13] list no fewer than 10 other papers that have "compare" in their task catalog (the number may be higher considering synonyms). The breadth of comparison within these surveys varies: in [13] it is deeply nested as a type of "query" which is a type of "motivation." In other taxonomies, such as Kehrer et al. [47] and Wehrend and Lewis [81], compare is a top level broad category. The work of Roth [68] makes comparison a major category of tasks and distinguishes several types of comparison (e.g., comparing within a relation vs. comparing between relations). Andrienko and Andrienko [6] also include comparison as part of their abstraction of task with a broad view that describes comparison between parts of a single data set; what others might call comparisons between data sets would be within a union set. While this breadth is similar to Sect. 3, they do not consider at a high level what one might do with these comparison targets. While the prior work distinguishes comparison from non-comparison, we instead seek to identify common ideas that can be applied across the diversity of comparison scenarios, and possibly to situations that are not obviously comparison.

Many papers provide specific, comparative tasks in discussing the motivations or evaluations for specific designs. These tasks are usually specific to their domain and data type. For example, Alper et al [4] list tasks for brain connectivity analysis and Piringer et al [65] provide tasks for comparing function ensembles. Sect. 3.2 identifies abstract categories in an argument for the diversity of comparative tasks.

[1]Technically, these are spaces, not typologies, because the categories are not mutually exclusive [9, 54].

[2]This definition is consistent across multiple online dictionaries, although they differ in how they refer to the items.

2.3 Surveying Comparison in Visualization

The lack of a crisp classification between "comparison" and "non-comparison" makes a broad and systematic literature survey challenging. Instead, this paper's framework is based on informal explorations of the literature, thought experiments, and experiences of applying these ideas to design visualization solutions. The paper uses representative samples[3] chosen to help make its points. To help assess the completeness of the categorizations of Sections 4-6, a survey of the literature augmented our experience to provide examples. This informal experiment helps confirm that all examples of "comparison solutions" fit into our framework.

A semi-automated literature search helped provide examples for the survey. The survey collected papers from visualization conferences, over the years 2007-2015. It included only major conferences (InfoVis, SciVis, VAST, Pacific Vis and its predecessors, IV, BioVis, etc.), providing a set of 2881 abstracts. Simple text tagging (word spotting) found forms of the word "compare" (e.g., comparison) appear in 354 abstracts, and variants of "similarity" (e.g., similar, differences) in 321. The unsatisfying nature of this simple approach helps expose the diversity of how comparison appears in the literature. The top scoring documents (percentage of comparison words in the abstract) often did describe a visual tool for supporting comparison, where multiple objects are specifically considered. However, this set of documents also included many that described comparisons of visualization methods or conditions in a perceptual experiment. Some papers that describe visual comparison techniques or systems did not show up at all (their abstracts did not contain the specific words). More sophisticated searching may help, but there are two deeper issues. First, the authors may not have written about their approach in terms of comparison, or even explicitly thought about comparison in the design and development of the system. Second, the authors and/or their users may not think of their task in terms of comparison (see Sect. 3.3).

Even an informal scan through the visualization literature provides a wide range of examples of tools and methods that assist users with comparisons. Some solutions are very specific to particular objects from specific domains, such as process plans (i.e., Gantt charts [41]) or molecular surfaces [70], while others consider generic data types such as graphs (e.g., [5, 24, 48, 55]) or scalar fields (e.g., [34, 62]).

3 What are the comparison elements?

The broad definition of "compare" involves two elements: the *targets,* the set of items being compared, and an *action* that is being performed on the relationship among them. The first consideration of comparison is to identify these elements in the analytic task. These elements lead to the challenges that may be addressed by a visualization tool.

Brehmer and Munzner [13, 59] also use targets and actions as elements in their general approach to task abstraction. However, in their typology, comparison is a type of "query" action that has multiple targets; they provide little discussion of what one does with the targets. With our focus on comparison, the action represents what the user wants to do with the *relationship* among the targets, while the targets are a set of things that are to be related. After discussing targets and actions for comparison, this section discusses issues in naming them.

3.1 Targets: what is being compared?

The *targets,* the set of items to be compared, play a central role in comparison. Once the targets are identified, the properties of the set can be used to identify comparison challenges (Sect. 4). General task abstractions (e.g., [13], [73]) often define comparison by the cardinality of this set: a comparison is an analysis with more than one target. However, when considering comparison broadly, there is more subtlety in how the target set is defined.

In some comparisons, the target set is a set of multiple, known and available items. We call this an *explicit* target set, and it is the commonly considered case. In contrast, *implicit* target sets have hidden targets, sometimes appearing to only have one (explicit) target. For example, the user may compare an item to their expectations or memory of an item they have seen previously. Implicit targets can enter into explicit comparisons, for example if multiple explicit targets are also compared against an implicit baseline. We term comparisons where implicit targets are important as *implicit comparisons.*

[3]This paper intentionally over-samples from work the author was involved in as we know how the comparative thinking process was applied.

Implicit comparisons are important, but not well-explored in the literature. The framing of a problem as implicit comparison may be helpful in developing mechanisms for engaging user knowledge in analytic tools. For example, the need to present objects in a similar manner to make them more easy to compare also implies that objects should be presented in ways that match the user's internal targets. Using familiar scaffolds for data can offer one approach. Sarikaya et al. [69] discuss an example where violating this principle led to failure: use of a correlation matrix was rejected by virologists who needed to see genetic sequence data in a familiar form. Similarly, the graph layout literature offers examples of trying to perform layout to match user models (see [7, 8, 33]), while Wood and Dykes [82] describe how to lay out treemaps to match spatial expectations (such as for geographic data). Liu and Stasko [52] explore mental maps more generally.

Explicit comparisons have identifiable target items. However, there may be a gap between what is identifiable and what has been identified. For example, the set of objects to compare may be known by the user, but not by the visualization tool, forcing the user to view the items independently. In other cases, the user may need assistance in identifying which items to compare, or may not even think about their comparison as multiple objects (see Sect. 3.3).

3.2 Actions: what to do with relationships?

In the spirit of Brehmer and Munzner [13], we categorize tasks in terms of the actions that people want to do; however, they (as well as most prior taxonomies, see Sect. 2.2) consider "compare" as a single broad category. This is insufficient as comparison is many more specific actions. In the spirit of Andrienko and Andrienko [6] we define these actions as being about a relationship abstractly, leaving flexibility in terms of what the relation and target set is. As in their work, we consider relations beyond similarities and differences, such as patterns and trends.

Action categories are verbs that describe what a user may want to do with a relationship on the target set. They are chosen to be abstract enough that a small set covers most of the common cases, with enough specificity that they suggest design goals:

- *Identify* the relationships among items.
- *Measure/Quantify/Summarize* those relationships.
- *Dissect* a relationship; that is, to examine the relationship in detail to understand it better.
- *Connect* relationships, for example, to put multiple differences together to assemble a more complete concept, or to understand the variety within a set of items.
- *Contextualize* how a similarity/difference fits into the bigger object of which it is part.
- *Communicate/Illuminate* a relationship (i.e., explain it to others).

This categorization of actions conveys the broad range of comparative tasks. In prior work, consideration of comparison is often limited to the first *identify* action.

Some action categories involve identifying the relationships, while others use ones already known. There is some parallel with Bertin's purposes for visualization (presentation vs. exploration) [11], which was extended by Schulz et al. [73] to *expose, confirm, present.* However, purpose and known relationship are not strictly connected: exploration might involve dissecting known similarities, while a presentation might require the viewer to identify differences important to them. Comparison can occur in all purposes.

3.3 Naming: what do users call the elements?

The framework defines comparison tasks in terms of targets and actions. However, domain users may have alternate ways to describe their tasks. They might consider the set, or the relationship directly. For example, a user may be comparing numbers or examining a series (that is composed of numbers). In biological applications, a user might consider their task to be examining a sequence alignment or browsing synteny, rather than than comparing among different sequences (see [2, 25, 56] for examples of this naming diversity). In scientific computing, users consider the task of exploring an ensemble (e.g., [31,44,49,65]); tasks include comparison between members within ensembles or between different ensembles.

There is a tradeoff between using the terms and metaphors preferred in a domain with making comparison explicit to better understand the tool design challenges. A mismatch in naming can be either a source of confusion, or an opportunity for insight into tool design:

- These namings often imply collection or comparison objects that can be useful in the design of systems. For example, thinking in terms of "understanding alignments" has inspired designs for sequence comparison, and may be valuable in other domains where alignments are common (e.g., registering images).
- There may be clues in the namings as to the user's true comparative intentions. For example, in looking at a scatterplot they might not be as interested in individual points as they are in comparing groups or clusters. In simulation ensembles, users often seek to understand variability or consensus rather than individuals.
- If a named object does not exist, there is an opportunity to introduce new concepts. For example, a system could make the concept of alignment explicit to users comparing small graphs, or explicitly consider groups and clusters.

Another variant of naming that can make the set of comparison objects less explicit are tasks that involve comparing parts or regions of a single object. In some sense, these internal comparison tasks are just a renaming (it is an explicit comparison of the parts). But such tasks often involve finding the set of parts to compare (e.g., [34, 43]), and less clear boundaries between them.

3.4 Lessons

> At the heart of quantitative reasoning is a single question: Compared to what? – E. Tufte [78, p. 68]

Calling a task "comparison" is neither necessary nor sufficient. Comparison may be a range of specific actions. The user may call it "examining a relationship" or may not consider the object set explicitly. Many analytic tasks can be framed in terms of relationships among target objects. Exposing the set of target objects, or the realization that the set is not known or implicit, can help match the user's task to what the system can support. Appreciating that a user's goal may be something beyond identifying a relationship can help focus tool design on user problems. Thinking in terms of comparison, with a target set and action on the relationship among them, enables the considerations of the next sections.

4 WHY IS IT A DIFFICULT COMPARISON?

Not all tasks with comparative elements require a designer to explicitly consider comparison in creating a solution. However, the comparative elements of a task, once identified, can be examined to see if they are likely to be challenging factors for the user, and, therefore, worthy of consideration in tool design. The elements of comparison (target set, relationship, and action) lead to three different ways that the "hardness" of an analysis problem may scale:

1. the *number* of items being compared;
2. the size or complexity of the individual *items;*
3. the size of complexity of the *relationships.*

These three factors form a space for comparison problems. The factors can occur independently: a given problem can be hard in any one axis or along multiple axes. There are often correlations — for example, larger items tend to have more complex relationships.

The three factors abstract the primary ways that comparison problems become challenging. An analysis problem that is low in all of these factors is less likely to benefit from thinking about it as a comparison in order to design a tool to support it. However, even simple comparisons may benefit from creative design — for example, Ortiz [63] discusses comparing two small numbers.

Number of Items Being Compared: Comparison grows difficult as the number of objects being compared increases. Comparing two objects is generally easier than comparing many objects, other factors being equal.

Many comparative visualization tools support comparison between two items (e.g., the original Unix *diff* program, or specialized tools such as TreeJuxtaposer [58], Mizbee [56], the designs in [3], and many others in our literature scan). Comparing two items seems different from comparing even "a few" items — many of the designs used for two-way comparison do not scale to even three. Graham and Kennedy [39] also distinguish designs for comparing two trees from comparing multiple ones. The prevalence of two-way comparison is an open question: Is there less need for multi-way comparison? Is three-way fundamentally harder than two-way?

The set of targets to be compared may be ordered, or represent samples along a continuous axis. For example, temporal comparisons, i.e., multiple comparison targets that are the same object measured at different times, have both of these properties. Challenges in visualizing temporal sets have been explored (for example [17,23]). Properties of the set of items should be considered in designing support for comparison. For example, a set's natural ordering may facilitate comparison because relationships between nearby elements and trends along the ordering are most interesting. Conversely, existence of a natural ordering may preclude reorganizing the data to expose relationships because violating this ordering may cause mismatches with the viewer's mental map.

Size/Complexity of Items Being Compared: Some items are easier to compare than others. Objects typically grow harder to compare as they grow larger or more complex. Simple objects, such as lists or time series, can grow challenging to compare. Complex objects such as weighted graphs [4] can be difficult to compare even when small. Either kind of growth can challenge users, and deserves consideration in tool design.

With large items, the parts that relate may be relatively small. Such scale mismatches cause finding a difference to be like searching for a needle in a haystack. Conversely, larger objects offer the potential to have larger relating parts. Large relationships are not equivalent to complex ones. For example, in comparing two (potentially very large) images, there might be a single pixel that is changed (change is small relative to the object), one image is dimmed in which case the relationship is big (change is the size of the large object), or the change may involve many sub-regions that move (a complex relationship). Multi-scale comparison problems, where the interesting relationships are much smaller than the items being considered, raise common challenges such as finding the small relationships (e.g., differences) and showing them with enough context such that they can be interpreted.

Size/Complexity of Relationships: The size and/or complexity of a relationship between objects is different from the size/complexity of the objects themselves. For example, with two long lists the relationship might be simple (e.g., the elements in each correspond, so comparison can be a simple element by element check), or complex (e.g., the lists can have different orderings and elements without simple one-to-one correspondences).

Larger and more complex items, and larger sets of items, afford more opportunities for more complex relationships. However, simple relationships may be meaningful in complex data sets. For example the YMCA mesh comparison system [72] considers collections (more than three) of large meshes, however, it focuses on examining small, localized changes in meshes that are otherwise very similar.

4.1 Lessons

Once the key challenges of a comparison are identified, a designer can focus on choosing an approach to address them. Similarly, evaluation should focus on the kinds of scalability that matter to the problem: if the designers are able to provide a tool that addresses the real challenges of a problem, sacrificing the ability to scale in some other way should not be a detriment (especially if the decision was explicit and stated).

Few systems tackle all three comparative scale problems at once. There are examples of systems for considering complex relationships among dozens of large objects. However, addressing all three often leads to compromises (e.g., the case study in Sect. 7.1). Successful approaches often limit which challenges they face: considering two (or a few) items, considering only simple items, or only considering simple relationships. While such restrictions narrow the scope of utility, they also can help lead to more successful designs.

5 What is the strategy?

The three challenges of the prior section are all issues of scale: the task has "too much" of something and this taxes the user's perceptual and cognitive limits. Effective comparative designs must address the scale challenges of the task, whether it is challenging in terms of number of items, size/complexity of items, or size/complexity of relationships. Methods to address scale can be categorized broadly by user strategy:

1. *Scan Sequentially:* the user will examine items serially;
2. *Select Subset:* the user will examine a smaller set of items;
3. *Summarize Somehow:* the user will examine an abstraction that concisely describes the items.

Identifying a user strategy can help in creating designs that address scale. Designs should provide affordances that encourage and support the strategy. Each strategy may be effective at addressing any of the challenge types. Some strategies may be more directly appropriate to a particular challenge, and each has common pitfalls.

Scan Sequentially strategies imply a linear examination as an ordered process. They are more directly adapted to challenges involving numerous or large items. However, a design must make scanning efficient to be effective at scale. Ordering is also critical to the success of scanning strategies. By showing the most important items first, good ordering enables the user to stop before the scan is complete. Ordering can also help address complexity by placing related items together.

Select Subset strategies imply that the user will not see all of the data. The strategy most readily applies to challenges involving numerous or large items. At scale, this requires a design that explicitly creates the subset, through selection, filtering, or sampling (see [27] or [12]). Designs can mitigate issues that arise from the reduction, for example to allow the user to retain context of the larger set or be aware of difficulties arising from the incompleteness.

Summarize Somehow strategies build abstractions of the larger set that concisely describe their properties. Such summaries are typically statistical (e.g., parametric models such as means or counting models such as binning). However, other options include visual summaries that provide an overview pre-attentively [75]. To support a summarization strategy, a visualization must consider two aspects: how to create the summary and how to present it.

In comparison, summarization can happen in two orders: first the relationships between objects can be found and these relations are summarized; or first the items are summarized and these simplified items are compared. This distinction is raised in discussions of flow comparison [64, 80]. They pose a trichotomy of comparison "levels:" data, feature, and image. These levels refer to stages in a flow analysis process where different degrees of abstraction have been applied. Data refers to the raw data, feature refers to abstracted data, and image refers to the resulting imagery. This idea extends beyond flow comparison: comparison may be applied at different levels of abstraction. Conversely, abstraction can be applied to to the targets, the relationships, or the visual representation. For example, a set of genetic sequences may be abstracted (e.g., subgroups of sequences abstracted into consensus sequences or simplified by considering genes instead of base pairs) before or after the relationship (alignment) is computed; the alignment itself may be simplified (e.g., using edge bundling), or even a depiction of the alignment may be abstracted (e.g., creating a density map from an edge diagram). Multiple summarizations can happen within a comparison process.

5.1 Lessons

Managing size and complexity is common across analytic applications. Categorizing the strategies enables identifying challenges and cataloging solutions from prior experience. Without support from their tools, users will apply some strategy to manage the complexity in their problem. Building tools that support what users are likely to do, for example, to examine a set of items (or a large item) systematically, or to help users keep track of selected subsets, is a step towards assisting with comparison. Conversely, exploring a different strategy might lead to more effective solutions.

Sometimes, designs inadvertently interfere with strategies. Spatial layouts may help expose patterns and clusters, but also make a systematic scan over the elements more challenging. Edges showing connections between items, for example in a node-link diagram or connecting matches in gene sequences, visually summarize to a blur [2]. Solutions may require either computational approaches (such as edge bundling [40]) or visual designs that support summarization by the perceptual system [2, 76].

Designers of analysis tools, especially those for comparison, must make a myriad of choices in creating a tool. Understanding the scalability challenges in a scenario and making an explicit choice of strategy provides a process that can help create solutions that address key needs.

6 Which visual design?

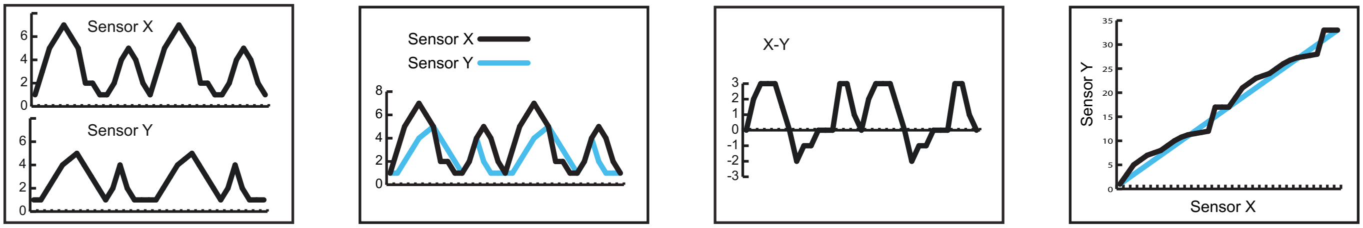
In a prior paper [37], we posited that there are three basic designs for visual comparisons (Fig. 2):

1. *Juxtaposition:* items placed in different spaces (next to each other).
2. *Superposition:* items placed in the same space (on top of each other). This is sometimes called superimposed.
3. *Explicit Encoding:* the relationships are visualized.

These basic designs are sometimes combined. This initial categorization focuses on one aspect of visualization design: it does not consider other key elements such as interaction, or what encodings should be used. Tominski et al. [77] show that the three designs have physical analogs and use these to extend them from visual designs to interaction techniques that provide control over different aspects of these layouts. They also introduce a spectrum of the sub-tasks common across many (if not all) comparisons.

Javed and Elmqvist [45] consider the ways to compose two views (for uses beyond comparison), and point out two strategies beyond juxtaposition and superposition (although they call it super-imposition): overloading and nesting. These two additional combination strategies can be applied for comparison and for multi-way combinations. For example, the VAICO image comparison system [71] insets small pieces from other images inside the primary image to show differences and would be considered a nested design in their categorization. Within the original three-way categorization, these new categories would be considered superposition as they show the objects to be compared in the same space. What these new categories suggest is a design space of different ways to combine objects in a single view to realize superposition.

The range of designs for combining multiple pieces of information in the same space is termed "Visual Multiplexing" by [15], who give an extensive categorization of designs and exploration of the design space. Their work is more general than comparison as they consider a wide range of situations where visual elements are shown in the same space. Their categorization is organized by the mechanisms used by the viewer to de-multiplex (i.e., pull apart) the different signals. While the multiplexing concept extends beyond superposition for comparison, it provides a diverse set of suggestions for superposition designs,



(a) Juxtaposition (b) Superposition (c) Explicit Encoding: Difference (d) Exp. Encoding: Time Warp

Fig. 2. The three visual designs from [37] shown on a simple example: comparing two time series. The three basic approaches for comparative visualization are (a) juxtaposition, (b) superposition, (c,d) explicit encoding of relationships.

and provides a framework for assessing and choosing amongst superposition designs. These strategies may provide designs that can address some of the issues in superposed designs. However, their model of multiplexing may not be a perfect fit for all comparison tasks as it focuses on the ability of the viewer to pick out the original information that has been combined. In contrast, for some comparison tasks the mixing of the signals is valuable as it helps the viewer see a combined object that can expose the relationship among the original ones.

6.1 Comparative Spaces and References

Layout of the individual items is important to all comparative designs. For superposition designs, this is critical as (by definition) the items being compared must be in the same space. For juxtaposition designs, having similar layouts can reduce the burden on the viewer. For explicit encodings, the space may or may not relate to the space of the original objects. In cases where the items are not naturally in the same space (that is, they need to either be given a spatialization or need to have their spaces aligned), something must choose the space. For example, in graph visualizations, even the same graph may have very different visual representations (see [61] for an example that exploits the diversity of possible layouts). Control over layout can have broad applicability in steering the kinds of questions a visualization can answer [74], so other concerns may complete with comparative needs.

In some designs, a particular item of the set being compared is chosen as a *reference.* All other items are shown relative to this reference, for example all are aligned to the reference, shown in the coordinate system defined by the reference, or shown with the reference superimposed (e.g., [26, 47]). Many systems allow for the user to select a reference object to which other items are aligned (e.g., [2, 18, 19], Fig. 4 and Fig. 5). Kehrer et al. [47] introduced the idea of multiple references which offers a powerful extension. In some situations *reference-free* designs are attractive because they do not emphasize one item over another.

6.2 Other Aspects of Design

The three-axis space focuses on a particular aspect of visualization design: the layout of the presentation. Careful design choices are required to make any of the three design strategies work. For example, careful choices of visual encodings are necessary to help viewers build connections and see differences between juxtaposed views, to merge information so it can be later separated in superposed views, or to present complex explicit encodings. The understanding of the perception of comparison (Sect. 2.1) and a growing understanding of perceptual abilities for summarization (see Szafir et al. [75] for a survey) can inform visualization design.

Interaction is often a mechanism for addressing scalability concerns in visualization. It applies across the space of comparative designs. Interaction can be used as a mechanism for addressing issues with a design strategy, for example, using brushing and linking to help establish connections between juxtaposed views or using focus+context or detail on demand techniques to reduce clutter in superposed views.

The process implied by our design strategy consideration is to choose the strategy first, and then to choose other design aspects (e.g., encoding and interaction) as details to address issues in applying these strategies. It seems more natural to fit these design aspects to a strategy, rather than vice-versa. However, choosing a good strategy is of little use if it cannot be realized in an effective manner. There is an emerging catalog of specific examples of layouts, encodings, and interactions for comparisons, finding ways to abstract these components is important future work.

6.3 Lessons

The visual comparison design space can be a useful tool in design. If a choice is not working, a designer can either experiment by making a different choice, or use the prior knowledge about design type to look for ideas. Organization of design ideas by layout strategy offers a way to find solutions from disjoint places. For example, looking at juxtaposition designs can offer ideas including interaction techniques (e.g., [60]) and visual design (e.g., [47, 78]).

The choice of design strategy relates closely to the comparison challenges. For example, neither superposition or juxtaposition designs naturally scale to many items. Superposition becomes cluttered with many objects in the same space. Juxtaposition with many items separates them, hindering some kinds of comparison. Either approach requires some strategy for managing the complexity, although juxtaposition may naturally support scanning.

One way to develop novel solutions is to challenge the common approach for a specific problem. Challenging prevailing wisdom requires finding ways to preserve the desirable elements of the prevailing approach. For example, Dasgupta et al. [26] noted that the common "spaghetti plot" for comparing time series does not scale to larger numbers of series, so they provided a new design that is primarily juxtaposition, but with superposition elements (including a reference) to aid in comparison. Similarly, Sequence Surveyor (Sect. 7.1, Fig. 3, [2]) uses juxtaposition, rather than the common explicit encoding, to address scalability concerns of existing approaches.

There have been experiments that compare specific designs for specific applications (e.g., [55] for matrices, [7] for small graphs, and [46] for time series). However, more general guidelines for choosing among the three design types, and developing effective combinations, have been elusive. Explicit encodings require the relationships to be known (so that they can be encoded), and often remove relationships from their contexts within objects. Superposition designs require the items to be similar enough that they can be shown in the same space, and require careful visual design to provide for scaling along any challenge axis. Juxtaposition places much of the burden of working with relationships on the viewer, which creates issues with scaling.

7 Case Studies

Case studies demonstrate the utility of the comparative considerations. In each, the sequence of considerations identified challenges, which enabled the creation of an appropriate design.

7.1 Sequence Surveyor

Our work in sequence comparison visualization, for both biological and text applications, provide clear examples of the considerations of comparison. Sequence Surveyor [2] illustrates the four considerations.

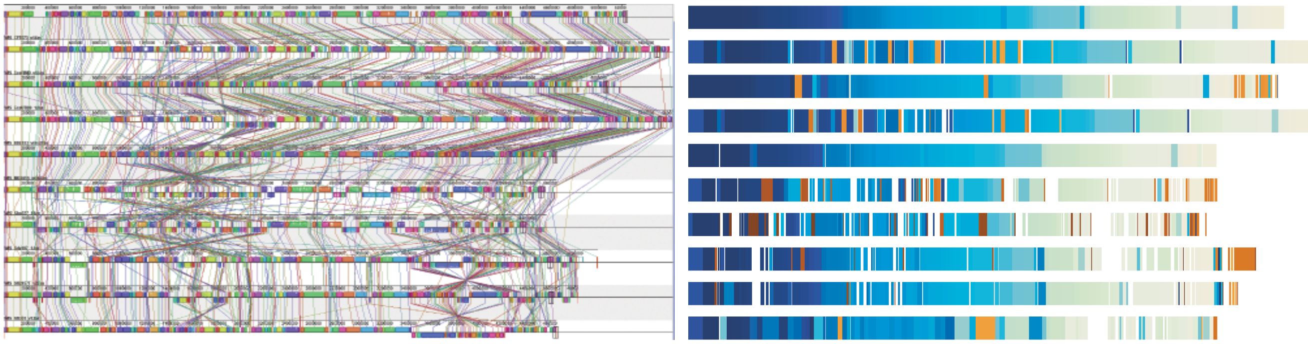


Fig. 3. Mauve [25] (left) and Sequence Surveyor [2] (right) displaying ten *E. coli* and *Shigella* genomes. Mauve uses the conventional explicit encoding showing connections between aligned blocks, while Sequence Surveyor uses a configurable colorfield that makes important relationships salient. Different color and ordering configurations expose different relationships.

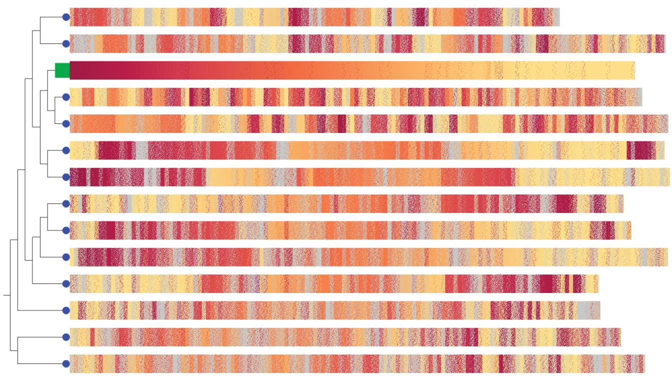


Fig. 4. Sequence Surveyor [2] showing ten *E. coli* and *Shigella* genomes, the one marked with a green square is set to be the reference. All other genomes are aligned to this row, allowing for similarities and differences to be quickly assessed.

The TextDNA System [76] is an evolution of Sequence Surveyor designed to address its issues in applying it to a different domain. LayerCake [18, 19] approaches a seemingly similar problem to Sequence Surveyor, but the considerations of comparison highlight the differences in the challenges and helped guide a different solution.

Sequence Surveyor was developed concurrently with the framework of this paper. It was designed to help genetics and evolution researchers make comparisons amongst multiple bacterial-sized genomes. The *targets* of the comparisons were explicit, the set of gene sequences that had been aligned. However, the domain collaborators refered to their task as "exploring a multiple sequence alignment." This naming was an important reminder that the alignment between genomes was an explicit object in their thinking. For example, a key task (and most successful use case for Sequence Surveyor) was to evaluate alignments for debugging. The comparative *actions* were less clear: such large and diverse collections of sequences were unprecedented, so the biologists were unsure of what they would be looking for. This lack of clarity in actions to support led to designs that emphasized flexibility, at the cost of ease of use.

The comparative *challenges* were easy to identify given the elements. The application demanded addressing all three challenge types: *number,* as the biologists were assembling collections of dozens of genomes to compare; *size,* as each genome was large, and *complexity,* as the alignments included significant re-arrangements, replacements, replications, and other complex patterns. No existing tool could address all of these dimensions of scale simultaneously. For example, Mizbee [56] and Combo [28] compared pairs of genomes, MAUVE [25] scaled to a handful of smaller genomes, and virology tools (discussed below) handled small genomes with simple relationships.

Sequence Surveyor chose *strategies* to address each of the scaling dimensions. To scale to longer sequences, it used summarization (specifically visual summarization). To scale to more sequences, it facilitated scanning through the list by providing a compact visual design that put many sequences on screen together in an ordered fashion and an interface for reordering that further enhances the utility of sequential reading. To combat relationship complexity, Sequence Surveyor provided different mechanisms that reordered the genes within the sequences to expose patterns, and for choosing the alignment reference interactively (Fig. 4).

For its *visual design,* Sequence Surveyor chose a juxtaposition design, going against the convention in sequence alignment visualization tools of using an explicit encoding to show connections between aligned sequences (Fig. 3). The design used a dense colorfield, with each sequence as a row. The juxtaposition design relies on the viewer's ability to make the connections to find patterns; however, the scalability strategies provided ways to make this work at scale. The reconfigurability of the display can cause meaningful (but complex) patterns to become visible.

In attempting to address all three types of comparative scalability, as well as to provide for "scalability in task," Sequence Surveyor made a tradeoff. Flexibility and reconfigurability allow the tool to expose many different types of relationships across many different scales. However this flexibility comes at the cost of usability: Sequence Surveyor provides a vast number of options, a user must somehow choose which configurations meet their needs and interpret the resulting display. The later TextDNA system [76], adapted the Sequence Surveyor approach to document collection exploration (i.e., comparing texts and groups of texts) but attempted to address the complexity of use issue in order to appeal to a broader audience.

7.2 LayerCake

The LayerCake system [18, 19] (Fig. 5) had a seemingly similar comparison task to Sequence Surveyor: compare a number of genetic sequences. However, the considerations of comparisons expose that the specific domain (virology) has much different needs, leading to a different solution.

The comparison *targets* in the LayerCake problem were explicit, there was a set of viral genomes, each a mutation of the basic virus (called a variant). Unlike in Sequence Surveyor, this set is relatively homogeneous, with only small differences between members, alignment is simple as there are no rearrangements. However, the basic virus is well known to the virologist users who spend years studying specific viruses. Thus, there was a non-trivial element of *implicit* comparison, as tool users needed to relate their observations to their prior knowledge of the virus' structure. The comparison *actions* were well defined. While identifying the sites of mutations could be automated (by differencing with a reference), two important tasks emerged: the differences had to be dissected to understand which were significant;

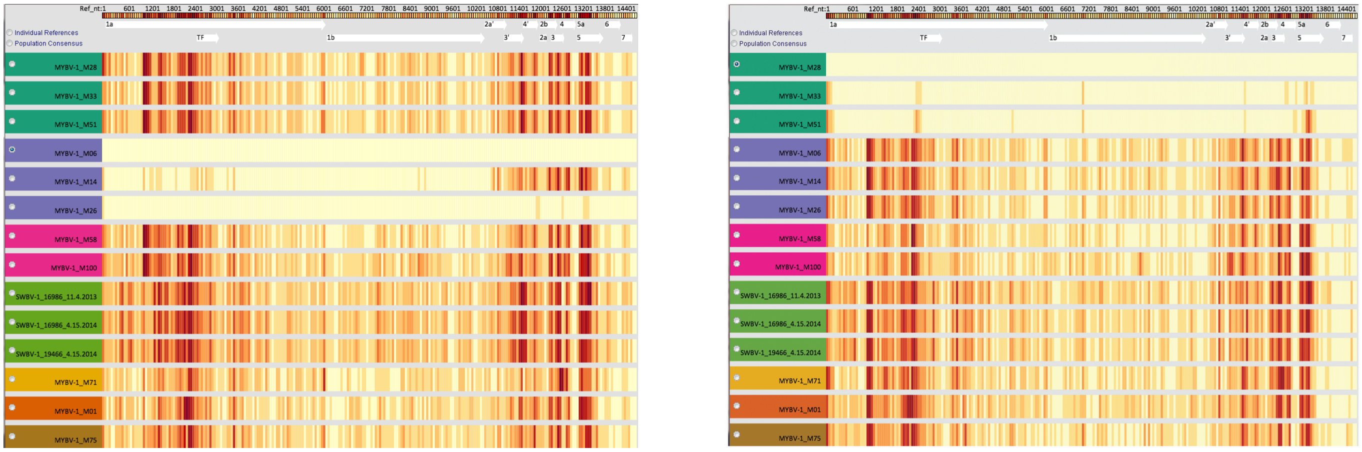


Fig. 5. The LayerCake system [18, 19] visualizing viral populations from 14 individuals. Two different reference selections are shown. Each row represents the mutation counts for each position along the virus' genome within an individual animal. Sequences are shown in reading order. A color encoding displays the counts in a dense manner. To scale to the viral lengths (size of items), the design uses a summarization strategy (binning) and provides focus+context tools to support close inspection required for key tasks (e.g., inspection and dissection). To scale to larger numbers of individuals, a juxtaposition design is used that supports re-ordering and the selection of any individual (or consensus of set of individuals) as a reference. The choice of different references allows for judgments even at a distance (different references are chosen in the left and right images). The juxtaposition orderings can place related individuals close together, allowing for comparison across groups (colored labels).

and the differences had to be connected to see which sets of mutations occurred commonly across different groups of sequences. The latter requires LayerCake to support comparison among groups of sequences, not just individual sequences. This meant that comparison also needed to consider targets that were groups of genomes.

LayerCake offered comparative *challenges* in *number* and *size* of the items to be compared. Virologists needed to compare dozens of variants. While viral genomes are relatively short (relative to the organisms considered in Sequence Surveyor), they need to be examined at the level of individual base pairs, meaning the sequences to be considered were far larger than the number of pixels (even on a massive display), yet individual element differences could be significant.

The *strategies* chosen in LayerCake were to summarize to address the size of the sequence, and to support scanning to address the number of sequences. The *visual design* is similar to Sequence Surveyor, a juxtaposition design where each genome is represented by a color band. However, the problem led to very different details. Because the virtologists needed to make implicit comparisons with their understanding, genomes must be presented their standard order. Re-ordering to expose patterns was not acceptable. Grouping and re-arranging the different variants allowed a virologist to expose connectiosn where variants had similar mutation patterns. Later versions of LayerCake [18] integrated more automated tools for filtering significant mutation sites, so that the viewer could more rapidly compare among sets of sequences. LayerCake makes an important tradeoff: LayerCake is highly specialized to its application, but this allowed it to provide a more effective solution by focusing only on the challenges of the application.

7.3 Topic Model Comparison

Our work on Task-Driven Topic Model Comparison [3] emphasizes the value of identifying the elements of comparison. The broad problem of comparing complex statistical models seemed daunting. However by considering specific comaprison tasks, each with specific *targets* and *actions*, we could design effective solutions. The *targets* were never the entire topic model, instead, examining the tasks revealed that users wanted to make comparisons between components of the models (such as word lists) or outcomes of the model (e.g., near-neighbors of documents).

Each specific topic modeling task had different targets and actions, leading to different *challenges*. In all cases, the tasks has similar *number,* i.e., comparing a pair of topic models. However, the *size* of the items varied: lists of topics were small, lists of words and documents were potentially long. Depending on the challenges of a specific task, an appropriate *strategy* and *visual design* were chosen. Three task specific designs are illustrated in Fig. 6. The three designs include all *strategies:* subset selection is applied to show only the strongest matches in the topic matching view (Fig. 6a), scanning is enabled by sorting the word lists in order (Fig. 6b), and (visual) summarization is employed to show document lists in (Fig. 6c).

8 Discussion

The four considerations for comparison, in sequence, provide a process by which to develop support for comparison within the development of a visualization solution. They each fit in with, and augment, a phase of the visualization design process[4]:

1. As part of *task/requirements analysis,* identify the comparative elements in the problem, the set of targets and an action on the relationship among them. Merely labeling the task as comparison or not (as most task taxonomies suggest) is neither necessary nor sufficient. Identifying the targets and actions allows consideration of user needs as well as enabling the subsequent steps that expose potential issues and help match them with solutions.

2. As part of *abstracting the data,* consider how the comparison creates scalability challenges that must be addressed in a successful solution. Comparative scale challenges — number of items, size of items, complexity of relationship — provide broad categories to look for in analysis and address with designs.

3. As part of *selecting a design strategy,* consider how the user will address the exposed scale challenges with the system. The three abstract categories of scalability strategies — scan, subset, summarize — can focus the choice of solution, suggest issues that must be considered, and help match the design to the users actions.

4. As part of *creating the specific visual design,* consider how to choose between, and potentially combine, the strategies for visual comparison designs. The three abstract categories — juxtaposition, superposition, explicit encoding — each have benefits and drawbacks in how they match with different comparative challenges, scalability strategies, and data characteristics.

[4] My visualization class terms this *Task, Data, Design, Details*, but the names from Munzner's nested model [58] also apply.

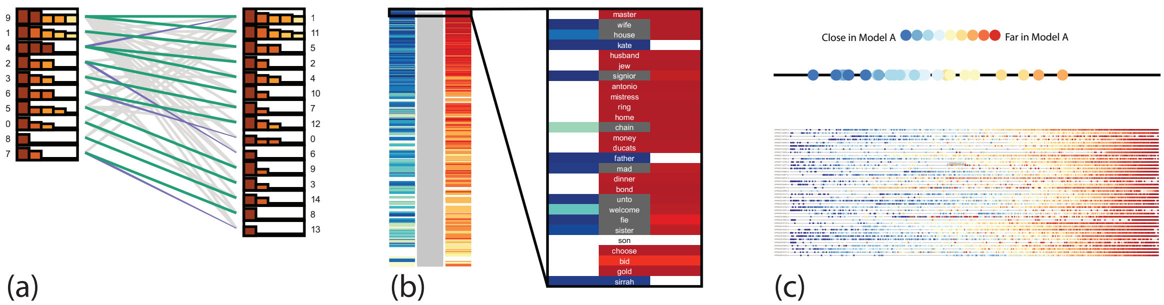


Fig. 6. Three of the visual comparison designs used for topic model comparison. Topic model comparison [3] addresses a number of analytic problems, each with different comparison targets and tasks providing different challenges. In turn, these were addressed using different strategies and designs. **(a)** To *identify* similar topics, algorithmic alignment is presented with an explicit encoding that shows only the most important matches (subset). **(b)** To *dissect* an identified pair of topics, the word lists for the topics are shown juxtaposed allowing the viewer to scan the lists to see if the differences are semantic. **(c)** To convey the effects of the differences over the model, a *buddy plot* uses visual summarization in a juxtaposition design to convey how different document distances are correlated between models. Each row represents the distance to a particular document, every other document is encoded as a circle with the distance in model B in the X coordinate, and the distance in model A in the color. A blue to red ramp along the row would indicate perfect correspondence.

The considerations offer broad groupings of the space of problems and solutions. In particular, the last three considerations offer a particularly concise set of axes as the "three threes." They seem to cover the range of comparison problems seen in the literature, yet provide a small enough set of categories that significant variation is connected in meaningful ways.

8.1 Integration of Visualization and Computation

This paper focused on visualization solutions for comparison problems. However, only the fourth consideration explicitly mentions the visual design. The framework may have utility in designing non-visual analytic tools. The four considerations also help in understanding how visualization may fit into the analytic arsenal for comparison and suggesting how visual and non-visual tools may be combined.

Computational and statistical analysis have important advantages for comparison. Statistics provides excellent tools for measuring and quantifying differences, and understanding whether a difference is significant. Computational approaches excel at finding specific things in large sets (including differences, providing the targets can be modeled). Descriptive statistics provides a robust and rigorous approach to summarizing data, if it falls into a form that is readily characterized. Approaches can be designed to be (statistically) unbiased (although unintended bias effects can still occur [30]), while biases in perception and cognition are harder to design around. However, these advantages have corresponding deficiencies, where visual approaches can be more desirable. Computational approaches require modeling what kinds of relationships are being sought; more complex relationships can be hard to model. Computational approaches require explicit modeling of invariances to denote what kinds of variation should be ignored, otherwise insignificant things can lead to large differences (e.g., noise or off-by-one errors). Computational approaches can give concise answers, but these answers can be difficult to contextualize with the broader data set.

However, computational and visual comparison approaches need not compete: the two methods can be used together. Hybrid approaches bring the benefits of each. There are several broad categories of ways that visual and computational approaches can fit together:

1. *Using analysis to drive visualization:* Computational methods can address comparison challenges. For example, explicit encodings require analysis to find the relationships to encode and most summarization techniques involve computational foundations. Computationally determined differences can drive navigation allowing the user to "tour" differentiated sites in order to help structure their sequential scanning. Analysis can factor out uninteresting variation, for example by aligning sequences or images. This saves the viewer from having to ignore unimportant differences and can enable superposition.

2. *Visualizing analysis results:* Analytic comparisons often benefit from visual presentation of their results. Even simple statistical comparisons (e.g. T-tests) can benefit from visual presentation [20, 22]. More complex analyses can be made easier to understand through a visual presentation. Visualization offers the potential for helping throughout the modeling process [36]. Visualizations can help contextualize analytic results by showing them in more familiar form and communicating the results by presenting them in a more broadly understandable form.

3. *Using visualization and interaction to control analysis:* Analytic techniques require some specification of what differences to look for. Visualization and interaction can address this issue. For example, interactive visualization approaches (such as [10, 14, 42]) build distance metrics for analytically measuring differences between items. There are many ways in which user control over automated analysis can be aided by visual an interative approaches, see Muhlbacher et al. [57] for a categorization.

8.2 Conclusion: Why Comparison?

This paper has avoided making a distinction with what is *not* comparison. Instead, we suggest considering the degree to which comparative elements cause challenges that are worthy of consideration. Tasks that are obvious and explicit comparison may not offer challenges worthy of consideration; conversely problems that are not obvious comparisons may be considered in terms of the comparative elements so that the considerations apply. Comparison is a lens to assess data analysis problems to inform and analyze the design of solutions.

Thinking in terms of comparison with these four considerations has helped us more effectively design solutions for comparison problems.

Acknowledgments

I am grateful to the collaborators who helped me develop these ideas over the years. In particular, my (now former) students including Eric Alexander, Michael Correll, Alper Sarikaya and Danielle Szafir listened to the process and used it in building systems. Steve Franconeri and Chuck Hansen helped me build a research project around visualizing comparison. Many people have suffered through drafts of this paper, and provided constructive criticism. This work was funded in part by NSF award 1162037. The domain applications were funded in part by NIH award 5R01AI077376-07 and an Andrew Mellon Foundation award.

Author's Pre-Print Version
Accepted, to appear in Proc. InfoVis 2017
IEEE TVCG 24(1), Jan 2018

References

[1] J. Ahrens, K. Heitmann, M. Petersen, J. Woodring, S. Williams, P. Fasel, C. Ahrens, Chung-Hsing Hsu, and B. Geveci. Verifying Scientific Simulations via Comparative and Quantitative Visualization. *IEEE Computer Graphics and Applications*, 30(6):16–28, nov 2010. doi: 10.1109/MCG.2010.100

[2] D. Albers, C. Dewey, and M. Gleicher. Sequence Surveyor: Leveraging Overview for Scalable Genomic Alignment Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2392–2401, dec 2011. doi: 10.1109/TVCG.2011.232

[3] E. Alexander and M. Gleicher. Task-Driven Comparison of Topic Models. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):320–329, jan 2016. doi: 10.1109/TVCG.2015.2467618

[4] 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*, p. 483. ACM Press, New York, New York, USA, apr 2013. doi: 10.1145/2470654.2470724

[5] K. Andrews, M. Wohlfahrt, and G. Wurzinger. Visual Graph Comparison. In *2009 13th International Conference Information Visualisation*, pp. 62–67. IEEE, jul 2009. doi: 10.1109/IV.2009.108

[6] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data*. Springer-Verlag, Berlin/Heidelberg, 2006.

[7] D. Archambault, H. Purchase, and B. Pinaud. Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, apr 2011. doi: 10.1109/TVCG.2010.78

[8] D. Archambault and H. C. Purchase. The "Map" in the mental map: Experimental results in dynamic graph drawing. *International Journal of Human-Computer Studies*, 71(11):1044–1055, nov 2013. doi: 10.1016/j.ijhcs.2013.08.004

[9] K. Bailey. *Typologies and Taxonomies: An Introduction to Classification Techniques*. SAGE Publications, Thousand Oaks, CA, 1994.

[10] J. Bernard, M. Hutter, D. Sessler, T. Schreck, M. Behrisch, and J. Kohlhamme. Towards a user-defined visual-interactive definition of similarity functions for mixed data. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 227–228. IEEE, oct 2014. doi: 10.1109/VAST.2014.7042503

[11] J. Bertin. *Semiology of Graphics*. ESRI Press, Redlands, CA, 2nd ed., 2010.

[12] E. Bertini and G. Santucci. Give Chance a Chance: Modeling Density to Enhance Scatter Plot Quality through Random Data Sampling. *Information Visualization*, 5(2):95–110, jun 2006. doi: 10.1057/palgrave.ivs.9500122

[13] M. Brehmer and T. Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, dec 2013. doi: 10.1109/TVCG.2013.124

[14] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 83–92. IEEE, oct 2012. doi: 10.1109/VAST.2012.6400486

[15] M. Chen, S. Walton, K. Berger, J. Thiyagalingam, B. Duffy, H. Fang, C. Holloway, and A. E. Trefethen. Visual Multiplexing. *Computer Graphics Forum*, 33(3):241–250, jun 2014. doi: 10.1111/cgf.12380

[16] W. S. Cleveland and R. McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531, sep 1984. doi: 10.2307/2288400

[17] D. Coffey, F. Korsakov, M. Ewert, H. Hagh-Shenas, L. Thorson, A. Ellingson, D. Nuckley, and D. Keefe. Visualizing Motion Data in Virtual Reality: Understanding the Roles of Animation, Interaction, and Static Presentation. *Computer Graphics Forum*, 31(3pt3):1215–1224, jun 2012. doi: 10.1111/j.1467-8659.2012.03114.x

[18] M. Correll, A. L. Bailey, A. Sarikaya, D. H. O'Connor, and M. Gleicher. LayerCake: a tool for the visual comparison of viral deep sequencing data. *Bioinformatics*, 31(21):3522–3528, nov 2015. doi: 10.1093/bioinformatics/btv407

[19] M. Correll, S. Ghosh, D. O'Connor, and M. Gleicher. Visualizing virus population variability from next generation sequencing data. In *2011 IEEE Symposium on Biological Data Visualization (BioVis).*, pp. 135–142. IEEE, oct 2011. doi: 10.1109/BioVis.2011.6094058

[20] M. Correll and M. Gleicher. Error Bars Considered Harmful: Exploring Alternate Encodings for Mean and Error. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2142–2151, dec 2014. doi: 10.1109/TVCG.2014.2346298

[21] P. Costigan-Eaves and M. Macdonald-Ross. William Playfair (1759-1823). *Statistical Science*, 5(3):318–326, 1990.

[22] G. Cumming and S. Finch. Inference by Eye: Confidence Intervals and How to Read Pictures of Data. *American Psychologist*, 60(2):170–180, 2005. doi: 10.1037/0003-066X.60.2.170

[23] J. E. Cutting. Representing Motion in a Static Image: Constraints and Parallels in Art, Science, and Popular Culture. *Perception*, 31(10):1165–1193, oct 2002. doi: 10.1068/p3318

[24] W. Czech and D. A. Yuen. Efficient Graph Comparison and Visualization Using GPU. In *2011 14th IEEE International Conference on Computational Science and Engineering*, pp. 561–566. IEEE, aug 2011. doi: 10.1109/CSE.2011.100

[25] A. C. Darling. Mauve: Multiple Alignment of Conserved Genomic Sequence With Rearrangements. *Genome Research*, 14(7):1394–1403, jun 2004. doi: 10.1101/gr.2289704

[26] A. Dasgupta, J. Poco, Y. Wei, R. Cook, E. Bertini, and C. T. Silva. Bridging Theory with Practice: An Exploratory Study of Visualization Use and Design for Climate Model Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):996–1014, sep 2015. doi: 10.1109/TVCG.2015.2413774

[27] G. Ellis and A. Dix. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, nov 2007. doi: 10.1109/TVCG.2007.70535

[28] R. Engels, T. Yu, C. Burge, J. P. Mesirov, D. DeCaprio, and J. E. Galagan. Combo: a whole genome comparative browser. *Bioinformatics*, 22(14):1782–1783, jul 2006. doi: 10.1093/bioinformatics/btl193

[29] B. Farell. "Same" — "different" judgments: A review of current controversies in perceptual comparisons. *Psychological Bulletin*, 98(3):419–456, nov 1985. doi: 10.1037/0033-2909.98.3.419

[30] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pp. 259–268. ACM Press, New York, New York, USA, dec 2015. doi: 10.1145/2783258.2783311

[31] F. Ferstl, K. Burger, and R. Westermann. Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, jan 2016. doi: 10.1109/TVCG.2015.2467204

[32] S. L. Franconeri. The Nature and Status of Visual Resources. In D. Reisberg, ed., *The Oxford Handbook of Cognitive Psychology*, pp. 1–16. Oxford University Press, mar 2013. doi: 10.1093/oxfordhb/9780195376746.013.0010

[33] M. Freire and P. Rodríguez. Preserving the mental map in interactive graph interfaces. In *Proceedings of the working conference on Advanced visual interfaces - AVI '06*, p. 270. ACM Press, New York, New York, USA, may 2006. doi: 10.1145/1133265.1133319

[34] S. Frey, F. Sadlo, and T. Ertl. Visualization of Temporal Similarity in Field Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2023–2032, dec 2012. doi: 10.1109/TVCG.2012.284

[35] J. Fuchs, P. Isenberg, A. Bezerianos, F. Fischer, and E. Bertini. The Influence of Contour on Similarity Perception of Star Glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2251–2260, dec 2014. doi: 10.1109/TVCG.2014.2346426

[36] M. Gleicher. A Framework for Considering Comprehensibility in Modeling. *Big Data*, 4(2):75–88, jun 2016. doi: 10.1089/big.2016.0007

[37] 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, oct 2011. doi: 10.1177/1473871611416549

[38] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of Average Value in Multiclass Scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2316–2325, dec 2013. doi: 10.1109/TVCG.2013.183

[39] M. Graham and J. Kennedy. A Survey of Multiple Tree Visualisation. *Information Visualization*, 9(4):235–252, dec 2010. doi: 10.1057/ivs.2009.29

[40] D. Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, sep 2006. doi: 10.1109/TVCG.2006.147

[41] D. Huang, M. Tory, S. Staub-French, and R. Pottinger. Visualization Techniques for Schedule Comparison. *Computer Graphics Forum*,

28(3):951–958, jun 2009. doi: 10.1111/j.1467-8659.2009.01441.x

[42] K. G. Jamieson, L. Jain, C. Fernandez, N. J. Glattard, and R. Nowak. NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning. In *Advances in Neural Information Processing Systems*, pp. 2638–2646, 2015.

[43] H. Janicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield Visualization using Local Statistical Complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, nov 2007. doi: 10.1109/TVCG.2007.70615

[44] M. Jarema, I. Demir, J. Kehrer, and R. Westermann. Comparative visual analysis of vector field ensembles. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 81–88. IEEE, oct 2015. doi: 10.1109/VAST.2015.7347634

[45] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *2012 IEEE Pacific Visualization Symposium*, pp. 1–8. IEEE, feb 2012. doi: 10.1109/PacificVis.2012.6183556

[46] W. Javed, B. McDonnel, and N. Elmqvist. Graphical Perception of Multiple Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, nov 2010. doi: 10.1109/TVCG.2010.162

[47] J. Kehrer, H. Piringer, W. Berger, and M. E. Groller. A Model for Structure-Based Comparison of Many Categories in Small-Multiple Displays. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2287–2296, dec 2013. doi: 10.1109/TVCG.2013.122

[48] D. Koop, J. Freire, and C. T. Silva. Visual summaries for graph collections. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 57–64. IEEE, feb 2013. doi: 10.1109/PacificVis.2013.6596128

[49] P. Köthur, C. Witt, M. Sips, N. Marwan, S. Schinkel, and D. Dransch. Visual Analytics for Correlation-Based Comparison of Time Series Ensembles. *Computer Graphics Forum*, 34(3):411–420, jun 2015. doi: 10.1111/cgf.12653

[50] A. Larsen and C. Bundesen. Size scaling in visual pattern recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 4(1):1–20, feb 1978.

[51] A. Larsen and C. Bundesen. Effects of spatial separation in visual pattern matching: evidence on the role of mental translation. *Journal of experimental psychology. Human perception and performance*, 24(3):719–31, jun 1998.

[52] Z. Liu and J. T. Stasko. Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, nov 2010. doi: 10.1109/TVCG.2010.177

[53] M. A. Livingston and J. W. Decker. Evaluation of Trend Localization with Multi-Variate Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2053–2062, dec 2011. doi: 10.1109/TVCG.2011.194

[54] A. Marradi. Classification, typology, taxonomy. *Quality and Quantity*, 24(2), may 1990. doi: 10.1007/BF00209548

[55] A. G. Melville, M. Graham, and J. B. Kennedy. Combined vs. Separate Views in Matrix-based Graph Analysis and Comparison. In *2011 15th International Conference on Information Visualisation*, pp. 53–58. IEEE, jul 2011. doi: 10.1109/IV.2011.49

[56] M. Meyer, T. Munzner, and H. Pfister. MizBee: A Multiscale Synteny Browser. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):897–904, nov 2009. doi: 10.1109/TVCG.2009.167

[57] T. Muhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, dec 2014. doi: 10.1109/TVCG.2014.2346578

[58] T. Munzner. A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, nov 2009. doi: 10.1109/TVCG.2009.111

[59] T. Munzner. *Visualization Analysis and Design*. CRC Press, Boca Raton, FL, 2014.

[60] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453, jul 2003. doi: 10.1145/882262.882291

[61] M. Nadal and G. Melanon. One Graph, Multiple Drawings. In *2013 17th International Conference on Information Visualisation*, pp. 416–421. IEEE, jul 2013. doi: 10.1109/IV.2013.55

[62] S. Nagaraj, V. Natarajan, and R. S. Nanjundiah. A Gradient-Based Comparison Measure for Visual analysis of Multifield Data. *Computer Graphics Forum*, 30(3):1101–1110, jun 2011. doi: 10.1111/j.1467-8659.2011.01959.x

[63] S. Ortiz. 45 ways to communicate two quantities. Blog posting on visual.ly, 2012. https://visual.ly/blog/45-ways-to-communicate-two-quantities/.

[64] H. Pagendarm and F. Post. Comparative Visualization - Approaches and Examples. In *Visualization in Scientific Computing*. Springer, Wien, 1995.

[65] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. Comparative Visual Analysis of 2D Function Ensembles. *Computer Graphics Forum*, 31(3pt3):1195–1204, jun 2012. doi: 10.1111/j.1467-8659.2012.03112.x

[66] R. A. Rensink. Change Detection. *Annual Review of Psychology*, 53(1):245–277, feb 2002. doi: 10.1146/annurev.psych.53.100901.135125

[67] J. C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pp. 61–71. IEEE, jul 2007. doi: 10.1109/CMV.2007.20

[68] S. F. Roth and J. Mattis. Data characterization for intelligent graphics presentation. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, pp. 193–200. ACM Press, New York, New York, USA, mar 1990. doi: 10.1145/97243.97273

[69] A. Sarikaya, M. Correli, J. M. Dinis, D. H. O'Connor, and M. Gleicher. Visualizing Co-occurrence of Events in Populations of Viral Genome Sequences. *Computer Graphics Forum*, 35(3):151–160, jun 2016. doi: 10.1111/cgf.12891

[70] K. Scharnowski, M. Krone, G. Reina, T. Kulschewski, J. Pleiss, and T. Ertl. Comparative Visualization of Molecular Surfaces Using Deformable Models. *Computer Graphics Forum*, 33(3):191–200, jun 2014. doi: 10.1111/cgf.12375

[71] J. Schmidt, M. E. Gröller, and S. Bruckner. VAICo: visual analysis for image comparison. *IEEE transactions on visualization and computer graphics*, 19(12):2090–9, dec 2013. doi: 10.1109/TVCG.2013.213

[72] J. Schmidt, R. Preiner, T. Auzinger, M. Wimmer, M. E. Groller, and S. Bruckner. YMCA — Your mesh comparison application. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 153–162. IEEE, oct 2014. doi: 10.1109/VAST.2014.7042491

[73] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A Design Space of Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2366–2375, dec 2013. doi: 10.1109/TVCG.2013.120

[74] A. Slingsby, J. Dykes, and J. Wood. Configuring Hierarchical Layouts to Address Research Questions. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):977–984, nov 2009. doi: 10.1109/TVCG.2009.128

[75] D. A. Szafir, S. Haroz, M. Gleicher, and S. Franconeri. Four types of ensemble coding in data visualizations. *Journal of Vision*, 16(5):11, mar 2016. doi: 10.1167/16.5.11

[76] D. A. Szafir, D. Stuffer, Y. Sohail, and M. Gleicher. TextDNA: Visualizing Word Usage with Configurable Colorfields. *Computer Graphics Forum*, 35(3):421–430, jun 2016. doi: 10.1111/cgf.12918

[77] 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, dec 2012. doi: 10.1109/TVCG.2012.237

[78] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[79] T. Urness, V. Interrante, E. Longmire, I. Marusic, S. O'Neill, and T. Jones. Strategies for the visualization of multiple 2D vector fields. *IEEE Computer Graphics and Applications*, 26(4):74–82, jul 2006. doi: 10.1109/MCG.2006.88

[80] Vivek Verma and A. Pang. Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624, nov 2004. doi: 10.1109/TVCG.2004.39

[81] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, pp. 139–143,. IEEE Comput. Soc. Press, 1990. doi: 10.1109/VISUAL.1990.146375

[82] J. Wood and J. Dykes. Spatially Ordered Treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, nov 2008. doi: 10.1109/TVCG.2008.165

Author's Pre-Print Version
Accepted, to appear in Proc. InfoVis 2017
IEEE TVCG 24(1), Jan 2018
DOI: 10.1109/TVCG.2017.2744199