Ph.D. Dissertation

Designing Information Visualization to Support Visual Comparison for Novices

시각화 초심자에게 시각적 비교를 돕는 정보 시각화 기술의 디자인

February 2020

Department of Computer Science and Engineering
College of Engineering
Seoul National University

Sehi L'Yi

Designing Information Visualization to Support Visual Comparison for Novices

Advisor: Jinwook Seo

Submitting a Ph.D. Dissertation of
Computer Science and Engineering

January 2019

College of Engineering
Seoul National University

Sehi L'Yi

Confirming the Ph.D. Dissertation written by
Sehi L'Yi

December 2019

Chair	Myung-Soo Kim	(Seal)
Vice Chair	Jinwook Seo	(Seal)
Examiner	Youngki Lee	(Seal)
Examiner	Jeongjin Lee	(Seal)
Examiner	Hyunjoo Song	(Seal)

Abstract

Sehi L'Yi

Department of Computer Science and Engineering

College of Engineering | Seoul National University

The visual comparison is one of the fundamental tasks in information visualization (InfoVis) that enables people to organize, evaluate, and combine information fragmented in visualizations. For example, people perform visual comparison tasks to compare data over time, from different sources, or with different analytic models. While the InfoVis community has focused on understanding the effectiveness of different visualization designs for supporting visual comparison tasks, it is still unclear how to design effective comparative visualizations due to several limitations: (1) Empirical findings and practical implications from those studies are fragmented, and (2) we lack user studies that directly investigated the effectiveness of different visualization designs for visual comparison.

In this dissertation, we present the results of three studies to build our knowledge on how to support effective visual comparison to InfoVis novices—general people who are not familiar with visual representations and visual data exploration process. Identifying the major stages in the visualization construction process where novices confront challenges with visual comparison tasks, we explored two high-level comparison tasks with actual users: comparing visual mapping (encoding barrier) and comparing information (interpretation barrier) in visualizations. First, we conducted a systematical literature review on research papers ($N = 104$) that focused on supporting visual comparison tasks to gather and organize the practical insights that researchers gained in the wild. From this study, we offered implications for designing comparative visualizations, such as actionable guidelines, as well as the lucid categorization of comparative designs which can help researchers explore the design space. In the second study, we performed a qualitative user study ($N = 24$) to investigate how novices compare and understand visual mapping suggested in a visual-encoding recommendation interface. Based on the study, we present novices' main challenges in using visual encoding recommendations and design implications as remedies. In the third study, we conducted a design study in the area on bioinformatics to design and implement a visual analytics tool, XCluSim, that helps users to compare multiple clustering results. Case studies with a bioinformatician showed that our system enables analysts to easily evaluate the quality of a large number of clustering results. Based on the results of three studies in this dissertation, we suggest a future research agenda, such as designing recommendations for visual comparison and distinguishing InfoVis novices from experts.

Keywords: Information Visualization; Comparative Analysis; Visual Comparison; InfoVis Novices; Literature Survey; User Study; Design Study
Student Number: 2013-23127

Contents

Abstract i

CHAPTER 1
Introduction 1

1.1 Background and Motivation 1

1.2 Research Questions and Approaches 4

1.2.1 Revisiting Comparative Layouts: Design Space, Guidelines, and Future Directions 5

1.2.2 Understanding How InfoVis Novices Compare Visual Encoding Recommendation 6

1.2.3 Designing XCluSim: a Visual Analytics System for Comparing Multiple Clustering Results 7

1.3 Dissertation Outline 8

CHAPTER 2
Related Work 9

2.1 Visual Comparison Tasks 9

2.2 Visualization Designs for Comparison 10

2.2.1 Gleicher et al.'s Comparative Layout 11

2.3 Understanding InfoVis Novices 12

2.4 Visualization Recommendation Interfaces 13

2.5 Comparative Visualizations for Cluster Analysis 14

CHAPTER 3

Comparative Layouts Revisited: Design Space, Guidelines, and Future Directions 19

3.1 Introduction 19

3.2 Literature Review 21

3.2.1 Method 22

3.3 Comparative Layouts in The Wild 23

3.3.1 Classifying Comparison Tasks in User Studies 25

3.3.2 Same Layout Is Called Differently 26

3.3.3 Lucid Classification of Comparative Layouts 28

3.3.4 Advantages and Concerns of Using Each Layout 30

3.3.5 Trade-offs between Comparative Layouts 36

3.3.6 Approaches to Overcome the Concerns 38

3.3.7 Comparative Layout Explorer 42

3.4 Discussion 42

3.4.1 Guidelines for Comparative Layouts 44

3.4.2 Promising Directions for Future Research 48

3.5 Summary 49

CHAPTER 4

Understanding How InfoVis Novices Compare Visual Encoding Recommendation 51

4.1 Motivation 51

4.2 Interface 53

4.2.1 Visualization Goals 53

4.2.2 Recommendations 54

4.2.3 Representation Methods for Recommendations 54

4.2.4 Interface 58

4.2.5 Pilot Study 61

4.3 User Study 62

4.3.1 Participants 62

4.3.2 Interface 62

4.3.3 Tasks and Datasets 65

4.3.4 Procedure 65

4.4 Findings 68

4.4.1 Poor Design Decisions 68

4.4.2 Role of Preview, Animated Transition, and Text 69

4.4.3 Challenges For Understanding Recommendations 70

4.4.4 Learning By Doing 71

4.4.5 Effects of Recommendation Order 71

4.4.6 Personal Criteria for Selecting Recommendations 72

4.5 Discussion 73

4.5.1 Design Implications 73

4.5.2 Limitations and Future Work 75

4.6 Summary 77

CHAPTER 5

Designing XCluSim: a Visual Analytics System for Comparing Multiple Clustering Results 78

5.1 Motivation 78

5.2 Task Analysis and Design Goals 79

5.3 XCluSim 80

5.3.1 Color Encoding of Clusters Using Tree Colors 82

5.3.2 Overview of All Clustering Results 83

5.3.3 Visualization for Comparing Selected Clustering Results 86

5.3.4 Visualization for Individual Clustering Results 92

5.3.5 Implementation 100

5.4 Case Study 100

5.4.1 Elucidating the Role of Ferroxidase in Cryptococcus Neoformans Var. Grubii H99 (Case Study 1) 100

5.4.2 Finding a Clustering Result that Clearly Represents Biological Relations (Case Study 2) 103

5.5 Discussion 106

5.5.1 Limitations and Future Work 108

5.6 Summary 108

CHAPTER 6

Future Research Agenda 110

6.0.1 Recommendation for Visual Comparison 110

6.0.2 Understanding the Perception of Subtle Difference 111

6.0.3 Distinguishing InfoVis Novices from Experts 112

CHAPTER 7
Conclusion 113

Abstract (Korean) 129

Acknowledgments (Korean) 131

List of Figures

Figure 1.1 The design space of comparative layouts observed in our literature survey. 5

Figure 1.2 The recommendation interface used in our user study for understanding how InfoVis novices compare visual-encoding recommendations. 6

Figure 1.3 The interface of XCluSim, a visual analytics tool for comparing multiple clustering results. 7

Figure 2.1 Three primitive visualization arrangements to support visual comparison tasks: (A) Juxtaposition, (B) Superposition, and (C) Explicit-Encoding. 11

Figure 3.1 A stacked bar chart showing the historical distribution of 359 publications which cited comparative layouts suggested by Gleicher et al. [37]. This data is based on Google Scholar on September 9, 2019. . . 20

Figure 3.2 Examples of comparative layouts observed in our literature survey. . 24

Figure 3.3 An interactive visual exploration tool for exploring the design space of the comparative layouts. 43

Figure 4.1 The seven types of recommendations (B-H) in our recommendation interface for alleviating over-plotting problems in the (A) specified scatterplots. 55

Figure 4.2 Three representation methods for a visualization recommendation: (A) Preview, (B) Animated Transition on mouse hover, and (C1-4) Textual Description. 56

Figure 4.3 Overall interface of the modified PoleStar[143] with the recommendation interface: (A) data panel, (B) encoding panel, (C) specified view, (D) pinned view, and (E) recommendation interface. 59

Figure 4.4 Configuration interfaces for recommendations: (A) toggle button for Represent Points Using Outlines and Represent Density of Points Using Color, (B) nominal field picker for Aggregate Points To Mean Position and Separate Graph By Category, (C) category picker for Filter By Category, and (D) slider bar for Change Point Size and Change Point Opacity. 60

Figure 4.5 Three combinations of representation methods used in our study: (A) Preview + Title (PT), (B) PT + Animated Transition (PTA), and (C) PTA + remaining Textual Description (PTAT). 64

Figure 4.6 Scatterplots constructed by the participants in our study in response to Question 1 to 6. Each scatterplot is labelled with the name of recommendations which are used to make the visualization. Recommendation names are abbreviated: (1) Filter By Category, (2) Change Point Opacity, (3) Change Point Size, (4) Represent Points Using Outlines, (5) Aggregate Points To Mean Position, (6) Separate Graph By Category, and (7) Represent Density of Points Using Color. The number in parentheses represents the number of participants who constructed the visualizations. 67

Figure 4.7 The number of times recommendations were chosen by their order during the task. 72

Figure 5.1 Visualization techniques for comparing multiple clustering results in XCluSim. There are three types of overviews: (A) parameter information view, (B) force-directed layout overview, and (C) dendrogram overview. They enable users to simultaneously compare multiple clustering results in a scalable way. When some clustering results are selected in the overviews, they are added to (D) enhanced parallel sets view for more in-depth comparison tasks. Users can access the detailed information of the selected clustering results with each result in each tab of (E) the tabular list view. 81

Figure 5.2 Three overviews supported in XCluSim. (A) The parameter information view provides the parameter settings used for the clustering results produced. The table in the parameter information view is for a clustering algorithm, and it shows a bar in each cell to represent the number of clustering results using the corresponding parameter setting. (B) The force-directed layout overview intuitively shows similarity among multiple clustering results with the distance between nodes representing similarity. (C) The dendrogram overview shows similarities between clustering results in a familiar dendrogram layout with a clustering result visualized at a terminal node. 84

Figure 5.3 The enhanced parallel sets view with various user interactions for in-depth comparison. (A) The parallel sets view provides rearranging algorithms that minimize line crossings. (B) When users hover a mouse pointer over the node of a cluster, the stable groups contained in it are highlighted while other stable groups fade out to reveal flows more clearly. By using a filtering feature on the stable group histogram at the bottom of the parallel sets view, users can hide less interesting bands. (C) Moreover, by using common angle plot [47], users can compare the sizes of different bands more accurately. 87

Figure 5.4 The tabular list view enables users to access numerical details. (a) Users can see detailed information for each item grouped by cluster or stable group. (b) Users also can see raw data in a heatmap form. When a user wants to access an item or a group directly, he/she can use the search box provided on the top of the tabular list view. 91

Figure 5.5 Interactive manipulation of input data supported in XCluSim: derive a new column (ratio, fold change), change color mapping, filter items using a range filter and RPKM adjustment. 93

Figure 5.6 Visualization techniques for individual clustering results in XCluSim. (a) Dendrogram + heatmap visualization for hierarchical agglomerative clustering results. (b) Force directed layout for every partitional clustering result and imported clustering results. (c) Common hive-shaped visualization for SOM clustering results. (d) Reachability plot together with parallel coordinates plot for OPTICS. 97

Figure 5.7 Results of the second case study are visualized in the enhanced parallel sets view. (A) The highlighted stable group contained the trp operon with yciF. (B) Visual comparison of two results: the best clustering result (“KMeans Clustering(5)”) derived from the case study and a result (“A Result from Original Paper”) presented in the original research paper [59]. 104

List of Tables

Table 3.1 The distribution of our target papers by venues. 25

Table 3.2 A summary of comparative layouts from 106 papers. The layouts were classified into five categories and one extra category (hybrid) for designs that combined layouts from two or more categories (e.g., chartwise juxtaposition and explicit-encoding). The number of hybrid layouts was broken down into the five categories and summed up for each category (numbers in parentheses). 27

Table 3.3 Diverse advantages and concerns of the comparative layouts, as well as the approaches to overcome the concerns observed from our literature review. 31

Table 3.4 A summary of eight papers with quantitative user studies in comparison tasks. 32

Table 4.1 Six questions based on scatterplot-related visualization tasks [107] used in our study. 63

Table 5.1 Taxonomy of visualization techniques for visualizing clustering results. Visualization components for visualizing clustering results use visual cues based on Gestalt principles of grouping [139] to represent cluster membership. We categorize the visualization components by principle and indicate how appropriate each visualization component is for showing clustering results by different types of clustering algorithms. (i.e. ”O“ for most appropriate, ”△“ for moderately appropriate, and ”X“ for not applicable). 94

Chapter 1

Introduction

1.1 Background and Motivation

During visual data exploration with information visualization (InfoVis), comparing multiple visualizations is one of the fundamental tasks that enables people to organize, evaluate, and combine information fragmented in visualizations. For example, people perform visual comparison tasks to compare data over time [2, 79, 118], from different source [10], or with different analytic models [18, 66]. While the InfoVis community has focused on understanding the effectiveness of different visualization designs for supporting visual comparison tasks [36, 37, 53, 81, 94, 106, 125], it is still unclear how to design comparative visualizations because of several limitations. (1) Insights are fragmented. Empirical findings and practical implications from these studies are fragmented in diverse domains and venues. For example, research papers that presented novel visualization designs based on *comparative layouts* [37]—three primitive visualization arrangements that facilitate visual comparison tasks (i.e., juxtaposition, superposition, and explicit-encoding)—are published to more than 40 different venues (Chapter 3). This sometimes caused inconsistency when researchers assessed the effectiveness of different comparative layouts; contrary to common belief on the ineffectiveness of juxtaposition [19, 24], user study results show that juxtaposition can be more effective than other two comparative layouts for some certain tasks [53, 94]. (2) We lack user study results. To assess the effectiveness of different visualization designs, we need to empirically evaluate them with actual users. However, we only find a few studies which directly investigated the effectiveness of different visualization designs for visual comparison; we found less than 10 research papers that conducted quantitative user studies with the comparative layouts (Chapter 3).

When we consider InfoVis novices target users, supporting effective visual comparison becomes much more challenging. Here, we follow Grammel et al.'s work [41] to define InfoVis novices: Novices are general people who are not familiar with visual representations and visual data exploration and can be any domain experts, such as bioinformaticians and system log analysts. Previous results from controlled user studies [41, 54] identified that InfoVis novices confront several barriers during visual data exploration. Based on three main steps in the visual exploration process [17], Grammel et al. [41] identified three barriers—data selection, encoding, and interpretation barriers—where we find that the last two can directly affect novices in performing visual comparison tasks. First, **interpretation barrier** challenges novices in understanding visual representations used in unfamiliar visualizations. This barrier is related to typical comparison tasks [53, 81, 94, 118, 125] where people compare information conveyed in multiple visualizations. Second, **encoding barrier** makes novices difficult to imagine and understand visual mapping operations, such as transforming numerical values to the length of bars or categorical values to distinguishable hues in bar charts. This barrier can cause cognitive burden in selecting visual mapping alternatives during constructing visualizations, for example, making novices difficult to choose proper visual encoding suggested in *visual-encoding recommendation interfaces* [141], such as Recommended Charts in Microsoft Excel [32] and Show Me in Tableau [120].

In this dissertation, we present the results of three main studies to build our knowledge on how to support effective visual comparison to InfoVis novices. We employed three different approaches for individual studies to broaden our understanding: (1) **literature survey**, (2) **user study**, and (2) **design study**. In the first study, we conducted a systematical literature review on research papers that focused on supporting visual comparison tasks to gather and organize the insights that researchers gained *in the wild*. In the second and third studies, we investigated the effectiveness of visualization techniques with actual users and real-world problems. In these two studies, we explored comparison tasks in two major stages of the visual exploration process where novices usually confront challenges [41] and are directly related to comparison tasks: **comparing visual mapping** (encoding barrier) and **comparing information** (interpretation barrier) in multiple visualizations. In the second study, we performed a qualitative user study ($N = 24$) to investigate how novices compare and understand visual mapping in the context of visualization recommendation. In the last study, we conducted a design study in the area of bioinformatics to design and implement a visual analytics tool that helps domain experts to interactively compare multiple results of cluster analysis.

Thesis Statement Carefully designed visualizations and interfaces by understanding people's abilities, challenges, and goals in visual data exploration can facilitate effective visual comparison, ultimately leading to a better understanding of their complex data.

1.2 Research Questions and Approaches

The research questions that motivated this dissertation are the followings:

RQ1. How should we design visualizations to support InfoVis novices in visual comparison tasks?

RQ2. How should we help InfoVis novices in comparing and understanding visual encoding in visualization recommendation?

RQ3. How should we design a visual analytics system to help InfoVis novices in comparing multiple analysis results?

To answer these research questions, we employed three main approaches: systematic literature survey, qualitative user study, and design study.

A1. Literature Survey: A systematical literature review on 104 papers that employed three primitive visualization arrangements to support visual comparison.

A2. User Study: A qualitative user study with InfoVis novices to investigate how they compare visual-encoding recommendations.

A3. Design Study: A design and implementation of XCluSim, a visual analytics tool for comparing multiple clustering results.

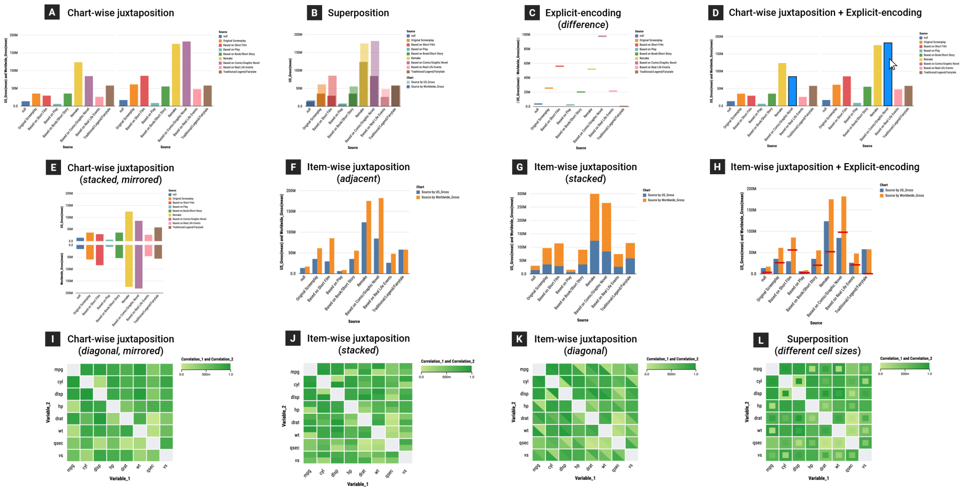
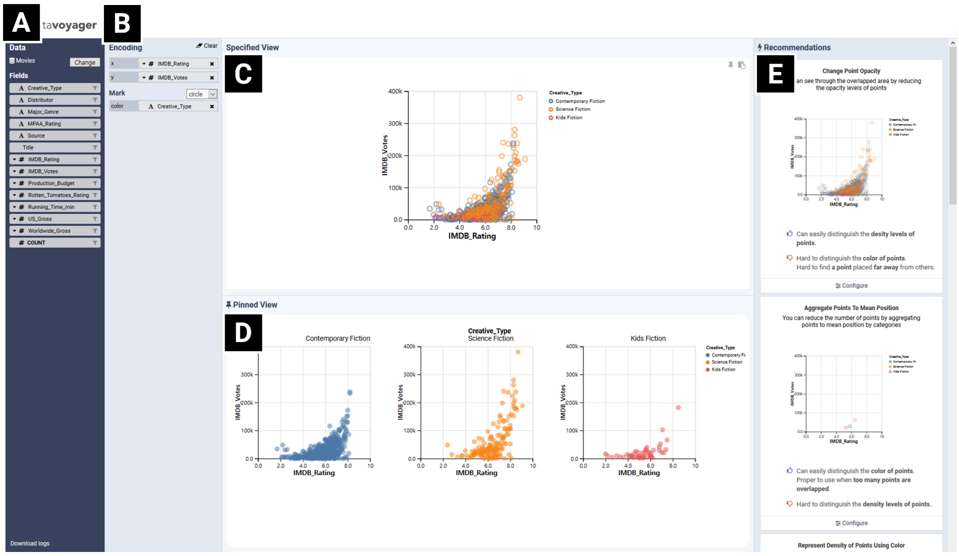


Figure 1.1: The design space of comparative layouts observed in our literature survey.

1.2.1 Revisiting Comparative Layouts: Design Space, Guidelines, and Future Directions

We present a systematic review on three comparative layouts—*juxtaposition*, *superposition*, and *explicit-encoding*—which are information visualization (InfoVis) layouts designed to support comparison tasks. In the last decade, these layouts have served as fundamental idioms in many visualization systems to support visual comparison. However, we found that the layouts have been used with inconsistent terms with confusion and the lessons and practical findings from previous studies are fragmented. We review 104 research papers that employed comparative layouts to combine and systematize the various insights researchers gained *in the wild*. Reflecting the diverse usage of the layouts (Figure 1.1), we classify the three layouts into six lucid categories, such as *chart-wise* and *item-wise juxtaposition*. We distill the advantages and concerns of using each layout, as well as the approaches to overcome the concerns. Combining our literature review and the results of eight papers



x: IMDB_Rating
y: IMDB_Votes
Mark: circle
color: Creative_Type

Change Point Opacity
an see through the overlapped area by reducing the opacity levels of points

Can easily distinguish the **desity levels of points**.

Hard to distinguish the **color of points**.
Hard to find **a point** placed **far away** from others.

Aggregate Points To Mean Position
You can reduce the number of points by aggregating points to mean position by categories

Can easily distinguish the **color of points**.
Proper to use when **too many points are overlapped**.

Hard to distinguish the **density levels of points**.

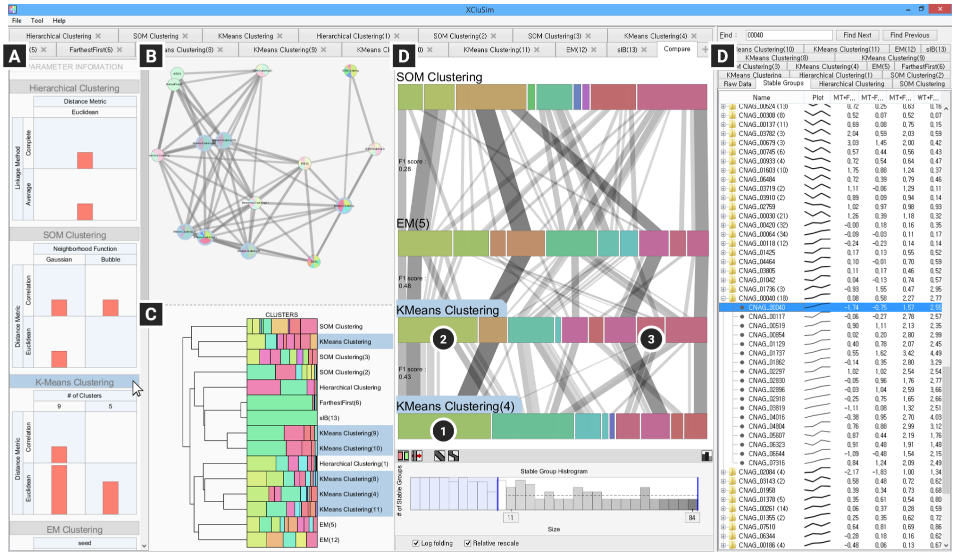
Represent Density of Points Using Color

Figure 1.2: The recommendation interface used in our user study for understanding how InfoVis novices compare visual-encoding recommendations.

with quantitative user studies, we suggest six actionable guidelines for the comparative layouts.

1.2.2 Understanding How InfoVis Novices Compare Visual Encoding Recommendation

We investigate the effectiveness of three representation methods—preview, animated transition, and textual description—in comparing and understanding the visual-encoding recommendation. Most visualization recommendation systems predominantly rely on graphical previews to describe alternative visual encodings. However, since InfoVis novices are not familiar with visual encoding and representations [41], novices might have difficulty comparing and understanding recommended visual encodings. We conducted a qualitative user study using a think-aloud protocol with 24 participants to explore the effectiveness of three representation methods for describing



Name	Plot	MT+F...	MT-F...	MT+F...	WT+F...
CNAG_00524 (13)		0,72	0,25	0,63	0,16
CNAG_00308 (8)		0,52	0,07	0,52	0,07
CNAG_00137 (11)		0,69	0,08	0,75	0,15
CNAG_03782 (3)		2,04	0,59	2,03	0,59
CNAG_00679 (3)		3,03	1,45	2,00	0,42
CNAG_00745 (6)		0,57	0,44	0,56	0,43
CNAG_00933 (4)		0,72	0,54	0,64	0,47
CNAG_01603 (10)		1,75	0,88	1,24	0,37
CNAG_06484		0,72	0,39	0,79	0,46
CNAG_03719 (2)		1,11	-0,06	1,29	0,11
CNAG_03910 (2)		0,89	0,09	0,94	0,14
CNAG_02759		1,02	0,97	0,98	0,93
CNAG_00030 (21)		1,26	0,39	1,18	0,32
CNAG_00420 (32)		-0,00	0,18	0,16	0,35
CNAG_00064 (34)		-0,09	-0,03	0,11	0,17
CNAG_00118 (12)		-0,24	-0,23	0,14	0,14
CNAG_01425		0,17	0,13	0,55	0,52
CNAG_04464		0,10	-0,01	0,70	0,59
CNAG_03805		0,11	0,17	0,46	0,52
CNAG_01042		0,04	-0,13	0,74	0,57
CNAG_01736 (3)		-0,93	1,55	0,47	2,95
CNAG_00040 (18)		0,08	0,58	2,27	2,77
CNAG_00040		-1,74	-0,75	1,57	2,55
CNAG_00117		-0,06	-0,27	2,78	2,57
CNAG_00519		0,90	1,11	2,13	2,35
CNAG_00854		0,02	0,20	2,80	2,99
CNAG_01129		0,40	0,78	2,07	2,45
CNAG_01737		0,55	1,62	3,42	4,49
CNAG_01862		-0,14	0,35	2,80	3,29
CNAG_02297		1,02	1,02	2,54	2,54
CNAG_02830		-0,05	0,96	1,76	2,77
CNAG_02896		-0,03	1,04	2,59	3,66
CNAG_02918		-0,25	0,75	1,65	2,66
CNAG_03819		-1,11	0,08	1,32	2,51
CNAG_04016		-0,38	0,95	2,70	4,03
CNAG_04804		0,76	0,88	2,99	3,12
CNAG_05607		0,87	0,44	2,19	1,76
CNAG_06323		0,91	0,48	1,91	1,48
CNAG_06644		-1,09	-0,48	1,54	2,15
CNAG_07316		0,84	1,24	2,09	2,49
CNAG_02084 (4)		-2,17	-1,83	1,00	1,34
CNAG_03143 (2)		0,58	0,48	0,72	0,62
CNAG_01958		0,39	0,34	0,73	0,68
CNAG_01378 (5)		0,35	0,61	0,54	0,80
CNAG_00261 (14)		0,06	0,37	0,28	0,59
CNAG_01355 (2)		0,25	0,35	0,62	0,72
CNAG_07510		0,64	0,81	0,69	0,86
CNAG_06344		-0,28	0,18	0,16	0,62
CNAG_00186 (4)		-0,48	0,06	0,13	0,67

Figure 1.3: The interface of XCluSim, a visual analytics tool for comparing multiple clustering results.

visualization recommendation. To conduct the user study, we design and implemented a visual-encoding recommendation interface to alleviate overlap reduction in scatterplots (Figure 1.2). Our results show how multiple representations cooperatively help users compare, understand, and choose recommended visualizations, for example, by supporting their expect-and-confirm process. Based on our study results, we discuss design implications for visualization recommendation interfaces.

1.2.3 Designing XCluSim: a Visual Analytics System for Comparing Multiple Clustering Results

We present XCluSim (Figure 1.3), a visual analytics system that helps people to interactively compare multiple clustering results. In collaboration with senior researchers in a bioinformatics laboratory, we conducted a design study to organize practical problems that data analysts confront in cluster analysis.

To assist the identified problems, we designed and implemented XCluSim based on Visual Information Seeking Mantra [114], allowing users to grasp overall information about multiple clustering results, such as the similarity between them, and to examine the small number of results in detail with two detail views. Finally, we conducted two case studies with a bioinformatician to evaluate the usefulness of XCluSim and found that XCluSim helped the analyst to find a clustering result that clearly represents the biological relations of genes.

1.3 Dissertation Outline

The rest of this dissertation is divided into four chapters. Chapter 2 discusses previous studies that are relevant to the work of this dissertation, including studies on information visualization techniques for supporting visual comparison tasks and understanding InfoVis novices through user studies. Chapter 3 illustrates the result of a systematical survey on research papers that employed comparative layouts: three primitive visualization layouts that support visual comparison tasks (i.e., juxtaposition, superposition, and explicit-encoding). Chapter 4 proposes the result of a qualitative user study with InfoVis novices to understand the effectiveness of three different comparative layouts—juxtaposition, animated transition, and explicit-encoding—in visual-encoding recommendation contexts. Chapter 5 presents XCluSim, an interactive visual analytics tool for comparing multiple clustering results of bioinformatics data. Lastly, Chapter 6 concludes this dissertation by proposing future research agenda.

Chapter 2

Related Work

2.1 Visual Comparison Tasks

The InfoVis community has focused on observing, identifying, and organizing visual analytic tasks in the real-world to design visualization systems that better reflect the practical usage of information visualization. Amar et al. [3] identified ten low-level tasks in information visualization systems, such as *computing derived value* and *finding extremum*. While this categorization is not specifically targeted for visual comparison, several tasks from this work can be employed in visual comparison contexts. For example, Howorko et al. [50] designed study tasks based on one of the low-level tasks (i.e., computing derived value) to evaluate the different designs of bar chart visualizations: *single-attribute* and *overall-attribute comparisons*. Gleicher [36] focused on visual comparison and proposed six fundamental actions that people perform, from low-level tasks (e.g., *identify* and *measure*) to high-level ones (e.g., *Contextualize*), which have been influenced on designing more effective visualizations for comparison [118]. Recently, Jardine et al. [53] divided comparison tasks into two categories—*global* and *local comparison*—which showed different performance in terms of consistency by participants from controlled user studies [53, 94]. The global comparison refers to comparing the overall characteristics of individual visualizations, such as examining the correlation of individual bar charts. On the contrary, the local comparison represents directly comparing visual elements, such as the length of two specific bars in bar charts. The authors found that people showed more inconsistent performance in global comparison tasks because of people's different perceptual heuristics for finding predefined relationships between visualizations. This dissertation employs these recent categorizations for visual comparison tasks [36, 53] to more systematically organize the performance of visualization designs found in our literature survey (Chapter 3).

2.2 Visualization Designs for Comparison

Many researchers have build the knowledge on designing information visualizations that effectively support comparison tasks through user studies [53, 77, 94, 106, 118] and literature surveys [36, 37]. For example, Ondov et al. [94] compared the effectiveness of different arrangements, such as overlaying and juxtaposing, for comparing a pair of bar, slope, and pie charts. Srinivasan et al. [118] focused on bar charts but with additional designs, such as overlaying 'tick' marks on top of a bar chart that represent subtraction values between two charts. While most of the studies focused on static visualizations, a few researchers introduced interactive designs in their studies, such as map visualizations with magic lens [81] and heatmap visualizations with interactive view replacements [125].

A body of pioneer studies for designing comparative visualizations is conducted by Gleicher et al. [36, 37] which proposed diverse insights for designing visualizations for comparison tasks, such as common design chal-

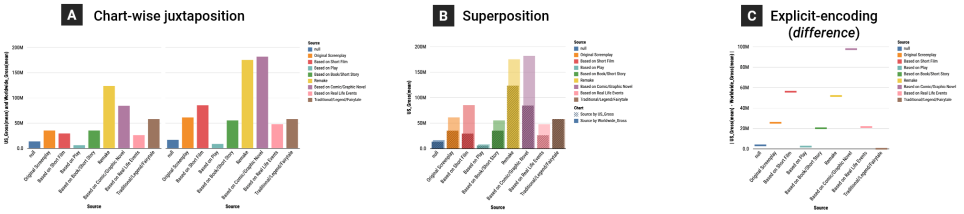


Figure 2.1: Three primitive visualization arrangements to support visual comparison tasks: (A) Juxtaposition, (B) Superposition, and (C) Explicit-Encoding.

lenges and their possible remedies, as well as design space of visual comparison, through a literature survey. The three primitive visualization arrangements to support visual comparison—juxtaposition, superposition, and explicit-encoding—have inspired on designing novel visualization representations and applications in the InfoVis community [4, 7, 9, 11, 14, 82, 137, 145, 151]. In Chapter 3, we based our literature review on the Gleicher et al.'s comparative layout to further develop our understanding of using the layouts to support visual comparison tasks.

2.2.1 Gleicher et al.'s Comparative Layout

Throughout this dissertation, we will use the terms from Gleicher et al. [37] to refer to comparative layouts: *juxtaposition*, *superposition*, and *explicit-encoding*. The three designs describe the arrangement of two or more visualizations to support comparison tasks. First, juxtaposition refers to placing visualizations next to each other (Figure 3.2A). It is sometimes called *spatial juxtaposition* to distinguish it from *temporal juxtaposition*, which temporally separates visualizations, for example, switching from one to another or using animated transition. The superposition layout refers to placing visualizations on top of each other, such as overlaying one bar chart on another (Figure 3.2B). Finally, explicit-encoding focuses on revealing the predefined relationship between visualizations. For example, if the difference between two trends is of interest, one can explicitly draw the difference on a bar chart with the two trends (Figure 3.2C). Note that the explicit-encoding layout is not limited to creating a new visualization with aggregated values but also includes visual elements overlaid on the original visualization (e.g., lines connecting the corresponding points in two scatterplots [76]). Designers can also combine the three layouts (i.e., *hybrid layout*), such as overlapping two node-link diagrams (superposition) with the common edges and nodes highlighted using a different color (explicit-encoding) [92].

2.3 Understanding InfoVis Novices

The InfoVis community has focused on understanding novices, people who are not familiar with visual representations, by performing various user studies. Using sketching [135] or tangible building blocks [51], researchers conducted exploratory studies to understand how novices transform data into visualizations. Smuts et al. [115] and Grammel and Storey [40] suggested several guidelines for supporting novices in designing visualization tools through user studies. Through an observation study, Grammel et al. [41] identified three challenges novices confront during a visualization construction process: *data selection*, *visual mapping*, and *interpretation barriers*. Motivated by Grammel et al.'s work, we presumed that novices might find it difficult to understand recommendations only with the most common representation (i.e., a preview for the result visualization) because novices have difficulties in interpreting visualizations (i.e., *interpretation barrier*). This dissertation investigated two more representation methods (i.e., animated transitions and textual descriptions) to explore how novices use recommendations with different representation methods.

Other studies compared visualization tools to understand how novices construct visualizations with different interfaces. Méndez et al. [87] compared novices' visualization construction process in two different types of interfaces: bottom-up approach (i.e., iVoLVER [88]) and top-down approach (i.e., Tableau [120]). Jo et al. [54] compared three visualization tools (i.e., TouchPivot, PivotTable of Microsoft Excel [32], and Tableau [120]) through controlled user studies and identified several hurdles for novices in the visualization tools. We go a step further to broaden the understanding of InfoVis novices with various recommendation representations through scatterplot construction tasks.

2.4 Visualization Recommendation Interfaces

Depending on the purpose of recommendations, interfaces may vary to some degree, but overall, recent visualization systems tend to use similar recommendation interfaces. In terms of layout, most systems use a gallery-based layout either showing multiple recommendations at once for easy comparison between alternatives [16, 25, 30, 32, 38, 58, 83, 108, 128, 131, 140, 142, 143] or a single recommendation while enabling easy exploration of alternatives [54, 111]. For representing individual recommendations, previews hold a dominant position [30, 32, 38, 58, 131, 142, 143], while simple textual descriptions are sometimes used with the preview [25, 54, 83, 108, 111, 128, 140]. We identified two types of previews in recommendation interfaces: abstract thumbnails and actual visualization results. As thumbnail previews provide abstract information about the recommended visualization, they are used to show chart types (e.g., Show Me in Tableau [83]). Although abstract thumbnails have a performance advantage for large data because they do not require detailed chart rendering, actual visualization results tend to be used for data-level and encoding-level recommendations (e.g., recommendations for using different data fields in the same chart type) for providing more detailed information.

In contrast, textual descriptions are used to provide additional information such as chart types (e.g., Bar Chart) [16, 30, 32, 38], data fields used in recommended visualizations (e.g., "IMDB Rating vs Rotten Tomatoes Rating") [58, 131, 142, 143], or more details about when to use a specific type of visualization [32] or what it is [142, 143]. Based on an exploratory study with InfoVis novices, Grammel et al. [41] claim that, to help users better understand recommendations, more in-depth explanations about the recommendations should be provided, including the advantages and disadvantages of using them. However, the effectiveness of textual descriptions in novices' visualization construction process has not been previously explored. In our study, we examined the effectiveness of three different representation methods for recommendations including the in-depth textual descriptions suggested by Grammel et al.

2.5 Comparative Visualizations for Cluster Analysis

Visualizations for Multi-dimensional Categorical Data

Since multiple clustering results can be treated as multi-dimensional categorical datasets, they can be visualized using various visualization techniques corresponding to the specific data types. These techniques include Parallel Sets [13] and Parallel Coordinate Plot [52]. Lots of prior work on the visual comparison of multiple clustering results employed these techniques [28, 43, 72, 74, 75, 98, 112, 150], but we focus our discussion on the ones that are most relevant to us in terms of utilizing ribbon-like bands to represent concordance/discordance among multiple clustering results.

In iGPSe [28], to visually compare clustering results of two different expression data types (i.e. gene expression and micro-RNAs expression), two-dimensional axes were juxtaposed, allowing for the use of parallel sets. By observing the flow of ribbon-like bands, users were easily able to see which items were shared between a pair of clusters from two different clustering results. HCE [112] also juxtaposed a pair of hierarchical clustering results in parallel to enable comparison tasks with the two results. In contrast to iGPSe, HCE used a partitioned heatmap instead of a simple node to show the details of each data item. To reveal the relations between items in a pair of heatmaps, matching items were connected with straight lines. However, these two visual analytics tools only supported the comparison of a pair of clustering results. Moreover, because they used connectivity between related items, it was often the case that there were too many crossing lines with a large dataset.

CComViz [150] alleviated the line crossing problem while focusing on the comparison tasks of more than two clustering results. In their work, multiple clustering results were visualized with a parallel coordinate plot: clustering results as dimensions, clusters as vertical positions in each dimension, and items as lines. Users could grasp the overall distribution of items across multiple clustering results by tracking the flow of lines crossing multiple dimensions. Similar representations were used in [43], but CComViz devised an algorithm for rearranging clusters and their members to minimize visual clutter between each dimension. Matchmaker [74] also utilized the parallel coordinate plot, but to show raw data simultaneously, partitioned heatmaps were shown in dimensional axes. The items in each dimension were rearranged by their average values so that heatmaps clearly showed the patterns of their raw data. Unlike the case of CComViz, in this case, partitioned heatmaps used a bundling strategy to maintain the position of each item in a dimension. This reduced line crossings between adjacent dimensions. Although this method generated a clearer overview of the distributions of items, it had some drawbacks. First of all, the flows of inner lines were invisible unless users explicitly highlighted the lines. Secondly, since the lines were bundled, the width of a band may not have accurately conveyed the number of the items belonging to the band.

CComViz and Matchmaker were probably most relevant to XCluSim. They depended on a linear ordering of dimensions (or clustering results), which made it difficult to do the all-pairs comparison with a large number of clustering results at once. For example, as the authors said, Matchmaker only enabled users to compare, at most, six clustering results simultaneously, even with the limited linear ordering of dimensions. Since the same dataset can yield a large number of different clustering results, it is necessary to provide a more scalable way of comparing them. In XCluSim, we present diverse overviews to help in comparison tasks with many clustering results.

Visualizations Using Similarity Measures

There are a few approaches to visualize measured similarity values between clusters (or items) in different clustering results instead of explicitly visualizing shared items among multiple clustering results. Sharko et al. [113] utilized a color-coded similarity matrix view to show the stability between items or clusters across different clustering results. Similarities were measured by counting how many times each pair of items was clustered together or how many items each pair of clusters shared. Kothur et al. [64] used bar charts arranged in a matrix layout to show similarity values between a pair of clusters. However, these two works were restricted to comparing a pair of clustering results since they both used a matrix layout.

iGPSe [28] used Silhouette Plot [104] to help compare a pair of clustering results. Each item got a standardized dissimilarity value ranging from -1 to 1. This value represented dissimilarity in such a way that, when a value was close to 1, its average dissimilarity from all other items in the same cluster was much smaller than the maximum average dissimilarity from all items in another cluster. When the value was close to -1, the meaning of the value was reversed. By representing these similarity values between clustering results using a bar chart, users were able to assess the relative quality of clustering results.

These previous works using similarity measures allowed for comparisons of only a small number of clustering results. However, it is clear that, by abstracting detailed differences to simpler similarity measures, the visual comparison could be rendered more scalable. In our work, we used a graph layout and a dendrogram to show similarity overviews in a more scalable way.

Color Encoding for Cluster Similarity

Color is a powerful visual cue for representing a cluster membership. It is used in many visualization techniques, including parallel coordinate plot [113, 150] and scatterplot [5, 49, 55], to discriminate clusters while revealing trends in raw data. Similar efforts exist in the visualizations of multiple clustering results. For example, when using the parallel sets view, a few distinct colors are used to encode each cluster to discriminate it from others [28, 150].

However, if there are clusters from different clustering results that share the same members, it is not desirable to encode them in distinct colors since it may mislead a user into thinking that those clusters are different. Moreover, when the number of clusters increases, it is hard to color-code clusters differently, because it is hard to discriminate between more than 10 colors.

A useful color encoding strategy is Tree Colors [123], which was devised for tree-structured data to represent similarities between nodes. A part of the parent's hue range is recursively assigned to its child nodes. As a result, nodes with the same parent have similar colors, while those that are less similar have different colors. Moreover, this color scheme reflects the level of a node by using differentially encoded chroma and luminance at each level. If the similarities between clusters from multiple clustering results can be represented as a tree structure, Tree Colors may be well-suited to represent similarity among them. In XCluSim, we used this color scheme to color-code clusters after building a hierarchical structure by running a hierarchical agglomerative clustering (HAC) [29] with all clusters.

Chapter 3

Comparative Layouts Revisited: Design Space, Guidelines, and Future Directions

This chapter introduces the results of a literature survey on research papers ($N = 104$) that employed three comparative layouts: juxtaposition, superposition, and explicit-encoding.

3.1 Introduction

A decade ago, Gleicher et al. [37] suggested three primitive information visualization (InfoVis) layouts that support comparison tasks—*juxtaposition*, *superposition*, and *explicit-encoding*—based on their literature survey on 104 research papers. These layouts has served as fundamental idioms for designing comparative visualizations in diverse areas such as radiology [116], biology [129], and geology [1]. In addition, the layouts have been also popular in academia, as shown in the rapid growth of the number of papers citing the comparative layouts (Figure 3.1).

To develop a better understanding of comparative layouts, researchers have also attempted to study the effectiveness of the three layouts by conducting user studies and extending the layouts to specific domains. Gleicher et al. [37] initially discussed the potential strength and weakness of the comparative layouts in terms of scalability, cognitive cost, and task performance, followed by many user studies in the human–computer interaction (HCI) field [53, 79, 81, 92, 94, 106, 109, 118, 125]. Ondov et al. [94], for example, compared the variants of juxtaposition and superposition, such as using adjacent, mirrored, and animated arrangements, in identifying max change and correlation between two visualizations.

However, we find the lessons and practical findings from those previous studies fragmented, sometimes even with inconsistent terms. For example, we encounter several visualizations techniques (e.g., variants of bar charts or heatmaps) that are inconsistently regarded as either juxtaposition or superposition [2, 61, 94, 106, 118, 148]. Moreover, contrary to the general consensus that superposition is more effective for small difference [19, 24, 45], recent

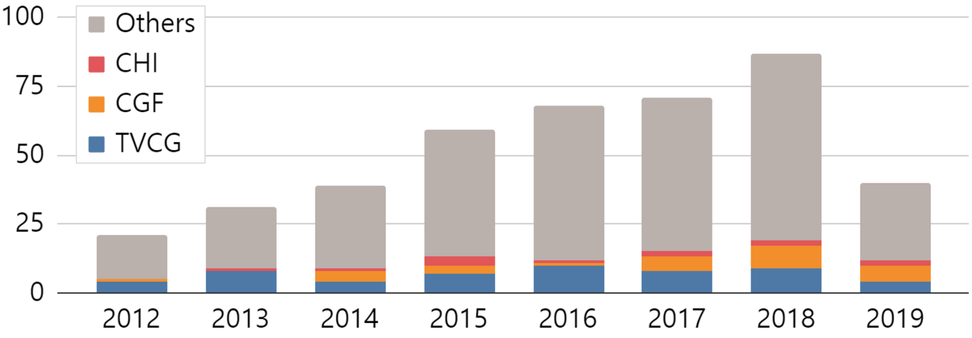


Figure 3.1: A stacked bar chart showing the historical distribution of 359 publications which cited comparative layouts suggested by Gleicher et al. [37]. This data is based on Google Scholar on September 9, 2019.

studies show that juxtaposition can be more effective for some tasks, such as comparing global characteristics between two bar charts [53, 94].

We present a systematic review on three comparative layouts with 104 research papers that employed the layouts. The focus of our study is to combine and systematize the insights gained *in the wild*, for example, during a visualization design process in collaboration with data analysts or in evaluation with actual users. To give implications in a more systematic and precise manner, we first alleviate the unambiguous boundaries between comparative layouts using lucid classification (e.g., *chart-wise* and *item-wise juxtaposition*). We explore the comparative layouts in diverse aspects, such as the advantages and concerns of using them in real-world scenarios and the researchers' approaches to overcome the concerns. Combining our literature review and the results of eight papers with quantitative user studies, we suggest six actionable guidelines for the comparative layouts. Finally, we propose a web-based interactive visual exploration tool to support designers in exploring the design space of the layouts. The contribution of this chapter is threefold:

1. We perform a systematic review on 104 research papers to better understand the comparative layouts in the wild.
2. We offer implications for using the comparative layouts as well as performing future research.
3. We propose a lucid classification of the comparative layouts with a web-based interactive tool for exploring the design space.

3.2 Literature Review

We reviewed 104 research papers that employed the comparative layouts to expand our understanding of the layouts.

3.2.1 Method

First, we looked into all the 354 publications that cited the work of Gleicher et al. [37] using Google Scholar. We then excluded irrelevant papers using the following criteria: (1) papers which do not explicitly use the comparative layouts or do not present any discussions about them (e.g., some papers mentioned the comparative layouts only to provide high-level contexts of comparative visualization in introduction), (2) duplicate publications (e.g., thesis papers), and (3) papers written in languages other than English. Lastly, we excluded (4) papers which mainly focusing on scientific visualization (e.g., 3D blood flow simulation [129]) to stick to the original focus of the comparative layouts [37], that is information visualization (InfoVis). After the filtering process, we obtained a set of 104 selected publications.

We surveyed the following factors from the selected papers, which were the factors discussed in previous papers [36, 37]:

- The type of visualizations placed using the layouts
- The number of visualizations to compare at once [36]
- How each of the comparative layouts [37] is used
- How researchers describe the advantages and concerns of using each layout
- Researchers' approaches to overcome the concerns

For more in-depth inspection of the papers with quantitative user studies, we additionally collected study conditions such as the comparative layouts used for independent variables, study tasks, and the number of participants.

To avoid ambiguity in collecting the usage of the comparative layouts, we mainly based our data collection on the authors' justifications described in the papers. Even though visualizations are placed adjacently as many general visualization systems support, we have not regarded this as using a comparative layout unless the authors explicitly stated because it is unclear whether the layout is used for visual comparison. We have not also considered the cases where the different visualization types are placed using the comparative layouts because comparison tasks are most likely to be taken with the same visualizations. One typical example in our review is the difference (explicit-encoding) overlaid on top of a grouped bar chart (juxtaposition) [118] (Figure 3.2H). In this case, consistent to the authors' explanation, we did not consider it as using an additional superposition layout between the juxtaposed bar chart (Figure 3.2F) and the explicit-encoding chart (Figure 3.2C), because these two charts are not arranged for comparing the two.

3.3 Comparative Layouts in The Wild

Overall, we found 197 visualization layouts from 104 papers (about 1.9 layouts per paper). The most widely used layout is juxtaposition (75), while superposition (38) and explicit-encoding (35) are used frequently as well. We also found 41 layouts that used multiple layouts at once (i.e., hybrid layout). The most widely used visualization types include bar charts (39), heatmaps (33), node-link diagrams (30), line charts (19), map visualizations (15), and scatterplots (12). Of the papers, eight papers presented quantitative user studies using the comparative layouts as independent variables in comparison tasks. The 104 papers have been published at 45 venues (Table

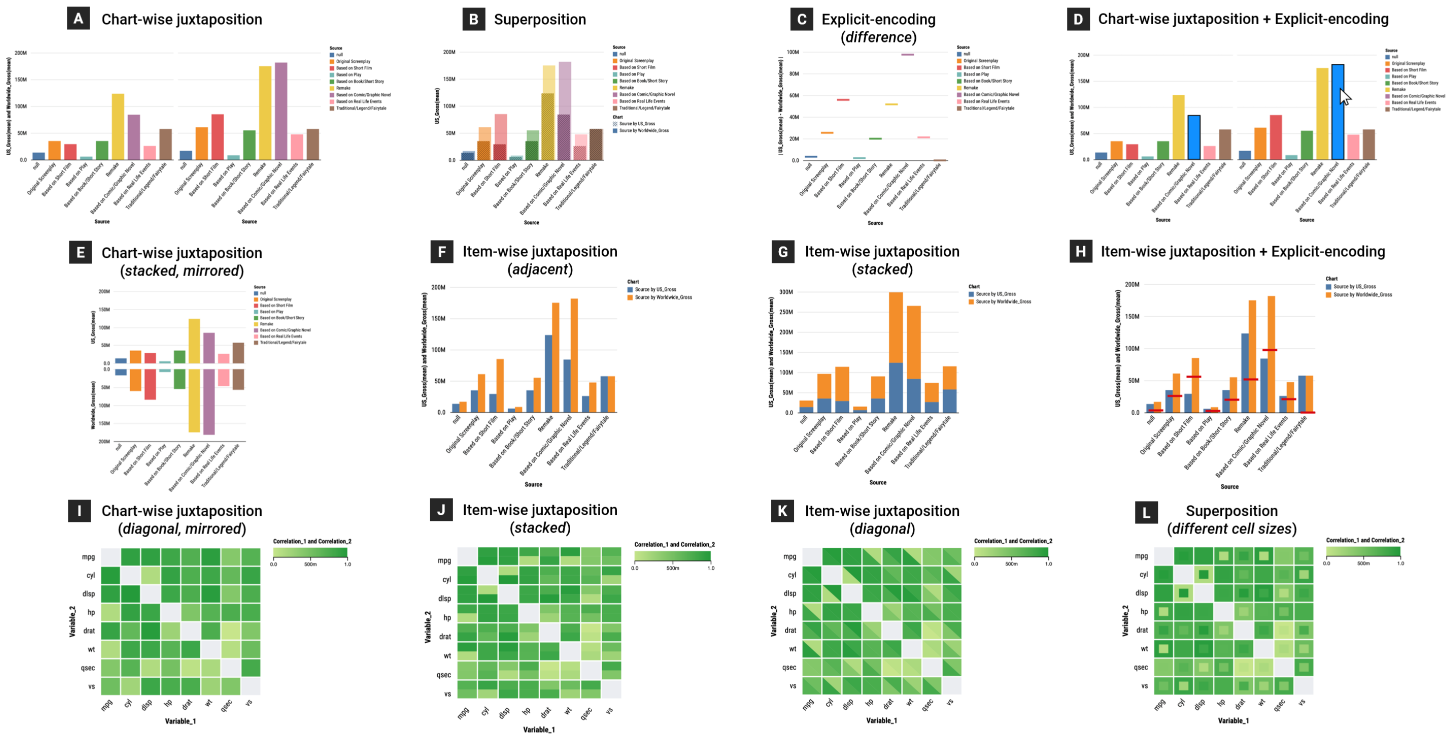


Figure 3.2: Examples of comparative layouts observed in our literature survey.

3.1); the majority of papers were from IEEE Transactions on Visualization and Computer Graphics (*TVCG*) (32), Computer Graphics Forum (*CGF*) (11), and ACM CHI Conference on Human Factors in Computing Systems (*CHI*) (7).

3.3.1 Classifying Comparison Tasks in User Studies

For a more comprehensive examination of the eight papers with quantitative user studies, we classified the comparison tasks (total 36 tasks) in a more detailed manner (Table 3.4). Following Gleicher et al.'s task categorization for visual comparison [36], we classified tasks with different user actions, where two of the actions were the primary focus across the papers: *identify* (all eight papers) and *measure* (2). Further, we categorized the study tasks by the target of comparison, that is ***chart*** (3) and ***item*** (all eight papers) because we find that these two types of tasks are quite distinguishable in terms of how people perform visual comparison. A similar categorization is also suggested in a very recent work [53]. *Chart-wise tasks* refer to comparing the overall characteristics of individual visualizations, such as comparing the correlation of each bar chart. In contrast, *item-wise tasks* refer to comparing between visual items, such as comparing bars in two bar charts. The main characteristic of

Venue	*Papers*
TVCG	32
CGF	11
CHI	7
PacificVis	4
IV	4
[45 other venues]	
total	106

Table 3.1: The distribution of our target papers by venues.

item-wise tasks compared to chart-wise ones is that people must link the corresponding visual elements between visualizations before actually comparing them (e.g., finding bars of the same category in two distant bar charts) unless a system explicitly highlights them. On the other hand, chart-wise tasks require more global perspectives that people seem to use more diverse perceptual heuristics in taking the comparison tasks [53].

3.3.2 Same Layout Is Called Differently

We found inconsistency in the use of the terms when referring to the three comparative layouts in the research papers. The most popular alias for juxtaposition was *side-by-side* [82, 137, 145]), followed by *small multiples* [10, 48, 133] especially for a grid arrangement to compare more than two visualizations at once. Ming et al. [89] called juxtaposition *separation* in that juxtaposition is used to separate visualizations spatially or temporally. Similarly, superposition was named *superimposition* [9, 14, 151] and *overlaying* [10, 57, 137]. Schmidt et al. [109] and Tominski et al. [125] called superposition *blending* and *shine-through*, respectively, in that they overlap two semi-transparent visualizations. People sometimes called explicit-encoding *direct encoding* [124] to emphasize that predefined relationships between visualizations are "directly" computed and represented. Schmidt et al. [109] used the term *aggregation* since explicit-encoding frequently derives aggregated values such as the difference between the categories in two bars charts [118]. Since explicit-encoding shows abstract values instead of raw data, Maries et al. [84] used a term *abstraction*. In contrast, Zaman et al. [144] termed explicit-encoding *additive encoding*, emphasizing that people generally use additional visual elements to represent the connection between two visualizations (e.g., lines connecting pairs of visual marks between two visualizations).

We also found that the same arrangement of visualizations is often called differently. One common case is to call a chart with juxtaposed visual marks (e.g., a grouped bar chart; Figure 3.2F) either juxtaposition or superposition. For example, Srinivasan et al. [118] called a grouped bar chart the juxtaposition layout in that the chart places bars side by side. In contrast, Ondov et al. [94] treated the same chart as a superposition layout, considering the chart as multiple bar charts overlaid with different offsets. Similar problems occur in the case of matrix visualizations [2, 148]. Temporal juxtaposition, animated transition between multiple charts, is sometimes considered as superposition in that it shows multiple visualizations [61, 106]. Superposition and explicit-encoding are also ambiguous for specific visualization designs. For example, in the case where two node-link diagrams are shown in a single view with common edges and nodes highlighted, one can consider it either as a single union node-link diagram with explicit-encoding [106] or as superposition of two node-link diagrams with explicit-encoding [92].

Layout		*#*
chart-wise juxtaposition	63	(89)
item-wise juxtaposition	17	(24)
superposition	32	(49)
explicit-encoding	33	(70)
animated transition	6	(6)
hybrid	40	–
total	191	–

Arrangement of Juxtaposition	*#*
adjacent	64
stacked	26
grid	17
mirrored	11
diagonal	5
free-form	4
others	3

Table 3.2: A summary of comparative layouts from 106 papers. The layouts were classified into five categories and one extra category (hybrid) for designs that combined layouts from two or more categories (e.g., chart-wise juxtaposition and explicit-encoding). The number of hybrid layouts was broken down into the five categories and summed up for each category (numbers in parentheses).

3.3.3 Lucid Classification of Comparative Layouts

To more systematically organize the insights gained in the literature review and provide implications for the comparative layouts in a more precise manner without confusion, we found it is necessary to alleviate the ambiguous boundaries between comparative layouts. We propose to classify the three comparative layouts into five categories: (1) chart-wise juxtaposition, (2) item-wise juxtaposition, (3) animated transition, (4) superposition, and (5) explicit-encoding and an extra hybrid category. Table 3.2 shows the overall distribution of each category observed in our target papers.

Chart-wise and Item-wise Juxtaposition

To reflect the diverse variants of juxtaposition layouts, we suggest two subcategories for juxtaposition with six different ways of arrangements. We classified original juxtaposition into ***chart-wise*** and ***item-wise juxtaposition***, distinguishing the type of targets that are arranged using juxtaposition (i.e., chart or visual elements). For example, placing two bar charts side by side (i.e., concatenating two bar charts) is chart-wise juxtaposition (Figure 3.2A), while arranging bars next to each other (i.e., grouped bar charts) is item-wise juxtaposition (Figure 3.2F). In chart-wise and item-wise juxtaposition, we discovered six different ways of arranging visualizations or visual elements—*adjacent*, *stacked*, *grid*, *mirrored*, *diagonal*, and *free-form* (Table 3.2)—where three terms are brought from the recent study [94] (i.e., adjacent, stacked, and mirrored). For example, adjacent and stacked arrangements refer to placing charts or visual elements in a horizontal and vertical axis, respectively, constructing either a grouped or a stacked bar chart in the item-wise version (Figures 3.2F and 3.2G). In our survey, several matrix-like visualizations used diagonal arrangements for chart-wise and item-wise juxtaposition layouts (Figures 3.2I and 3.2K). The free-form arrangements are supported when people can interactively rearrange the visualizations without any restrictions. The mirrored arrangement is placing visualizations symmetrically (Figure 3.2E), which can be used with another arrangement where the adjacent arrangement is most frequently used with the mirrored layout.

Superposition refers to designs that combine multiple visualizations into one visualization with a unified coordinate system. In contrast to chart-wise or item-wise juxtaposition, visual elements can overlap in superposition (e.g., nodes and links can overlap if two node-link diagrams are superposed [2]). While juxtaposition and superposition refer to static designs, the *animated transition* category refers to the designs that use the temporal transition from one chart to another to highlight the difference between multiple charts. The transition usually takes place on the same visualization space, showing a single chart at a time that distinguishes animated transition from juxtaposition or superposition. *Explicit-encoding* refers to the use of extra visual elements that help comparison. For example, one can draw lines between two scatterplots to connect the corresponding points [57]) or highlight common edges or nodes between two network diagrams with a different color [92]. Note that explicit-encoding can be used without juxtaposition or superposition; for example, if the difference between two bar charts is of interest, one can draw a separate bar chart that only shows the difference without the original bars (Figure 3.2C).

In practice, two layouts from different categories can be used together, which refers to *hybrid layout*. For example, to help people more easily find the related bars in juxtaposed bar charts, systems can highlight them using a different color (explicit-encoding) upon user interaction (Figure 3.2D). A separate visualization that is constructed using explicit-encoding can be also overlaid on top of juxtaposed bar charts (Figure 3.2H), to support accessing both the difference and the original information. Highlighting common or unique visual elements in superposed node-link diagrams also belongs to this layout.

3.3.4 Advantages and Concerns of Using Each Layout

In this section, we reflect on the advantages and concerns of using each layout suggested in the papers to develop our understanding of the comparative layouts in the real-world scenarios (Table 3.3). As we were able to find only a few discussions of item-wise juxtaposition, we discuss item-wise juxtaposition in the later section.

Chart-wise Juxtaposition

The advantage of chart-wise juxtaposition mainly stems from its characteristic that it does not significantly change the original visualization [22, 79, 81, 84], which is sometimes the main reason for choosing chart-wise juxtaposition over other layouts [84]. Another related advantage is its ability to support separate analyses of individual visualizations [22, 97, 106], which is an important factor for professional analysts in network analysis [106]. Researchers also advocate its applicability to any visualizations [6] or its simplicity in implementation: "[Juxtaposition is] simple, even trivial" [9]. When two visualizations are juxtaposed and mirrored, it is known that the human perception system effectively recognizes the symmetry between two visual representations [127] which facilitates comparison between the two. A recent work [94] provided practical evidence that juxtaposing two charts in a mirror manner was more efficient than using animated transition or item-wise juxtaposition for comparing the correlation of individual bar charts.

Juxtaposition

Advantages	#
no visual interference	8
supporting other tasks	3
simple implementation	2
straight-forward comparison	2
suitable for large difference	2
easy comparison	2
familiar	2
high preference	1
convenient comparison	1

Concerns	#
limited scalability	12
cognitive burden	6
ineffective comparison	5
unsuitable for subtle difference	4
difficulty in linking items	4
difficulty in comparison	4
managing consistency	4
unsuitable for complex stimuli	2
long eye movement	2
unsuitable for large difference	1

Approaches	#
using hybrid layout	7
managing consistency	5
shortening distance	4
filter	2
change arrangement	1

Explicit-Encoding

Advantages	#
suitable for subtle difference	4
effective comparison	3
reasonable scalability	3
high preference	1

Concerns	#
information loss	4
precluding other tasks	3
unfamiliar	1

Approaches	#
using hybrid layout	1

Superposition

Advantages	#
effective comparison	4
suitable for subtle difference	4
easy comparison	3
effective interpretation	2
minimize eye movement	2
suitable for large difference	2
suitable for spatial data	2
less cognitive burden	1
high preference	1

Concerns	#
visual interference	16
limited scalability	3
visual separation	1
lacking intuitiveness	1
unfamiliar	1
precluding other tasks	1

Approaches	#
using hybrid layout	2
aggregating visual elements	1
managing opacity	1
filter	1

Animated Transition

Advantages	#
suitable for temporal data	2
effective comparison	2
structural change	1
showing constancy	1
showing causality	1
showing narratives	1
supporting other tasks	1

Concerns	#
no concurrent comparison	2
cognitive burden	2
unsuitable for large difference	2
ineffective comparison	2
requiring constant attention	1
difficulty in comparison	1

Approaches	#
managing consistency	1

Table 3.3: Diverse advantages and concerns of the comparative layouts, as well as the approaches to overcome the concerns observed from our literature review.

Paper	*Layout*	*Arrangement*	*N*	*Study Task*	*Visualization*
Jardine et al. [53]	juxtaposition (chart), juxtaposition (item), animated transition	adjacent, stacked, adjacent + mirrored	104 (MTurk)	identify chart (MaxMean, MaxRange)	bar chart
Liu and Shen [79]	juxtaposition (chart)	adjacent, adjacent + mirrored, diagonal + mirrored	28	identify item (IsChanged), general task	matrix
Lobo et al. [81]	juxtaposition (chart), superposition, interactive designs	adjacent	15	identify item (IsChanged, Unique)	map
Naragino and Misue [92]	juxtaposition (chart), superposition, juxtaposition (chart) + explicit-encoding, superposition + explicit-encoding	adjacent	18	identify item (Common, IsChanged, Unique), measure item (Delta)	node-link
Ondov et al. [94]	juxtaposition (chart), juxtaposition (item), superposition, animated transition	adjacent, stacked, adjacent + mirrored	200 (MTurk)	identify chart (MaxCorrelation), identify item (MaxDelta)	bar chart, donut chart, line chart
Sambasivan et al. [106]	juxtaposition (chart), animated transition, superposition + explicit-encoding	adjacent	26	identify chart (Delta), identify item (MaxDelta, IsChanged)	node-link
Schmidt et al. [109]	juxtaposition (chart), superposition + explicit-encoding	grid	11	identify item (Common, Delta, MaxDelta)	image
Srinivasan et al. [118]	juxtaposition (item), explicit-encoding, juxtaposition (item) + explicit-encoding, single chart + explicit-encoding	adjacent	74	identify item (Unique, MaxDelta), measure item (Delta), general task	bar chart

Table 3.4: A summary of eight papers with quantitative user studies in comparison tasks.

On the other hand, six studies have commonly claimed that the key concern of chart-wise juxtaposition is its limited scalability [37, 81, 109, 118, 134, 149]. For example, it is challenging to juxtapose a large number of visualizations simultaneously since the screen space is limited; as an extreme case, it is sometimes impossible to place even two visualizations at the same time in a mobile environment [149]. Another concern regarding chart-wise juxtaposition lies in its effectiveness in comparison. Tominski et al. [125] described this problem as “eyes have to move from one part to the other part,” which consequently leads people to rely on the mental image of the first part to compare it with the other part. In this sense, chart-wise juxtaposition has been criticized for such cognitive cost [79, 92, 93, 125] and considered as the least effective layout for comparison tasks compared with other layouts [2, 6, 24].

Specifically, researchers claimed that the subtle difference between juxtaposed visualizations is especially difficult to recognize [24, 94, 117, 137]: "Spot the difference games, in which observers try to detect small changes ..., illustrate the difficulty of [comparing between two regions]" [94]. Comparing complex visualizations (e.g., two node-link diagrams) is also claimed to be inefficient [57, 148] since people have to temporally remember a complicated representation. Another concern on chart-wise juxtaposition is that it is difficult to couple the corresponding visual elements from two distant visualizations [22, 45, 80, 122]. For example, Correll et al. [22] found that people often make mistakes when identifying relevant cells in two heatmaps with chart-wise juxtaposition. Emphasizing this issue, Lobo et al. [80] claimed that chart-wise juxtaposition can be effective "only if objects can easily be matched." Many researchers also added that, to be effective, designers should carefully optimize the consistency between visualizations [14, 27, 57, 61], such as using the same range for the axes in chart-wise juxtaposition or placing relevant visual elements in the same logical position in juxtaposed node-link diagrams.

Superposition

Superposition has been advocated for supporting comparison tasks [2, 56, 92, 93, 106], allowing a "quick and easy" comparison [23]. Subtle difference, which is challenging to recognize in chart-wise juxtaposition, can be visually salient in superposition [19, 24, 45] because target visual elements are arranged closely. Wang et al. [137] argued that superposition is "especially useful when the spatial location is a key component of the comparison," such as in geographical visualizations. The key concern on superposition is visual interference, that is, visual elements being overlapped challenge people in interpreting visualizations, which can lead to a scalability issue [61, 79, 92, 125, 133, 134, 149]. For example, Viola et al. [132] mentioned the complexity of this concern: "[T]he display of several data attributes quickly leads to visual clutter. There is thus no general methodology on how to design effective integrated multi-attribute visualizations." In this context, Caruso et al. [19] asserted that superposition can be useful only when target visualizations are similar enough. A qualitative study by Tominski et al. [125] showed that it is hard to compare two superposed heatmaps because of the blended color of each cell.

Explicit-Encoding

The main advantage of explicit-encoding is that it allows direct access to the predefined relationship [93, 124, 148]: "[T]he viewer does not need to make a mental comparison or find the difference, as it has already been calculated" [93]. For this reason, explicit-encoding can be used for designs where visualizing subtle difference is of importance [65]. Its second advantage is the scalability in terms of the number of target visualizations since it usually focuses only on showing the predefined relationships without showing the original visualization. For example, in a mobile environment, explicit-encoding can be more effective than juxtaposition or superposition [149] since the screen space is limited. Based on user studies, researchers also found explicit-encoding is useful when overlaid with other layouts (i.e., hybrid layouts). The hybrid layouts allowed a faster and more accurate comparison between node-link diagrams [92] and were more preferred by people [118] compared with using a single layout.

However, it can be ineffective if people can only see a specific relationship without the original information: "Ideally, we would like to see the entire dataset without missing any detail, but explicit-encoding concedes this design goal ... in favor of others" [61]. This seems a considerable drawback as data analysts described in a research paper [27] did not like such information abstraction: "[D]ue to information loss, scientists were not comfortable with the idea of smoothing by computation of average."

A relevant problem of explicit-encoding is called *decontextualization*, which involves losing contexts of data in visual representations: "The user sees the result of a comparison but cannot interpret it without additional visualization of the original data. This increases the complexity of the visualization" [134]. Another concern for explicit-encoding is its unfamiliarity. A study with treemap visualizations [77] showed that people occasionally misinterpreted a novel textual representation that encodes the direction of value changes. Similarly, participants from another study had difficulties in interpreting explicitly encoded differences (Figure 3.2F), and they rated explicit-encoding least effective compared with item-wise juxtaposition or hybrid designs [118].

Animated Transition

Animated transition is especially useful for recognizing a small local difference between two visualizations, as it outperformed item-wise and chart-wise juxtaposition in finding the maximum difference between a pair of bar charts or donut charts [94]. Because animated transition shows visualizations separately in time, it allows people to take independent analyses [106].

However, the drawback of animated transition is that people cannot see target visualizations at once [61, 118], which is known to be less effective than comparing concurrently visible representations [91] especially when the number of target visualizations increases. Moreover, animation requires constant attention and interaction (e.g., switching between views repeatedly) [2, 94, 148], which "may increase the time requirement" [2]. The performance of animated transition on comparison tasks is controversial; while animated transition showed outstanding performance in a study [94] with an item-wise task, it resulted in inaccurate comparison even with confusion with node-link diagrams [106]. Similarly, experts who used animated scatterplots to see multiple t-SNE results mentioned that watching animated transition was cognitively challenging: "[T]racking the nodes in an animated manner requires a mental map comparison, which is demanding ..." [76].

3.3.5 Trade-offs between Comparative Layouts

To assist designers in selecting comparative layouts, we suggest more practical design implications with trade-offs between the four most frequently used layouts—chart-wise juxtaposition (CJ), item-wise juxtaposition (IJ), superposition (S), and explicit-encoding (E)—in terms of four main themes: scalability, effectiveness in recognizing a relationship, familiarity, and supporting other types of tasks. We present a general consensus made by researchers in the effectiveness of each layout in the parentheses next to the names of each theme, where "*T* (*L1* > *L2*)" represents that the *L1 layout* is commonly said to be better than the *L2 layout* in terms of the *T* theme, and $\approx$ represents that their effectiveness depends on situations.

Scalability ($E > CJ \approx IJ \approx S$). Explicit-encoding is commonly regarded as the most scalable layout for the increasing number of target visualizations because it focuses only on a specific relationship. This seems a strong advantage for explicit-encoding since the other three layouts are commonly complained about because of their limited scalability. For this reason, explicit-encoding was favored by researchers when dealing with small screen space or a large number of visualizations. However, the scalability of the rest seems to depend on other factors such as screen space availability and visual representation complexity, leading to the consideration between space efficiency and visual interference.

Effectiveness in Recognizing a Relationship ($E > S \approx IJ \approx CJ$). Researchers commonly claimed that recognizing a specific relationship is most effective with explicit-encoding because it directly calculates and represents the relationship for people. Between the rest, though the general consensus is that shorter distance between comparison targets is more effective, we found chart-wise juxtaposition is sometimes more effective in chart-wise tasks compared with item-wise juxtaposition [53, 94]. Therefore, their effectiveness may split depending on what relationship people are dealing with.

Familiarity ($CJ > IJ \approx S > E$). Although it may depend on the visualization types used, chart-wise juxtaposition seems to provide the most familiar visualization to people because it does not require any significant modification to individual visualizations. Between item-wise juxtaposition and superposition, neither seems to entirely outperform the other as we find both the familiar and unfamiliar examples for each layout: Grouped bar charts and multi-class scatterplots can be considered as familiar visualizations of using item-wise juxtaposition and superposition, respectively, while variants of heatmaps [148] and node-link diagrams [2] as the unfamiliar ones. Explicit-encoding is likely to provide the least familiar outcomes because it frequently employs novel visual representations with data aggregation, which is known to be unfamiliar to InfoVis novices [41].

Supporting Other Types of Tasks ($J > IJ > S > E$). Because visual analytics involves performing a series of multiple tasks, the importance of supporting other types of tasks, as well as comparison tasks, is emphasized by many researchers. The consensus in this respect is that explicit-encoding is least effective since it generally eliminates the original visualizations. On the other hand, chart-wise juxtaposition is commonly claimed to support general tasks the best by separately showing individual visualizations. Among the two, because of the visual interference in superposition, item-wise juxtaposition is likely to provide more effective support for the general tasks [106].

3.3.6 Approaches to Overcome the Concerns

To develop deeper insights of the comparative layouts with diverse design options, we discuss researchers' previous attempts to overcome the concerns of each layout.

Chart-wise Juxtaposition

We found four main approaches for chart-wise juxtaposition to overcome its limited scalability and ineffectiveness in comparison tasks.

Using Hybrid Layout. Explicit-encoding is frequently used to complement chart-wise juxtaposition [22, 45, 61, 65, 133]. We identified two major purposes of this approach: (1) assisting to couple the corresponding visual elements and (2) improving the effectiveness in the recognition of difference. For example, egoComp [78] used lines connecting visual elements in multiple visualizations "to decrease the user's memory cost." Heimerl et al. [45] suggested explicitly showing bin boundaries in multi-class scatterplots to “[h]elp with mapping bins across different plots.” To address the difficulty in comparing a large number of heatmap visualizations in chart-wise adjacent arrangements, BayesPiles [133] allowed people to select a reference heatmap to temporally color-encode differences (i.e., subtraction values) in the rest of the matrices. Results from user studies [92, 118] support the effectiveness of a hybrid layout, as using explicit-encoding overlays with chart-wise and item-wise juxtaposition in bar charts and node-link diagrams showed better performance compared with solely relying on the juxtaposition layouts.

Shortening Distance. Juxtaposing visualizations or visual elements as close as possible is one of the simplest but effective methods. A body of studies showed empirical evidence that comparison is easier when visual representations are closer together [70, 99, 121]. We identified four studies that explicitly mentioned using similar approaches [15, 118, 124, 125]: “When the two stimuli are far away from each other, the subject has to frequently move the eyes to switch the focus. Therefore, ... we have placed the stimuli as close to each other as possible” [15]. With user interaction, Tominski et al. [125] allowed people to crop and bring the rectangular part of a visualization close to the area to which they want to compare it. We also found two studies that used item-wise juxtaposition for this purpose; for example, Srinivasan et al. [118] “opted to use a grouped bar chart instead of a concatenated bar chart (bar charts with chart-wise juxtaposition) since comparisons are likely to be more accurate with no distracting bars in between corresponding values.” In a geographical visualization, CompaRing [124] brings a few regions of comparison candidates near a reference region upon user selection. Study results support the effectiveness of item-wise juxtaposition [92, 94, 106] in enhancing comparison performance in terms of time and accuracy, especially in item-wise comparison tasks.

Maintaining Consistency. Gleicher et al. [37] mentioned the importance of maintaining the consistency of visual properties in chart-wise juxtaposition to minimize cognitive burden. This is relevant to consistency management in multiple coordinated views [100], such as determining whether to use shared or independent data domains and ranges on the screen for individual visual channels (e.g., color, size, and the x and y axes). Likewise, Kim et al. [61] mentioned, "[Keeping visualizations consistent] seems to be particularly useful for juxtaposition because they provide a common context to link the data instances ..." Examples include arranging categories in the same order between heatmaps [146] or using a constant height for all visualizations [46]. We also found that almost all studies that employed chart-wise juxtaposition used this approach by using a constant color scheme [124], size [125], or the x and y axes [145].

Filter. The number of items or visualizations being compared simultaneously is known to determine the difficulty in comparison tasks [36]. For example, CompaRing [124] automatically selects a few number of comparison targets to reduce the complexity, and Zaman et al. [144] proposed "subtractive encoding," which removes common nodes and edges from network visualizations to highlight the differences.

Superposition

We discuss two approaches to alleviate the main drawback of superposition, visual interference.

Using Clutter Reduction Methods. To manage the visual interference, clutter reduction methods can be employed, which can be categorized into Ellis et al.'s taxonomy of clutter reduction techniques based on literature survey [31]. For example, Dasgupta et al. [27] aggregated multiple lines as a band to prevent them from being a "spaghetti plot." Many studies controlled the transparency [125] or size [2] of visual elements, while filtering visual elements [144] is also a popular method. Other methods include jittering or adding offsets along axes in line charts [26] and node-link diagrams [92].

Using Hybrid Layout. Although not commonly suggested, complementing superposition using explicit-encoding seems promising to overcome the visual interference and further enhance its performance in comparison tasks. For example, inspired by natural behaviors with printed papers, one study [125] allowed people to peek at the summary of occluded regions through a folding interaction and found that this kind of explicit-encoding on demand complements the weakness of superposition. Similarly, VAICo [109] used explicit-encoding in superposed images to summarize and show the clusters of inconstant regions with user interactions. Another result shows that highlighting common or unique nodes and edges in superposed node-link diagrams outperformed a single layout with few exceptions [92] and were preferred by professionals [106].

Approaches for Other Layouts

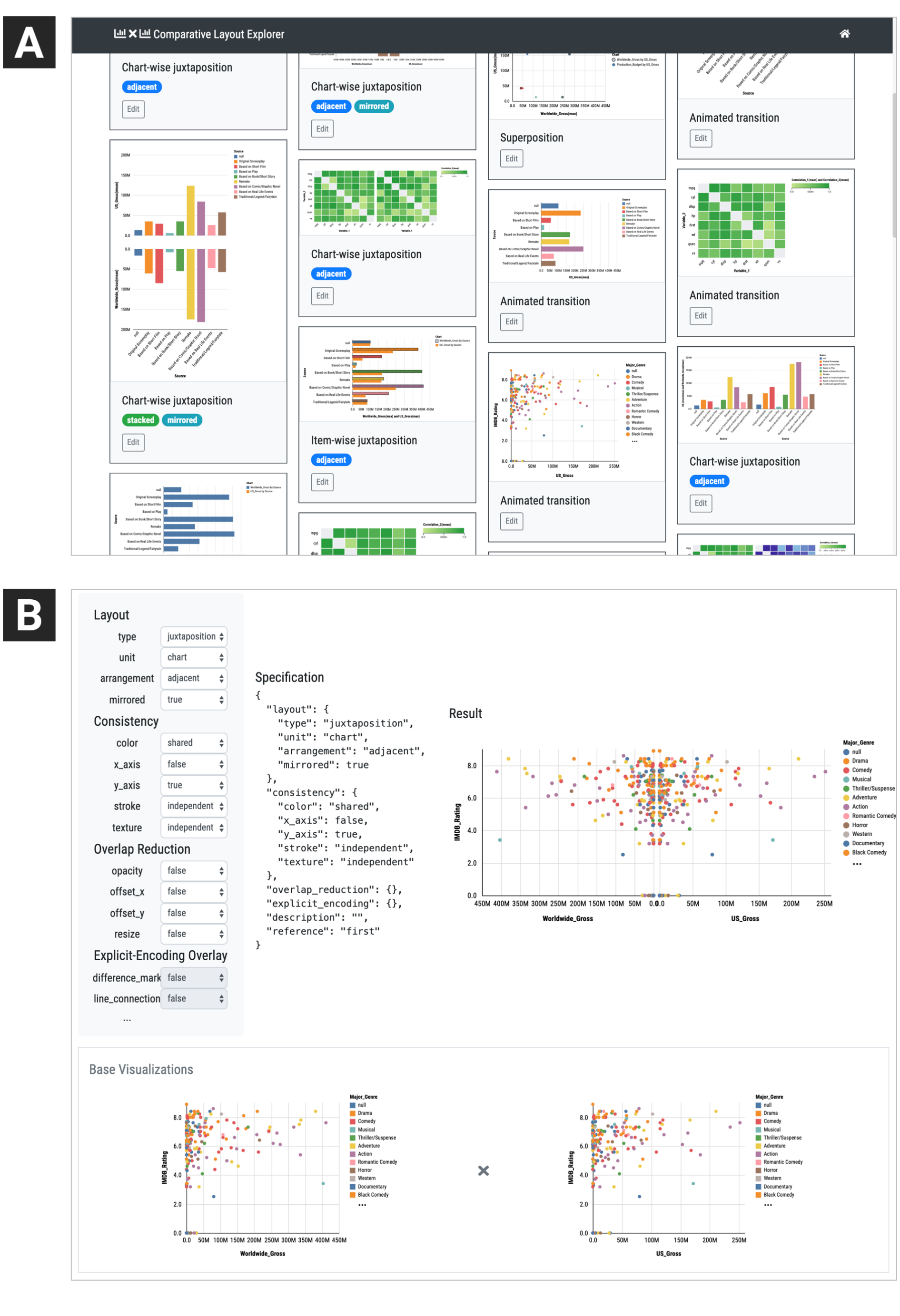
In explicit-encoding, researchers used hybrid layouts to complement the weaknesses of explicit-encoding (i.e., decontextualization and unfamiliarity). One study [85] discussed this issue and suggested using additional layouts as a remedy: "To avoid decontextualization using only explicit-encoding ..., we also use juxtaposition." A similar approach was evaluated in a study [118] that using a single explicit-encoding chart showed least preference by the unfamiliarity, but when used with an item-wise juxtaposed visualization, the preference became the best compared to other variants of bar charts. For animated transition, the use of staged changes between spatial locations is advocated, as the animation often confused people when transition between two visualizations with a large amount of difference took place [106].

3.3.7 Comparative Layout Explorer

To better help designers more systematically explore the design options of comparative layouts with interactive examples (Table 3.2), we designed and implemented a web-based interactive visual exploration tool, *Comparative Layout Explorer* (Figure 3.3). The system shows diverse designs that are observed in the literature review (Figure 3.3A). People can interactively change the layout in heatmaps, bar charts, and scatterplots based on a comparative layout specification (Figure 3.3B left), which is designed to specify the comparative layouts. Based on the specification, people can select one of three comparative layouts (i.e., juxtaposition, superposition, and explicit-encoding) and determine the diverse ways of arranging visualizations in juxtaposition: the unit of comparison targets (i.e., chart or visual element), different arrangements (i.e., adjacent, stacked, diagonal, animated), and the use of mirrored arrangements. Because visual consistency and visual interference are important factors for the comparative layouts according to our survey results, we allow users to configure them, such as using shared, independent, or distinct color palette for individual juxtaposed bar charts or using the different size of cells in superposed heatmaps.

3.4 Discussion

We offer practical implications for comparative layouts by suggesting actionable guidelines and revealing promising directions for future research. To organize the insights in a more systematical and precise manner, we reviewed and analyzed the 104 target papers, as well as the eight papers with quan-



Layout

type: juxtaposition
unit: chart
arrangement: adjacent
mirrored: true

Consistency

color: shared
x_axis: false
y_axis: true
stroke: independent
texture: independent

Overlap Reduction

opacity: false
offset_x: false
offset_y: false
resize: false

Explicit-Encoding Overlay

difference_mark: false
line_connection: false
...

Specification

```
{
  "layout": {
    "type": "juxtaposition",
    "unit": "chart",
    "arrangement": "adjacent",
    "mirrored": true
  },
  "consistency": {
    "color": "shared",
    "x_axis": false,
    "y_axis": true,
    "stroke": "independent",
    "texture": "independent"
  },
  "overlap_reduction": {},
  "explicit_encoding": {},
  "description": "",
  "reference": "first"
}
```

Figure 3.3: An interactive visual exploration tool for exploring the design space of the comparative layouts.

titative user studies (Table 3.4), using our classifications of the comparative layouts (e.g., chart-wise and item-wise juxtaposition) and study tasks (e.g., local and global comparison tasks).

3.4.1 Guidelines for Comparative Layouts

We suggest six actionable design implications for the comparative layouts.

Use item-wise juxtaposition for item-wise comparison

When looking into the studies with item-wise tasks [92, 94, 106, 118] (i.e., comparing visual elements in visualizations) chart-wise juxtaposition has ***never*** outperformed any other layouts in terms of accuracy, and it has ***barely*** outperformed in performance time. Considering the diverse factors used in the studies (e.g., visualization types, stimuli complexity, data size and amount of difference), these consistent results give a very strong implication that if detecting local differences is the main task, designers must alternatively use the item-wise juxtaposition. This implication align with other existing studies [70, 99, 121], but we confirm it again in the context of comparative layouts by categorizing tasks in terms of comparison targets (i.e., chart and visual elements).

If chart-wise juxtaposition is inevitable, provide landmarks

In the study results for item-wise tasks [81, 92, 94, 106, 118], we found a few exceptions where chart-wise juxtaposition showed comparable results to that of item-wise juxtaposition or superposition. The first case is when target visual elements are highlighted to people (explicit-encoding) so that people did not have to manually link them [92]. The second case is when dealing with geographical visualizations of showing dense regions so that some kinds of landmarks already existed, for example, buildings and roads, which people can use when identifying the corresponding visual elements [81]. Therefore, it is desirable to provide landmarks using grid or reference lines or further using explicit-encoding for highlighting to enhance the performance to some extent; however, please note that providing landmarks in the chart-wise juxtaposition did not made dramatic performance improvements to outperform item-wise juxtaposition and superposition.

Avoid blending colors for superposition

According to an observation study, using superposition in heatmap visualizations resulted in less effective comparison because people had difficulty distinguishing the blended color of cells [125]. Consistent to the observation, we found only a few examples of using superposition for heatmaps. To prevent the blending problem, designers can use one of six alternative methods that we discovered in our review. For comparing a pair of heatmaps, first, designers can simply use glyph visualizations [125], such as encoding the radius of circles rather than their color. Second, if two quantitative values are orthogonal (e.g., value and uncertainty), designers can consider using different color channels, such as, hue and saturation, following a successful design in uncertainty visualization [22]. Third, superposing heatmaps with different cell sizes can be an effective design for comparison tasks as several studies showed [2, 148] (Figure 3.2L). Fourth, instead of superposition, variants of item-wise juxtaposition can be used with stacked or diagonal arrangements [2, 148] (Figures 3.2J and 3.2K). Lastly, when the number of visualizations become larger, weaving techniques [45] or using explicit-encoding to reveal accurate difference in chart-wise juxtaposed heatmaps [133] can be used.

Avoid solely using explicit-encoding

Explicit-encoding seems to be the most delicate layout, which has strong advantages and strong weaknesses at the same time. Although its effectiveness in recognizing the predefined relationship was advocated by many researchers, others also suggest strong drawback. One strong drawback is the unfamiliarity, which can affect InfoVis novices in learning and interpreting visualizations [41]. Explicit-encoding is commonly received low preference to InfoVis novices [106, 118] and often showed poor performance by the unfamiliarity [77]. Moreover, by the decontextualization, it is often criticized by professionals in the real-world scenarios [27, 106], which reflects the weight of drawbacks that explicit-encoding has. As many researchers gave strong reasons for using explicit-encoding in their paper (e.g., perceptual advantages [133] or scalability [149]), we think explicit-encoding should be used when its advantages are certain and surpass its diverse shortcomings. One such example would be using explicit-encoding for alleviating perceptual distortions in superposed line charts such as Playfair's charts [126].

When explicit-encoding is necessary, use a hybrid layout

According to our review, hybrid layouts seem to well complement the disadvantages that a single layout has. Using it was the common approach for individual comparative layouts to overcome their weaknesses, and the hybrid layout was one of the most frequently used layout in our target papers. Moreover, all the user studies (four out of eight) that used hybrid layouts showed some kinds of superior performance with the layouts compared with a single layout, such as effectiveness in detecting and measuring local changes [92, 106, 109], high preference [106, 118], and better scalability [109]. The only user study [118] that compared explicit-encoding with and without another layout well explain the ability of the hybrid layout for complementing other layouts: Although solely using explicit-encoding (Figure 3.2C) was least preferred by people, using it with familiar visualizations (Figure 3.2H) made it most preferred while showed the best performance with the comparable results with independent explicit encoding. Therefore, we think that to protect the comparative visualizations from the strong weaknesses that explicit-encoding have, designers should consider using other layouts together.

Refrain from using animation for large difference

One study showed that animated transition showed best performance for detecting small difference in item-wise comparison, outperforming all other layouts (i.e., chart-wise and item-wise juxtaposition) [94]. However, its performance seems very sensitive to tasks, visualization types, and visual complexity. For example, in chart-wise tasks such as identifying max correlation [94] and structural change [92], the performance became weaker. Moreover, large amount of changes between two node-link diagrams [106] confused people, leading to poor task performance in accuracy. As a remedy, designers can consider using staged animation [44], which was helpful for large changes. However, we still identify many unexplored areas for animated transition in visual comparison tasks (e.g., task types, visual representations, and data complexity). As animated transition showed relatively large performance variations across different designs, designers should use animated transition with care and refrain from using it for detecting large difference.

3.4.2 Promising Directions for Future Research

Researching Human Factors in Chart-wise Comparison

As we lack empirical results for the performance of chart-wise comparison tasks (two out of eight papers), exploring the comparative layouts with diverse chart-wise tasks seems a promising direction to expand our understanding about the layouts. When we looked into the study results with item-wise comparison tasks, we were able to find relatively consistent results among the comparative layouts. However, it seems that for chart-wise tasks, the task performance is much more sensitive. For example, as the authors well demonstrated, using mirrored and adjacent chart-wise juxtaposition showed best performance in correlation tasks [94], but, for comparing mean of individual visualizations [53], stacked arrangement showed best performance. As recent work suggested [53], different perceptual heuristics seem to greatly influence the performance, resulting in varying performance by target relationships (e.g., correlation, range, mean) or visual representations (e.g., bars or lines).

Investigating the Effectiveness with Varying Difference

In our review, one of the factors that researchers most frequently discussed for their designs was the amount of difference in terms of size or complexity. For example, chart-wise juxtaposition is generally regarded as least effective for detecting a small difference because of the longer distance between visual elements. However, recent study results [53, 94] suggested that the performance might depend on what kinds of small difference users are dealing with, either a global or a local difference, as chart-wise juxtaposition performed better than item-wised juxtaposition for a certain task. As we find none of user studies in our survey directly confirmed these aspects by varying size or complexity of difference, it looks worth-exploring research topic for the comparative layouts.

Investigating the Scalability of Comparative Layouts

Most user studies (seven out of eight) focused only on one-to-one comparison. However, in the real world, more than two visualizations are frequently compared together [36]. In research papers, juxtaposition and superposition are considered to suffer from the limited scalability, compared to explicit-encoding [61, 81, 125, 133, 134, 149]. Therefore, although an independent use of explicit-encoding showed worst performance for comparing only the small number of visualizations in a study [118], it might show opposite results when the number of visualizations increases to some extent. To develop a better understanding of the comparative layouts in the real world, it seems promising to investigate the ability of the comparative layouts in terms of scalability.

3.5 Summary

In this chapter, we presented a systematic review of 104 research papers to better understand the three comparative layouts for visual comparison: juxtaposition, superposition, and explicit-encoding. Combining and systematizing the insights previously gained in the wild, we offered implications of using the comparative layouts as well as performing future work. We explored the diverse aspects of the comparative layouts, including the advantages and concerns of each layout, the approaches to overcome the concerns, and the trade-offs between them. Based on eight papers with quantitative user studies, we proposed six actionable guidelines. Finally, we revealed the unexplored research area to present promising future directions.

Chapter 4

Understanding How InfoVis Novices Compare Visual Encoding Recommendation

This chapter[1] presents the result of a qualitative user study (N=18) using a think-aloud protocol with InfoVis novices to understand the effectiveness of graphical previews, animated transitions, and textual descriptions for describing visual encoding recommendation.

4.1 Motivation

The InfoVis community is paying more attention to non-expert users who are unfamiliar with either visual representations or visualization construction processes. Among the most prominent research and development efforts in this regard is visual encoding recommendations [141] for InfoVis novices. Recommended Charts in Microsoft Excel [32] and Show Me in Tableau [120] are typical examples of visualization interfaces for recommending visual en-

[1] The preliminary version of Chapter 4 was published as a journal article [68] in Computer Graphics Forum of Wiley Online Library and also presented in EuroVis 2019.

coding alternatives based on user-selected data fields. With the recent evolution of data analysis techniques such as machine learning and deep learning, recommendation models can become even more effective, for example, by using the ranked effectiveness of visual encodings from visual perception experiments [90].

In contrast to the actively researched analytic side of visualization recommendations, research on user interface designs for more effective and understandable depictions about the suggested visual mappings has received relatively less attention in the InfoVis community. Most recommendation systems predominantly rely on graphical previews to describe alternative visual encodings [30, 32, 38, 58, 131, 142, 143]. However, because InfoVis novices are known to have difficulties in understanding visual encoding and representations in general [41], we cannot expect novices to fully understand suggested visual encodings with the graphical previews. Misunderstanding the suggestions might hider novices from producing the visual encodings they envision. To facilitate novices' learning about new visual encodings, Grammel et al. [41] suggested using in-depth textual descriptions to explain about visual encodings such as the advantages and disadvantages of using new visual encodings. However, the effectiveness of such alternative methods for describing the recommended visual encodings (e.g., in-depth textual descriptions) have not been explored in previous studies.

As an initial step toward understanding the effectiveness of different representation methods for visualization recommendations, we conducted a qualitative user study with InfoVis novices under scatterplot construction tasks. By reviewing studies related to visualization recommendations and InfoVis novices, we came up with three primary representations: previews, animated transitions, and textual descriptions. We then designed a prototype of a recommendation interface for the user study using three representation methods. Through the user study (N=18), we found that although previews remained the most preferred representations, novices still relied on textual representations. Our findings also illustrate that combining multiple representations can help users better understand the recommendations by supporting them expect and confirm about the behaviors of recommendations. Based on the findings, we present implications for designing interfaces for effective visualization recommendations for novices.

4.2 Interface

We designed a recommendation interface for our user study to understand how novices understand and choose suggested visual encodings with different representation methods during the visualization construction process.

To more efficiently identify the effects of different representation methods, we encouraged participants to actively use recommendations within the limited time of the user study. For this purpose, we assumed scenarios in which users perform goal-oriented visual analysis tasks [41] with recommendations in our prototype assisting them to accomplish sub-goals to complete the main goal.

4.2.1 Visualization Goals

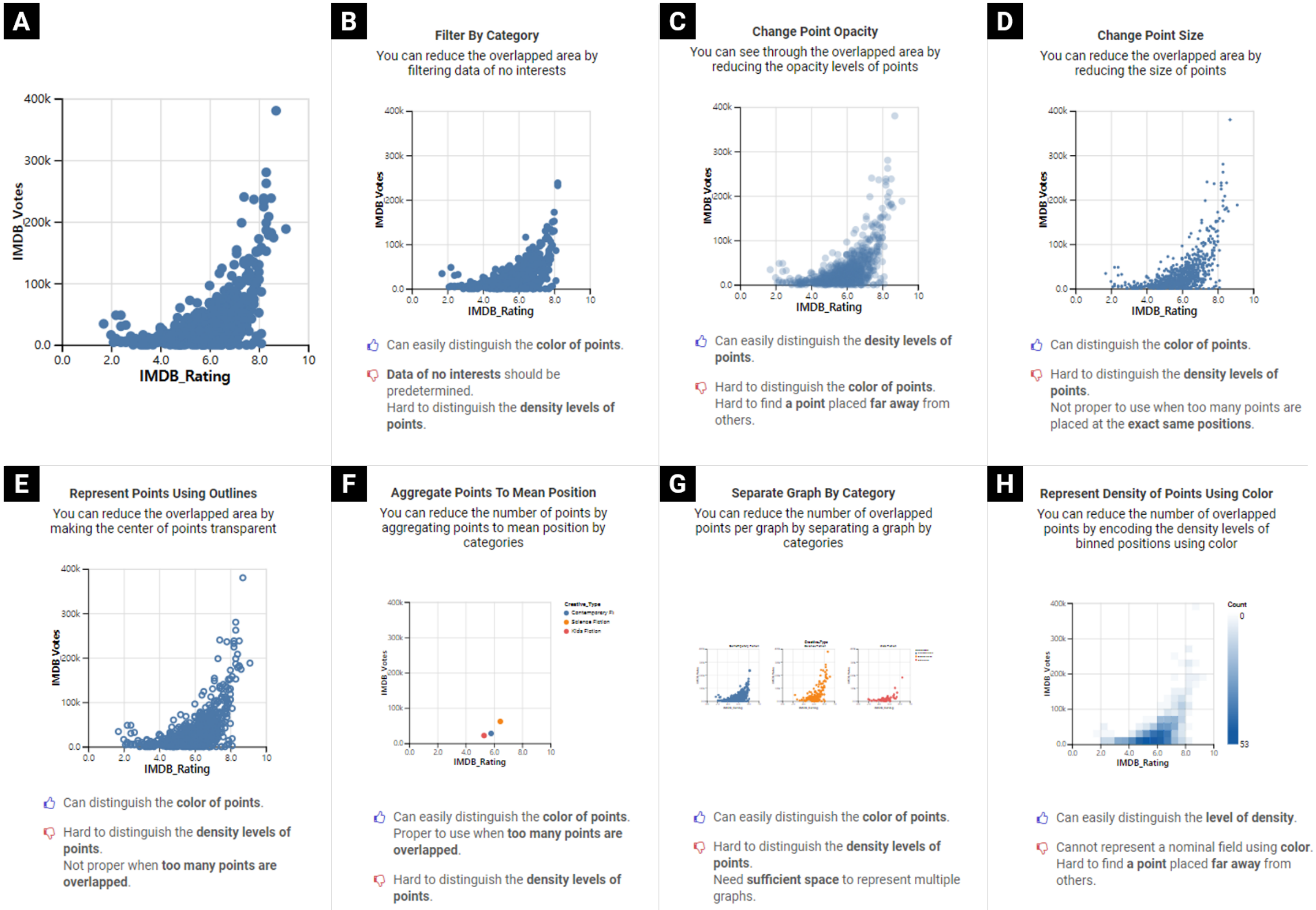
We defined the participants' main goal as constructing scatterplots to complete major scatterplot-specific analysis tasks [107]. The reasons for using the scatterplot visualization are that the scatterplot is one of the most familiar visualizations to novices [71] and that it has been widely adopted in visual exploration and recommendation systems [25, 120, 143]. We defined the users' sub-goals as alleviating over-plotting problems in scatterplots, as overdrawing in visualizations is one of the most well-known problems in the InfoVis community and is frequently addressed in InfoVis literature for novices [34]. We designed a recommendation interface for supporting the sub-goals (clutter reduction), and the main goals (scatterplot tasks) were provided as the main tasks in our study (i.e., participants had to use recommendations to complete their tasks in the study).

4.2.2 Recommendations

We designed seven scatterplot clutter reduction strategies for visualization recommendations in our prototype by referring to the clutter reduction taxonomies [31, 34] (Figure 4.1): (B) *Filter By Category*: remove points of no interests; (C) *Change Point Opacity*: change the level of opacity to see through the overlapped area; (D) *Change Point Size*: re-size points to reduce the overlapped area; (E) *Represent Points Using Outlines*: remove fill color of points to reduce the overlapped area; (F) *Aggregate Points To Mean Position*: show mean values of each category to reduce the number of points in the display; (G) *Separate Graph By Category*: divide graphs to reduce the number of points per scatterplot; and (H) *Represent Density of Points Using Color*: show density by binned area rather than displaying individual points.

4.2.3 Representation Methods for Recommendations

By reviewing studies related to recommendation systems [143] and InfoVis novices [41, 44, 105], we designed three representation methods to describe each of the seven recommendations to support novices in understanding recommendations and their usefulness: Preview, Animated Transition, and Textual Description.



B Filter By Category

You can reduce the overlapped area by filtering data of no interests

👍 Can easily distinguish the **color of points**.

👎 **Data of no interests** should be predetermined.
Hard to distinguish the **density levels of points**.

C Change Point Opacity

You can see through the overlapped area by reducing the opacity levels of points

👍 Can easily distinguish the **desity levels of points**.

👎 Hard to distinguish the **color of points**.
Hard to find **a point** placed **far away** from others.

D Change Point Size

You can reduce the overlapped area by reducing the size of points

👍 Can distinguish the **color of points**.

👎 Hard to distinguish the **density levels of points**.
Not proper to use when too many points are placed at the **exact same positions**.

E Represent Points Using Outlines

You can reduce the overlapped area by making the center of points transparent

👍 Can distinguish the **color of points**.

👎 Hard to distinguish the **density levels of points**.
Not proper when **too many points are overlapped**.

F Aggregate Points To Mean Position

You can reduce the number of points by aggregating points to mean position by categories

👍 Can easily distinguish the **color of points**.
Proper to use when **too many points are overlapped**.

👎 Hard to distinguish the **density levels of points**.

G Separate Graph By Category

You can reduce the number of overlapped points per graph by separating a graph by categories

👍 Can easily distinguish the **color of points**.

👎 Hard to distinguish the **density levels of points**.
Need **sufficient space** to represent multiple graphs.

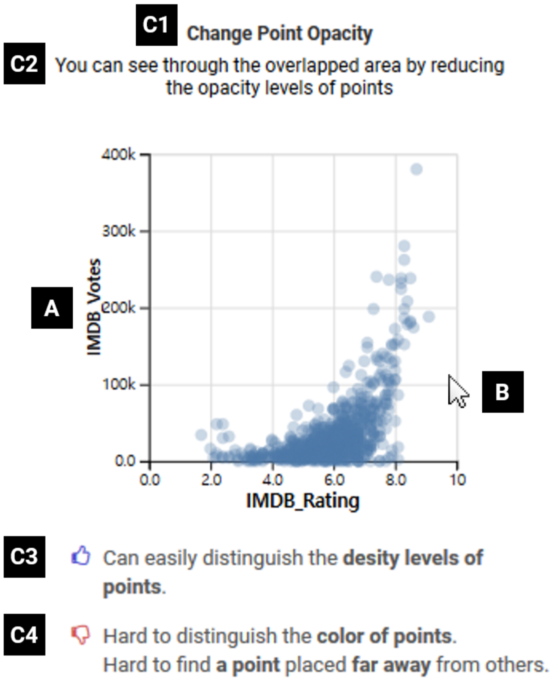
H Represent Density of Points Using Color

You can reduce the number of overlapped points by encoding the density levels of binned positions using color

👍 Can easily distinguish the **level of density**.

👎 Cannot represent a nominal field using **color**.
Hard to find **a point** placed **far away** from others.

Figure 4.1: The seven types of recommendations (B-H) in our recommendation interface for alleviating over-plotting problems in the (A) specified scatterplots.



Change Point Opacity

You can see through the overlapped area by reducing the opacity levels of points

Can easily distinguish the **desity levels of points**.

Hard to distinguish the **color of points**.
Hard to find **a point** placed **far away** from others.

Figure 4.2: Three representation methods for a visualization recommendation: (A) Preview, (B) Animated Transition on mouse hover, and (C1-4) Textual Description.

Preview

Preview, the most widely used method to represent visualization recommendations in existing tools [16, 25, 30, 32, 38, 58, 83, 108, 128, 131, 140, 142, 143] (Figure 4.2A), shows the visualization result where the suggested visual mapping is applied over the current visualization. By showing the suggested visualization result in advance, users might easily presume and compare the usefulness of recommendations as illustrated by Grammel et al. [41]. Between two types of previews, we used actual visualization results rather than thumbnails because our recommendations are data- or encoding-level suggestions, which require detailed representations.

Animated Transition

While Preview shows the result visualization as a static image, Animated Transition (Figure 4.2B) connects the gap between the current visualization and Preview by showing smooth transitions. According to previous work [44, 105], animated transitions allowed novices to better understand new visual mappings. Since InfoVis novices often confront *visual mapping* and *interpretation barriers* [41] in the visualization construction process, Animated Transition might further help users understand the behavior of new visual mappings in recommendations.

For the relatively large difference between the current visualization and Preview (i.e., *Aggregate Points To Mean Position* in Figure 4.1F), we used a staged transition [44] to help users follow the changes: Points are first colored by a default nominal field and then moved to mean positions of their categories.

Textual Description

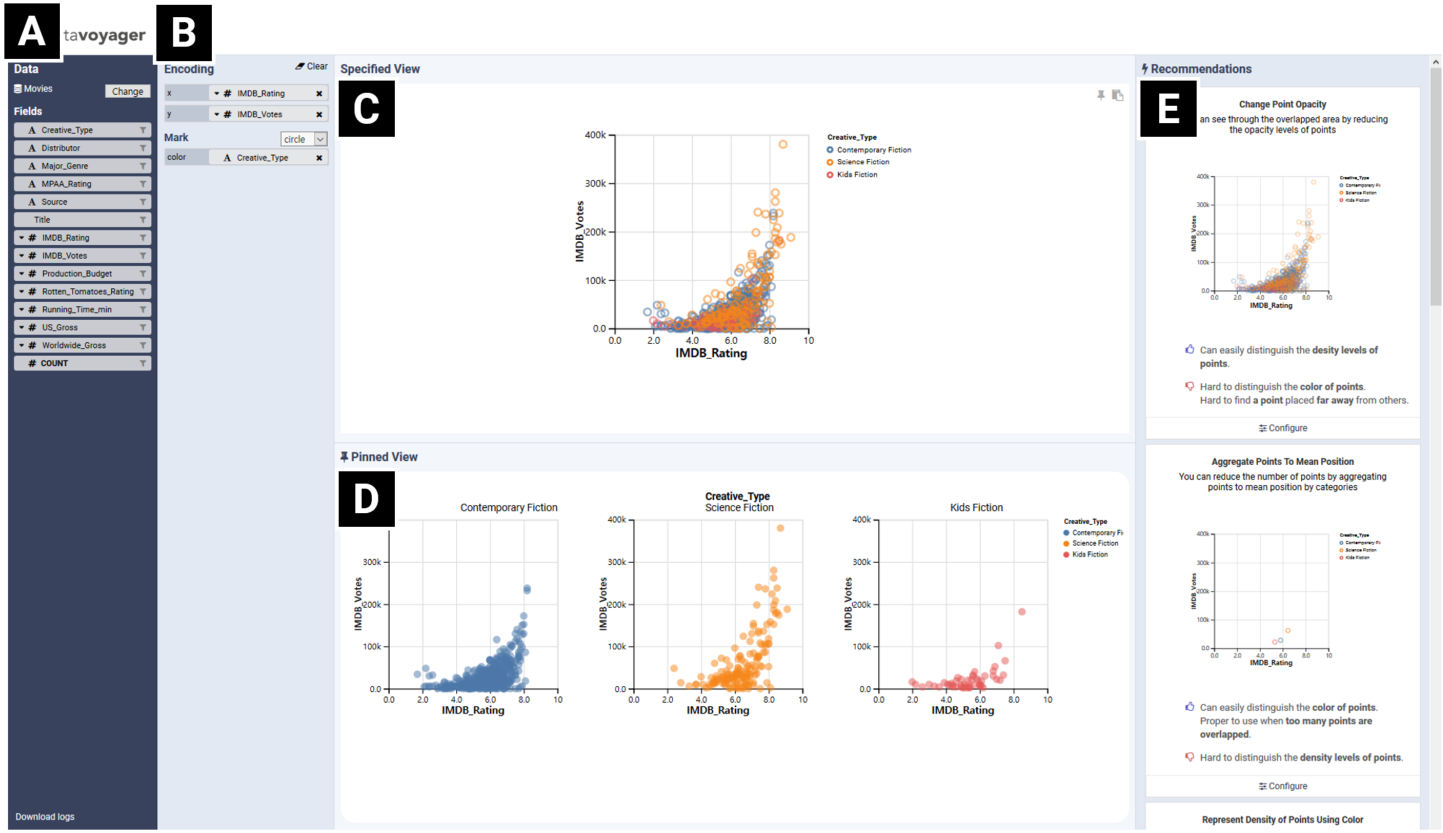
According to Grammel et al. [41], providing explanations about recommendations is important to give deeper insight. Such explanations include *what* the recommendation is about, as in [143]; *why* it is important; and what *advantages* and *disadvantages* there are. We generated the four types of descriptions in our interface (Figure 4.2C1-4). For the advantages and disadvantages, we generated descriptions based on four major criteria referring to a clutter reduction taxonomy [31]: *can show point color*, *can show overlap density*, *can show outlier*, and *is scalable to large data*.

Because the readability of textual descriptions would affect InfoVis novices' ability to understand them, we constructed and revised the textual descriptions with care to make them readable to novices. We extracted explanations about each recommendation in the literature [31, 34] and then conducted a two-hour discussion session with an InfoVis novice to create novice-friendly expressions. During this in-person interview, we reviewed four types of textual descriptions (i.e., *what*, *why*, *advantages*, and *disadvantages*) of seven recommendations sentence by sentence. The text we created was targeted to *users* rather than *designers* because we assumed that novices are more likely to view themselves as users; for example, we used "*Can **see** point color*" rather than "*Can **show** point color.*" In addition, we clarified ambiguous expressions (e.g., "*Not scalable to **large data***" had been changed to "*Not appropriate when **too many points overlap***"). We then assessed the readability of the text descriptions in a pilot study (section 4.2.5) before the main study.

4.2.4 Interface

We implemented our recommendation interface on PoleStar [143], an open-source visualization tool that allows users to construct visualizations based on a Cartesian coordinate system. The main reason for using the system is that it uses a shelf-configuration interface [39], which is one of the most widely used interfaces in existing tools such as Tableau [120], Polaris [119], and PivotTable in Microsoft Excel [32]. By using the familiar interface, we expected users might easily learn about the tool within a short training session. As we focused on constructing scatterplots, we modified PoleStar to support only scatterplots. Moreover, to encourage participants to actively use recommendations, some visual encoding features in the modified PoleStar, such as separating graphs or filtering, were hidden from PoleStar and supported only in the recommendation panels.

The overall interface of modified PoleStar with the recommendation interface is shown in Figure 4.3. The data panel shows a list of data fields (Fig-



Change Point Opacity

an see through the overlapped area by reducing the opacity levels of points

Can easily distinguish the desity levels of points.

Hard to distinguish the color of points.
Hard to find a point placed far away from others.

Aggregate Points To Mean Position

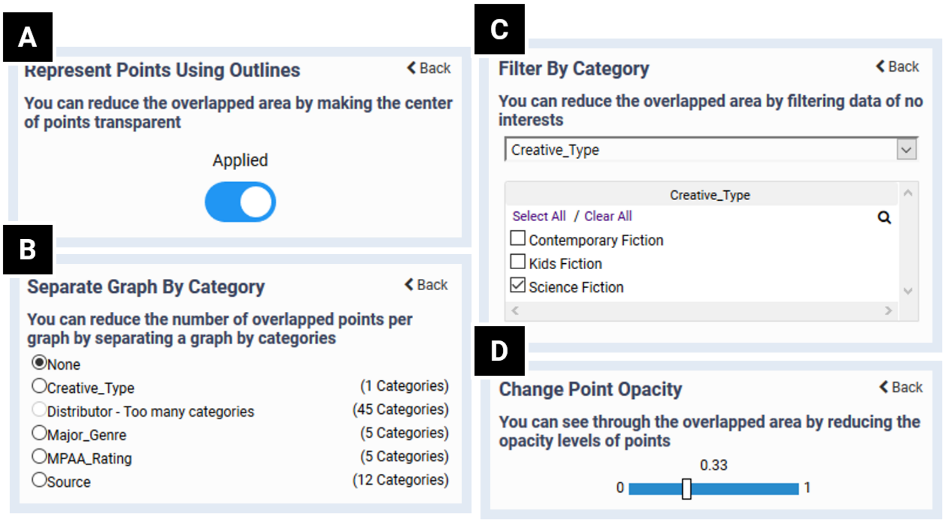
You can reduce the number of points by aggregating points to mean position by categories

Can easily distinguish the color of points.
Proper to use when too many points are overlapped.

Hard to distinguish the density levels of points.

Represent Density of Points Using Color

Figure 4.3: Overall interface of the modified PoleStar[143] with the recommendation interface: (A) data panel, (B) encoding panel, (C) specified view, (D) pinned view, and (E) recommendation interface.



Represent Points Using Outlines

You can reduce the overlapped area by making the center of points transparent

Applied

Separate Graph By Category

You can reduce the number of overlapped points per graph by separating a graph by categories

● None
○ Creative_Type (1 Categories)
○ Distributor - Too many categories (45 Categories)
○ Major_Genre (5 Categories)
○ MPAA_Rating (5 Categories)
○ Source (12 Categories)

Filter By Category

You can reduce the overlapped area by filtering data of no interests

Creative_Type

Creative_Type
Select All / Clear All
□ Contemporary Fiction
□ Kids Fiction
☑ Science Fiction

Change Point Opacity

You can see through the overlapped area by reducing the opacity levels of points

0.33

0 1

Figure 4.4: Configuration interfaces for recommendations: (A) toggle button for Represent Points Using Outlines and Represent Density of Points Using Color, (B) nominal field picker for Aggregate Points To Mean Position and Separate Graph By Category, (C) category picker for Filter By Category, and (D) slider bar for Change Point Size and Change Point Opacity.

ure 4.3A), and users can connect the fields to visual properties (e.g., x/y axis or color) in the encoding panel (Figure 4.3B). The specified view in the middle (Figure 4.3C) shows the visualization that is defined in the encoding panel. To facilitate comparison of the visualizations users construct, we enabled users to pin their visualizations to the bottom of the window (Figure 4.3D) by clicking on the Pin button.

The recommendation panel in the right-most area (Figure 4.3E) shows the seven recommendations in a gallery-style layout [41] for easy comparison between alternatives. In each recommendation, Preview and Textual Description are shown as static representations, while Animated Transition is displayed upon mouse hover on Preview. Users can apply the suggested visual mappings to the specified view after they adjust parameters (e.g., change the level of opacity for *Change Point Opacity* or select a data field and categories for *Filter By Category*, Figure 4.4C-D). For the recommendations that do not support adjustable parameters (e.g., *Represent Density of Points Using Color*), the interface shows simple toggle buttons to apply recommended visual mappings over the specified view (Figure 4.4A).

4.2.5 Pilot Study

We conducted a pilot study with six participants to evaluate the feasibility of the recommendation interface and the study design. The participants used the interface for about half an hour to solve six questions related to major scatterplot tasks [107]. After the pilot study, we improved the interface based on the participants' feedback. Firstly, we highlighted keywords in the Textual Description using font weight to improve readability (e.g., "Can see **point color**"); we did not use other highlighting methods with better pop-out effects such as a yellow background or larger font [105] because we assumed that such methods would distract users during the visualization construction process. Secondly, we changed the trigger method for Animated Transition. We initially had placed a Play button for the transition in each recommendation so that users could see the transition on demand. However, users occasionally forgot about the existence of Animated Transition during cognitively challenging tasks in the study. As we wanted to see the effects of Animated Transition during the study, we instead chose to display animated transitions when users hover the mouse over Preview. Thirdly, we empirically chose the duration of the animated transition considering participants' feedback: one second long for each staged transition, consistent with previous design guidelines (e.g., [103]). Finally, Animated Transition was initially positioned on top of the specified view but moved to the recommendation panel because some participants commented that it being placed far from Preview and Textual Descriptions somewhat confused them.

4.3 User Study

To better understand how InfoVis novices use visualization recommendations during a visualization construction process, we conducted a qualitative study on our recommendation prototype using a think-aloud protocol.

4.3.1 Participants

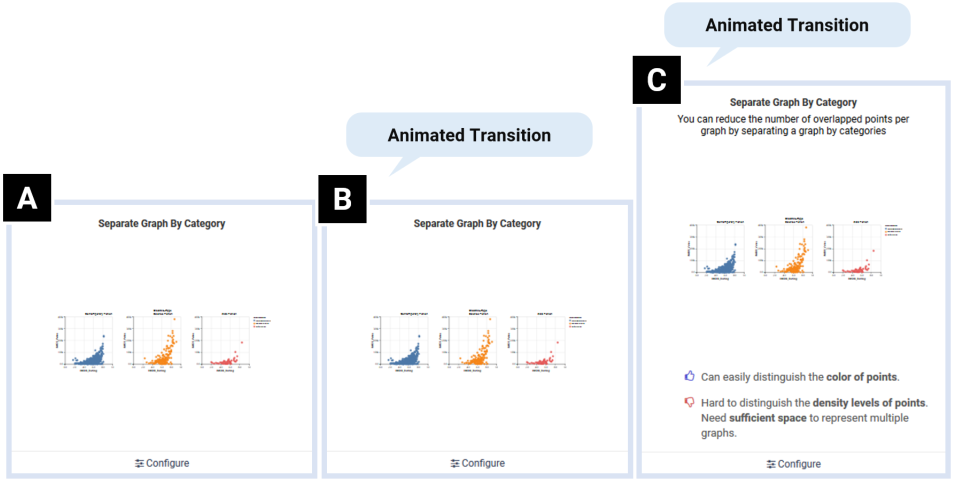
We recruited 24 participants (10 females), ages 18 to 33 years, from a university. They were self-reported to use visualization tools 4.2 times per month on average. The most frequently used visualization tool was Microsoft Excel [32] (21 participants), while a few participants also used R [101], Origin [95], MATLAB [86], and Tableau [120]. Most participants (21 participants) reported to have no prior knowledge about information visualization; only three participants were aware of InfoVis from lectures related to statistics tools (e.g., R or MATLAB) at university or at work. Participants received about $10 for their participation.

4.3.2 Interface

Participants used one of three combinations of representation methods in our qualitative user study. Three combinations of representations were designed to provide different levels of information about recommendations: (1) Preview + Title (PT, Figure 4.5A), (2) PT + Animated Transition (PTA, Figure 4.5B), and (3) PTA + remaining Textual Description (PTAT, Figure 4.5C). The main reason for providing Title (i.e., Textual Description about

ID	Questions
Q1	Is there a correlation between **US_Gross** and **Worldwide_Gross**?
Q2	Draw a scatterplot that shows a relationship between **IMDB_Score** and **IMDB_Votes**. In which area is movies most densely placed?
Q3	Find a point far away from others in a scatterplot that shows a relationship between **Production_Budget** and **Worldwide_Gross**.
Q4	Which **Genre** has the highest **Production_Budget** and **Running_Time** on average?
Q5	Of movies whose **Distributor** is either *Paramount Pictures* or *Sony Pictures*, what are the ranges of **Production_Budget** and **Worldwide_Gross**?
Q6	Of movies whose **IMDB_Scores** and **Rotten_Tomatoes_Scores** are higher than *6* and *60*, what is the most common **Creative_Type**?

Table 4.1: Six questions based on scatterplot-related visualization tasks [107] used in our study.


Animated Transition
C
Animated Transition
B
A
Separate Graph By Category
Separate Graph By Category
Separate Graph By Category
Configure
Configure
Configure

You can reduce the number of overlapped points per graph by separating a graph by categories

Can easily distinguish the **color of points**.

Hard to distinguish the **density levels of points**. Need **sufficient space** to represent multiple graphs.

Figure 4.5: Three combinations of representation methods used in our study: (A) Preview + Title (PT), (B) PT + Animated Transition (PTA), and (C) PTA + remaining Textual Description (PTAT).

what) for all conditions was that most of the encoding-level recommendations (e.g., [142, 143]) use previews with simple titles, possibly because novices are unlikely to fully understand the small difference between the specified view and Preview.

The layout of the modified PoleStar was fixed across all participants, and the width of recommendation panels was 410 px. We limited the space of the recommendation panels to reflect common recommendation interfaces that show only a few recommendations at once, making users interact with scroll views (e.g., [58, 142, 143]). In the study layout, only two recommendations were visible for the PTAT condition (all methods together), while other conditions showed an additional recommendation (i.e., three recommendations). We randomly ordered seven recommendations across all participants to prevent order effects.

4.3.3 Tasks and Datasets

We designed six questions (Table 4.1) based on scatterplot-related visualization tasks [107], which are constructed by surveying scatterplot-specific analysis scenarios in InfoVis literature, and are frequently employed in controlled user studies as study tasks (e.g., [20, 62]). Each question was designed to reflect either a browsing-related task or an aggregate-level task [107] (i.e., Q1 and Q6 are browsing-related tasks while Q2-Q5 are aggregated-level tasks). We had not considered object-centric tasks because they are less related to over-plotting problems.

When visualizing the prepared dataset with scatterplots, over-plotting problems made it difficult to answer four of the questions (all questions except Q1 and Q3) without alleviating the problems. Therefore, the participants had to use the recommendation interface to answer the questions.

For a training session, we prepared an SAT score dataset that consisted of scores and grades of 143 students in five subjects and some demographic data (i.e., gender, region, education level). For the main task, we used a movie dataset [142] that contained classifications of 746 movies (e.g., genre, creative type, MPAA rating, and distributor) and their budgets, worldwide/US gross, playtimes, review scores and the number of votes.

4.3.4 Procedure

After signing a consent form and completing a pre-study questionnaire, participants were introduced to the overall procedure and the task for about five minutes. Because the focus of our study is to explore how participants understand and use unfamiliar recommendations instead of unfamiliar visualizations themselves, participants were also introduced to a scatterplot visualization to make them get familiar with it. They then had a training session during which they were introduced to the interface and practiced constructing scatterplots using the interface to answer six practice questions based on the SAT score dataset. The participants had to understand about the seven recommendations only with the given interface; the experimenter did not explain any about the recommendations. By answering the practice questions, the participants became familiar with their tasks. After the training session, participants were asked to complete the main task in which they used the interface to construct scatterplots based on the movies dataset [142] to answer six questions (Table 4.1). Participants were asked to construct scatterplots that clearly show the answers to the questions by using the recommendation panel. After they constructed each scatterplot, they pinned the scatterplot, answered to the question, and moved on to the next question. Participants repeated this process until they answered the last question. We recorded the screen during the practice and main tasks. Upon answering all the questions, they were asked to complete a questionnaire that included an assessment of how much (7-point scale) each representation was helpful for understanding the recommendations and the reasons for thinking that each of them is useful or not (e.g., participants in the PTAT condition assessed all three representation methods). Then, we asked participants to think aloud about their visualization construction process by watching the recorded video before conducting an open-ended interview. Participants were allowed to rest at any time during the study. The entire study procedure took about 45 minutes on average per participant.

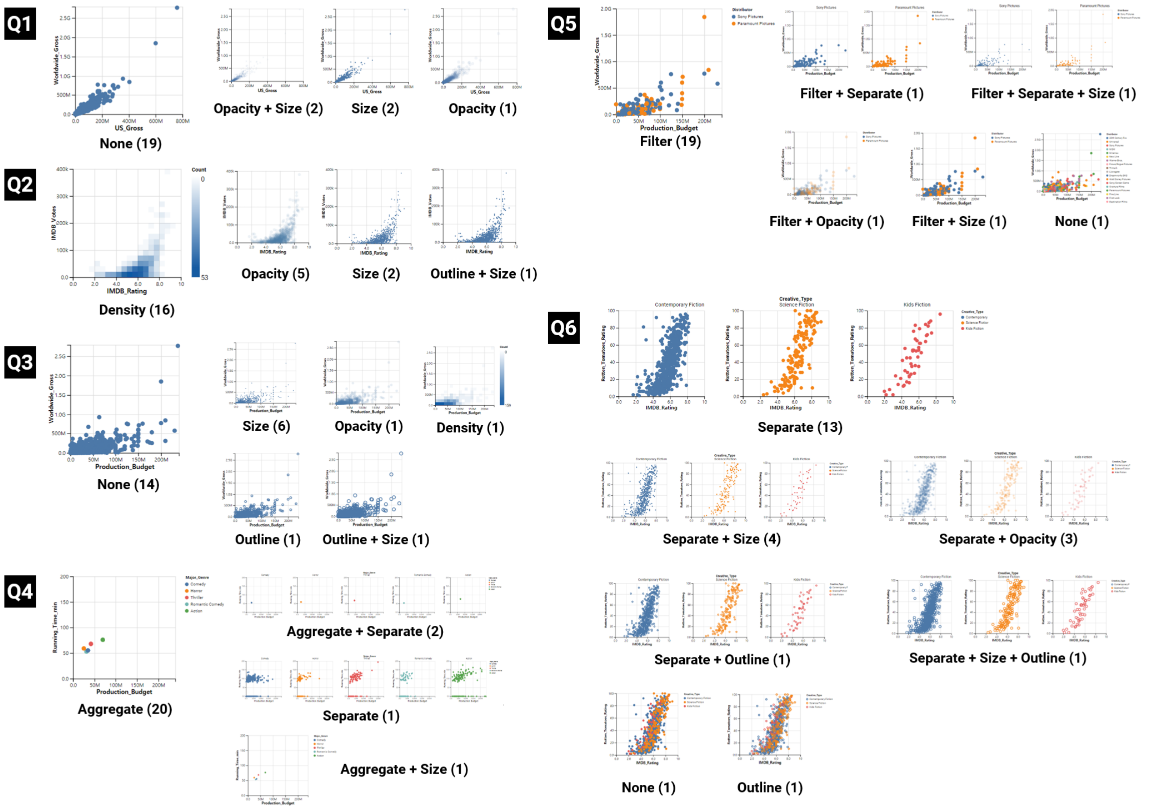


Figure 4.6: Scatterplots constructed by the participants in our study in response to Question 1 to 6. Each scatterplot is labelled with the name of recommendations which are used to make the visualization. Recommendation names are abbreviated: (1) Filter By Category, (2) Change Point Opacity, (3) Change Point Size, (4) Represent Points Using Outlines, (5) Aggregate Points To Mean Position, (6) Separate Graph By Category, and (7) Represent Density of Points Using Color. The number in parentheses represents the number of participants who constructed the visualizations.

4.4 Findings

Participants produced diverse scatterplot designs using different recommendations, more than four different designs per question (Figure 4.6). Of 144 scatterplots (24 participants x 6 questions), 35 scatterplots were constructed without using any recommendations, mostly for Q1 and Q3; 89 scatterplots were generated using only one recommendation; and the rest (20 scatterplots) were constructed using two or more recommendations together. In all cases, recommendations were used to alleviate the over-plotting problems in the scatterplots, except one participant (P12_{PTAT}) who used *Change Point Size* to make the outlier more visually salient by increasing the size (Figure 4.6 Q3-*Outline+Size* and Q5-*Filter+Size*).

4.4.1 Poor Design Decisions

Of 144 answers to the questions, two of them were incorrect: P13_{PT} reported to have read the category name incorrectly in the color legend (Figure 4.6 Q4-*Aggregate*), and P20_{PTAT} did not use *Filter By Category* because she did not understand it well, making it difficult for her to answer Question 5 (Figure 4.6 Q5-*None*). We further discuss such challenges for understanding recommendations in subsection 4.4.3. Although the rest of the scatterplot designs derived correct answers, we identified several poor design decisions. For example, the goal of Question 3 was to clearly show the outlier in the scatterplots, but some participants (8 of 24 participants) either reduced the size or opacity of points or used density plots, which unintentionally led to making the outlier hard to notice (Figure 4.6 Q3-*Change Point Opacity*, Q3-*Change Point Size*, and Q3-*Represent Density of Points Using Color*). Three participants made similar poor decisions in Q5 (i.e., Figure 4.6 Q5-*Filter+Opacity* and Q5-

Filter+Separate+Size). Of all 10 participants (8 for Q3 and 2 for Q5), only one used the PTAT condition, possibly because the Textual Description about advantages and disadvantages contained explanations about the outlier (e.g., *"Hard to find a point placed far away from others"* in *Change Point Opacity*).

4.4.2 Role of Preview, Animated Transition, and Text

The most common representation method–Preview–was reported to be most useful when understanding and selecting recommendations (5.9 out of 7) and identified as the most intuitive: *"I was able to understand recommendations at a glance by Preview"* (P5$_{\text{PTA}}$). On the contrary, Animated Transition was less helpful than Preview on average (3.9 out of 7) but still useful for understanding recommendations when the difference between the specific view and Preview is relatively large (e.g., *Aggregate Points To Mean Position* and *Separate Graph By Category*): *"[Animated Transition] was not essential but helpful when understanding large changes."* (P18$_{\text{PTA}}$). Although Preview was the most intuitive representation for most participants, a few (12.5%) said that they preferred textual descriptions. One said that *"[advantages and disadvantages] give insight about recommendations."* (P17$_{\text{PTAT}}$). This is consistent with the study result: Participants who read advantages and disadvantages (i.e., PTAT condition) barely used *Change Point Opacity*, *Change Point Size*, or *Represent Density of Points Using Color* when they had to make the outlier noticeable in Q2 and Q5. Some participants provided other reasons for preferring Textual Description: They found it hard to compare differences between previews. We might interpret this tendency by *interpretation barrier* [41], where novices are likely to confront difficulties in interpreting visualizations. Because of the barrier, some participants seemed to intensively rely on Textual Description. For example, to solve Question 2, some reported that they used density plots rather than *Change Point Opacity* simply because the title or the advantage description contained *"density."* This seems to be the one of the main reasons why *Represent Density of Points Using Color* was used much more than *Change Point Opacity*, as illustrated in Figure 4.6 Q2 (i.e., 16 times for the density plots and five for the other).

Although participants preferred a specific representation of the three methods, most reported to have used multiple methods together, as they expected and confirmed the behavior of suggested visual mappings to more clearly understand them. For example, they saw a preview and then expected the behavior of the recommendation. Whenever they had not clearly understood about the recommendation, they saw textual descriptions or animated transitions to confirm their hypothesis.

4.4.3 Challenges For Understanding Recommendations

The biggest challenges participants confronted in understanding recommendations was identifying the difference between pairs of visualizations. This includes distinguishing 1) **between the specified view and Preview** and 2) **between recommendations themselves**. For example, P20$_{\text{PTAT}}$ did not use *Filter By Category* in the study. During the interview, he said he had not clearly understood the recommendation because the difference between the specified view and Preview was subtle (Figure 4.1A and B). He had not tried to understand the recommendation clearly, and this led to never using it. Similarly, P12$_{\text{PTAT}}$ reported that whenever the previews were not distinguishable, he did not use them. Animated Transition seemed not to show the difference clearly as P20$_{\text{PTAT}}$ said, *"Fun to see, but the [visual change] of Animated Transition was subtle."*

Distinguishing between recommendations themselves also includes **comparing textual descriptions**. P6$_{\mathrm{PTAT}}$ and P12$_{\mathrm{PTAT}}$, for example, said they mistakenly thought that *Filter By Category* and *Separate Graph By Category* are the same because they contained the same keyword (i.e., *category*, Figure 4.1B and G). Moreover, P6$_{\mathrm{PTAT}}$ and P20$_{\mathrm{PTAT}}$ said it was hard to compare the descriptions of advantages and disadvantages between recommendations because some sentences are redundantly placed across a few recommendations (e.g., "Can easily distinguish the **density** levels of points", Figure 4.1C and H).

4.4.4 Learning By Doing

Six participants reported that playing with configurable parameters of recommendations (e.g., re-sizing points by a slider bar in Figure 4.4) in addition to using the three representation methods helped them understand the recommendations (i.e., *learning-by-doing* [67]). For example, P10$_{\mathrm{PTAT}}$ said he better understand *Change Point Opacity* when he adjusted the level of opacity using the slider bar (Figure 4D): *"[The] difference [between the specified view and Preview for Change Point Opacity] was subtle, but I understood [Change Point Opacity] by adjusting it."* Similarly, the behavior of *Filter By Category* was not initially clear to some participants because the changes between the specified view and Preview were subtle for them. However, they reported that once they adjusted and applied the recommendation, they clearly understood what it does: *"Once I configure [...], I understand it clearly"* (P21$_{\mathrm{PT}}$).

4.4.5 Effects of Recommendation Order

Figure 4.7 shows the number of times participants chose recommendations by their order during the task. Note that the seven recommendations were randomly ordered for each participant. As can intuitively be expected, the last one was least frequently selected: *"I haven't seen the density plot (the last one) when using the system"* (P11$_{\text{PTA}}$). The reason for such a trend seems to be that the participants regarded the later ones as less important; as P9$_{\text{PTAT}}$ said, *"I felt that recommendations on the bottom are less effective than the first few ones. So perhaps I skipped using the last one."* Interestingly, the number of times participants selected recommendations in the middle (i.e., 4th and 5th) dropped to some degree. P12$_{\text{PTAT}}$ gave a possible reason for this tendency: *"I think I occasionally skipped recommendations on the middle. Perhaps it is because previews looked similar to each other to me when scrolling down."* According to the feedback, making the differences more visually salient might address the problem of missing recommendations in the middle while scrolling down.

4.4.6 Personal Criteria for Selecting Recommendations

We were also interested in the criteria that participants have in their minds when selecting recommendations. Knowing the users' diverse criteria, designers might consider users' needs when designing recommendation systems. Because their task was constructing visualizations that best illustrate

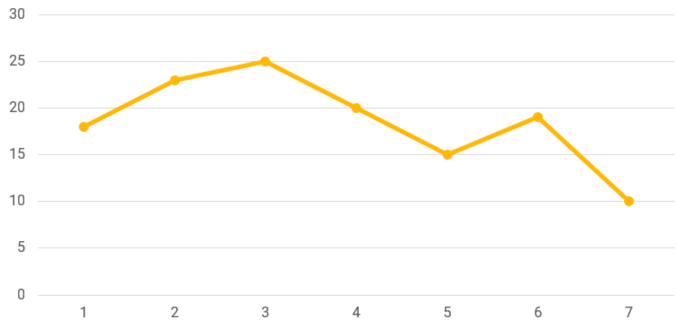


Figure 4.7: The number of times recommendations were chosen by their order during the task.

answers to the questions, all participants tried to select recommendations that make the visualization perceptually better. However, they still had options to chose between recommendations that provide similar information (e.g., density plot or *Change Point Opacity* to see the density of the overlapped area). The most frequent criterion was an aesthetic perspective (35.5% participants) followed by familiarity (12.5%). Two participants used recommendations that were more familiar to them, while one participant wanted to use unfamiliar recommendations on purpose: *"I tried to use recommendations that I have never used before like [density plots]. I wanted to learn new visualizations"* (P15$_{\text{PTA}}$). Another participant said he used recommendations that support adjustable parameters: He used *Change Point Opacity* rather than *Represent Points Using Outlines* because the former supports changing the level of opacity, while the latter did not support such an adjustable parameter.

4.5 Discussion

4.5.1 Design Implications

Based on our findings, we propose three implications for improving the design of recommendation interfaces in visualization tools.

Highlight Subtle Differences

When providing recommendations, each recommendation should be distinguishable from the others in terms of graphical previews and textual descriptions (e.g., titles), and each recommendation should also be distinguishable from the specified view. When differences are subtle, novices might have a hard time understanding the behaviors of the recommendations or might miss some of them while using the recommendation interfaces. One method to avoid subtle differences might be making the difference clearer to novices using additional visualization techniques. Using the animated transition could be one option, but in our study, some participants still found it hard to see the visual changes in transitions when the differences are relatively small (e.g., *Represent Points By Outline*). We used one second for each staged transition, consistent with previous design guidelines [103], but designers should consider increasing the duration to make animated transitions more noticeable. Moreover, several other techniques would be useful to further make the changes clearer, such as emphasizing the differences using annotation methods [102] or extending visualization techniques for visual comparison [37] to recommendation interfaces. If additional techniques cannot be used, aggregating recommendations by their visual similarities would be another possible method (e.g., clustering recommendations as in [143]).

Use Multiple Representations Together

Recommendation interfaces should combine multiple representations to support the novices' *expect-and-confirm* process. Since novices often experience *interpretation barriers* [41], a single representation would not be enough for them to clearly understand the recommendations. In such situations, seeing another representation helped users more clearly understand unfamiliar recommendations. For example, in MS Excel 2016 [32], recommended visualizations are provided with thumbnail previews. However, users might find it hard to distinguish between recommendations such as between Stacked Bar and 100% Stacked Bar only with the preview. Our findings suggest that recommendation interfaces should at least provide previews with clear titles unless rendering the actual chart is not feasible within given resources. Although previews are the most intuitive representations, novices still prefer textual descriptions because novices sometimes do not feel confident about what they have understood by previews.

Support Learning By Doing

The learning-by-doing approach [67], which is known to be useful for learning parallel coordinate plots, was also useful for understanding the behavior of recommendations during the visual construction process. Therefore, we believe visualization recommendation interfaces must support the learning-by-doing approach by giving users the opportunity to play with recommendations. In our recommendation interface, we showed adjustable interfaces (e.g., a slider bar) after users pressed a button. Possibly because of this, one participant misunderstood a recommendation and had no chance to try it. Hence, it might be more effective to make adjustable interfaces visible to users together with other representation methods (e.g., Preview), regarding the adjustable interface as one of the representation methods for describing recommendations.

4.5.2 Limitations and Future Work

Our controlled user study had several limitations in terms of external validity. First, we limited the users' visualization tasks to scatterplot clutter reductions to make the study analysis more efficient. To extend our findings to a more general visualization construction process, it would be necessary to explore representation methods with different visualizations and tasks. Second, our study prototype provided a limited number of recommendations. However, the number of recommendations can become larger in the real-world, which complicates the generation process of textual descriptions. In our study, we manually constructed the textual descriptions with care because the readability can disturb novices in the cognitively challenging tasks of the visualization construction process. State-of-the-art natural language generation (NLG) techniques [33] might help generate the descriptions in a more efficient manner, but the readability should be carefully assessed. Constructing NLG models for textual descriptions in recommendations would be a promising research direction.

We believe evaluating recommendation forms in terms of task time and accuracy is an equally promising research direction. In our study, we did not evaluate them in terms of the quantitative aspects because we wanted to let the participants use recommendations for enough time during the visualization construction tasks. We thought if participants construct visualizations with the time pressure, they might end up using only first few recommendations without sufficiently thinking about their visual encodings or ignoring to use some of the representations (e.g., textual descriptions or animated transition), which are the cases we tried to prevent for understanding the usage of each representation/recommendation. We leave the quantitative evaluation as a separate future study.

In the future, it would also be interesting to design and evaluate visualization techniques for emphasizing subtle differences between visualizations or illustrating the causality of visual changes. Analyzing novices' behaviors related to recommendation systems based on gaze patterns would be equally promising to explore. Additionally, it would be also interesting to determine the effect of other combinations of representation methods, such as using only textual descriptions or preview without animated transitions, or even additional representation methods we had not used.

4.6 Summary

In this chapter, we performed a qualitative user study to broaden the understanding of the behavior of InfoVis novices when using recommendation systems to perform scatterplot clutter reduction tasks. We designed a recommendation interface using three primary representation methods—Preview, Animated Transition, and Textual Description—and found that different representations individually and cooperatively help users understand and choose recommended visualizations. Based on the study results, we presented three design implications for designing more efficient visualization recommendation interfaces for InfoVis novices.

Chapter 5

Designing XCluSim: a Visual Analytics System for Comparing Multiple Clustering Results

This chapter[1] introduces XCluSim, a visual analytics system that supports data analysts to compare multiple clustering results.

5.1 Motivation

Since Eisen lab's Cluster and TreeView [29] popularized cluster analyses and visualizations of microarray data, cluster analysis has been widely used in the bioinformatics community. As genetic probing technologies rapidly improve in capacity and accuracy (e.g., Next Generation Sequencing), cluster analysis is playing an even more important role in the descriptive modeling (segmentation or partitioning) of the large data produced by high-throughput probing technologies. Though cluster analysis has become a rou-

[1] The preliminary version of Chapter 5 was published as a journal article [69] in BMC Bioinformatics and also presented in BioVis 2015.

tine analytic task for bioinformatics research, it is still arduous for a researcher to quantify the quality of a clustering method's clustering results.

There have been a few attempts to develop objective measures for clustering quality assessment; however, in most practical research projects, determining the quality of a clustering result is subjective and application-specific [112]. To make things even more challenging, there are a large number of clustering methods, which could generate diverse clustering results. Moreover, even an individual clustering algorithm could end up with different results depending on the clustering parameters.

Since there is no generally accepted objective metric for selecting the best clustering method and its parameters for a given dataset, researchers often have to run multiple clustering algorithms and compare different results while examining the concordance and discordance among them. Such a comparison task with multiple clustering results for a large dataset is cognitively demanding and laborious.

In this chapter, we present XCluSim, a visual analytics tool that enables users to interactively compare multiple clustering results and explore individual clustering results using dedicated visualizations.

5.2 Task Analysis and Design Goals

When performing a cluster analysis with a gene expression dataset, bioinformaticians typically follow an iterative analytics process: (1) they filter out unnecessary genes from the dataset for more focused analysis; (2) they run a clustering algorithm with the selected genes; and (3) they validate clusters in the clustering result to determine whether genes are clustered properly in the biological context. When the quality of the clustering result is not satisfactory at the validation stage, they often have to return to previous steps and run the same clustering algorithm with different parameters or run a different clustering algorithm.

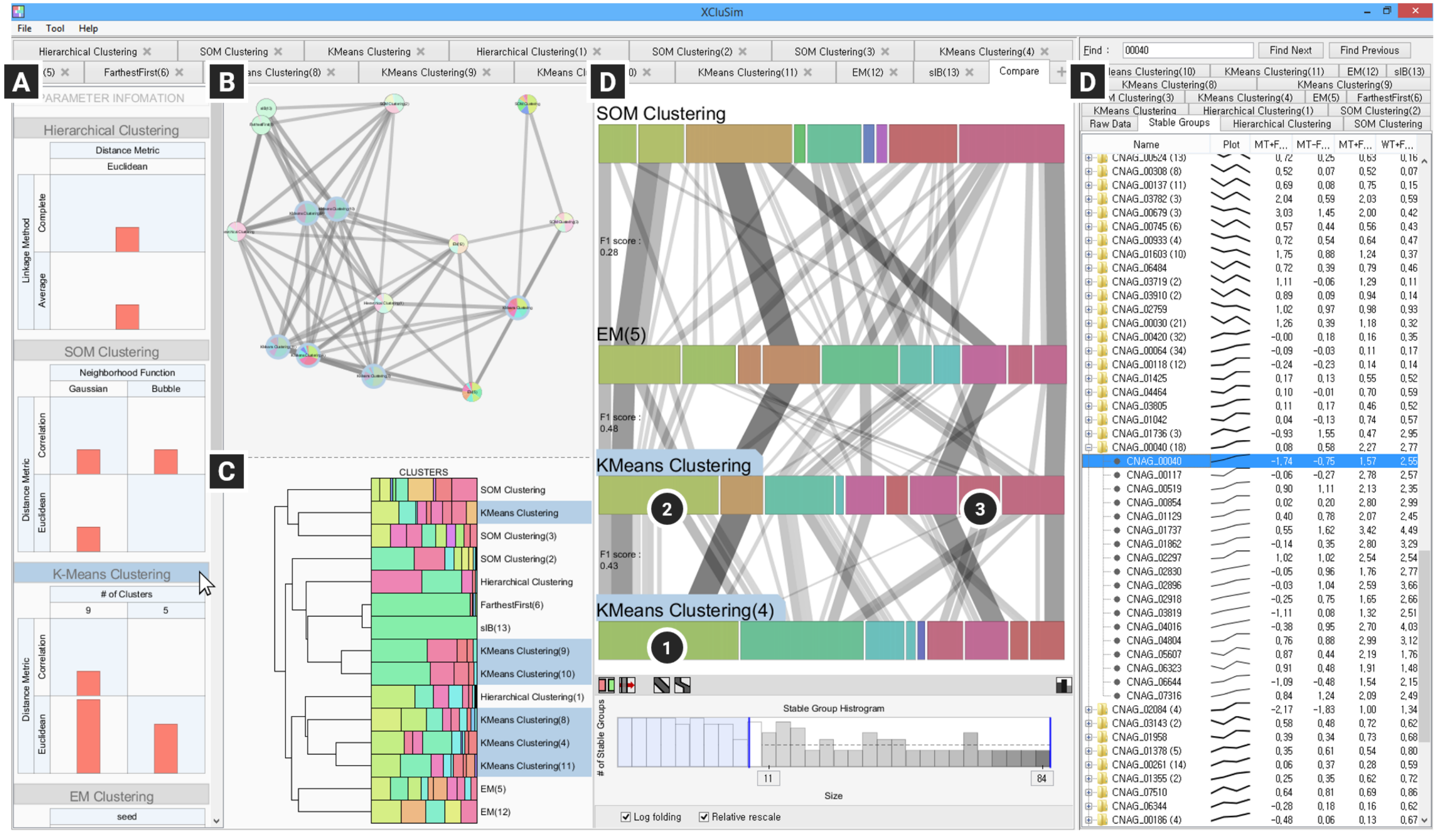
Years of close collaboration with bioinformaticians have revealed to us that they often faced challenges in this iterative analytics process. First of all, there is no flexible analytics environment that supports them through the iterative process while providing diverse clustering algorithms and keeping track of their exploration history (i.e., the sequence of the clustering algorithms and parameter settings). Moreover, it is challenging for them to effectively compare different clustering results generated during multiple iterations while investigating the quality of the results at diverse levels (i.e., clustering results level, cluster level, and gene level).

To address these challenges in the iterative process of cluster analysis, we set the following design goals for our visual analytics tool:

- To facilitate scalable visual comparison of many clustering results at diverse levels;
- To support the generation of diverse clustering results;
- To promote understanding of the characteristics of each clustering algorithm and its parameters in results;
- To provide dedicated visualizations effective for different types of individual clustering results.

5.3 XCluSim

We designed XCluSim based on the visual information seeking mantra [114]—*overview first, zoom and filter, and details-on-demand*—to better support scalable



Name	Plot	MT+F...	MT-F...	MT+F...	WT+F...
CNAG_00524 (13)		0,72	0,25	0,63	0,16
CNAG_00308 (8)		0,52	0,07	0,52	0,07
CNAG_00137 (11)		0,69	0,08	0,75	0,15
CNAG_03782 (3)		2,04	0,59	2,03	0,59
CNAG_00679 (3)		3,03	1,45	2,00	0,42
CNAG_00745 (6)		0,57	0,44	0,56	0,43
CNAG_00933 (4)		0,72	0,54	0,64	0,47
CNAG_01603 (10)		1,75	0,88	1,24	0,37
CNAG_06484		0,72	0,39	0,79	0,46
CNAG_03719 (2)		1,11	-0,06	1,29	0,11
CNAG_03910 (2)		0,89	0,09	0,94	0,14
CNAG_02759		1,02	0,97	0,98	0,93
CNAG_00030 (21)		1,26	0,39	1,18	0,32
CNAG_00420 (32)		-0,00	0,18	0,16	0,35
CNAG_00064 (34)		-0,09	-0,03	0,11	0,17
CNAG_00118 (12)		-0,24	-0,23	0,14	0,14
CNAG_01425		0,17	0,13	0,55	0,52
CNAG_04464		0,10	-0,01	0,70	0,59
CNAG_03805		0,11	0,17	0,46	0,52
CNAG_01042		0,04	-0,13	0,74	0,57
CNAG_01736 (3)		-0,93	1,55	0,47	2,95
CNAG_00040 (18)		0,08	0,58	2,27	2,77
● CNAG_00040		-1,74	-0,75	1,57	2,55
● CNAG_00117		-0,06	-0,27	2,78	2,57
● CNAG_00519		0,90	1,11	2,13	2,35
● CNAG_00854		0,02	0,20	2,80	2,99
● CNAG_01129		0,40	0,78	2,07	2,45
● CNAG_01737		0,55	1,62	3,42	4,49
● CNAG_01862		-0,14	0,35	2,80	3,29
● CNAG_02297		1,02	1,02	2,54	2,54
● CNAG_02830		-0,05	0,96	1,76	2,77
● CNAG_02896		-0,03	1,04	2,59	3,66
● CNAG_02918		-0,25	0,75	1,65	2,66
● CNAG_03819		-1,11	0,08	1,32	2,51
● CNAG_04016		-0,38	0,95	2,70	4,03
● CNAG_04804		0,76	0,88	2,99	3,12
● CNAG_05607		0,87	0,44	2,19	1,76
● CNAG_06323		0,91	0,48	1,91	1,48
● CNAG_06644		-1,09	-0,48	1,54	2,15
● CNAG_07316		0,84	1,24	2,09	2,49
CNAG_02084 (4)		-2,17	-1,83	1,00	1,34
CNAG_03143 (2)		0,58	0,48	0,72	0,62
CNAG_01958		0,39	0,34	0,73	0,68
CNAG_01378 (5)		0,35	0,61	0,54	0,80
CNAG_00261 (14)		0,06	0,37	0,28	0,59
CNAG_01355 (2)		0,25	0,35	0,62	0,72
CNAG_07510		0,64	0,81	0,69	0,86
CNAG_06344		-0,28	0,18	0,16	0,62
CNAG_00186 (4)		-0,48	0,06	0,13	0,67

Figure 5.1: Visualization techniques for comparing multiple clustering results in XCluSim. There are three types of overviews: (A) parameter information view, (B) force-directed layout overview, and (C) dendrogram overview. They enable users to simultaneously compare multiple clustering results in a scalable way. When some clustering results are selected in the overviews, they are added to (D) enhanced parallel sets view for more in-depth comparison tasks. Users can access the detailed information of the selected clustering results with each result in each tab of (E) the tabular list view.

visual comparison. Since each combination of different clustering algorithms and their parameters may yield different clustering results, it is inevitable from those many clustering results to (1) see their overall similarity first, (2) choose a subset of them, and then (3) perform detail comparisons and explore individual clustering results.

XCluSim provides as many clustering options as possible by implementing famous clustering algorithms and linking the clustering algorithms available in Weka [42]. It also keeps track of clustering options that users try during the analysis process.

In the following subsections, we introduce visualization techniques and user interactions for comparison tasks. They include overview, filtering/selection, and detail view. Then we present visualization techniques that help users to explore individual clustering results. For better comprehension of the visualization components in XCluSim, we first describe a color encoding strategy for clusters, which we consistently apply to every visualization component of XCluSim prior to explaining each visualization.

5.3.1 Color Encoding of Clusters Using Tree Colors

To help users identify similarities among multiple clustering results, we color-code each cluster based on Tree Colors [123], which provides a color-coding scheme for tree-structured data. We first hierarchically cluster all clusters from every clustering result using HAC. The correlation coefficient is used as the similarity measure between a pair of clusters as in [150]. This maintains consistency in the use of the cluster similarity measure in XCluSim, which is also used for rearranging bands (i.e., clusters) in the enhanced parallel sets view (see the Enhanced parallel sets view section). In the resulting tree-structured cluster hierarchy, we assign an appropriate color to each cluster based on the Tree Colors color-coding scheme so that similar clusters have similar colors.

This color encoding helps users intuitively assess the similarity of clusters. For example, in Figure 5.1D (the enhanced parallel sets view), ① and ② have very similar colors while ① and ③ do not, which means that ① and ② share most items while ① and ③ barely share any item. This color-coding scheme is consistently applied to overviews, detail views, and every visualization for individual clustering results.

5.3.2 Overview of All Clustering Results

Parameter Information View

XCluSim provides an overview of parameters for all clustering results in the parameter information view (Figure 5.1A, 5.2A). This view is vertically divided into subsections, each of which corresponds to an individual clustering algorithm (e.g., "K-means clustering"). Inside each subsection, there are multiple bar charts arranged in a matrix layout. Each bar chart shows the number of clustering results generated by the corresponding algorithm with the corresponding parameter setting. For example, in Figure 5.1, the parameter information view is divided into more than four subsections (some subsections are hidden under the scroll view) since a user made clustering results using algorithms such as HAC, self-organizing map (SOM) clustering, K-means clustering, and expectation-maximization (EM) clustering. As shown in Figure 5.1, the bar in the left bottom cell of K-means clustering is taller than any bars shown in any clustering algorithms, indicating that the K-means clustering algorithm with a distance measure of Euclidean distance and with 9 as the number of clusters is the one mostly used (Figure 5.1). We note here that bioinformaticians often run a clustering algorithm

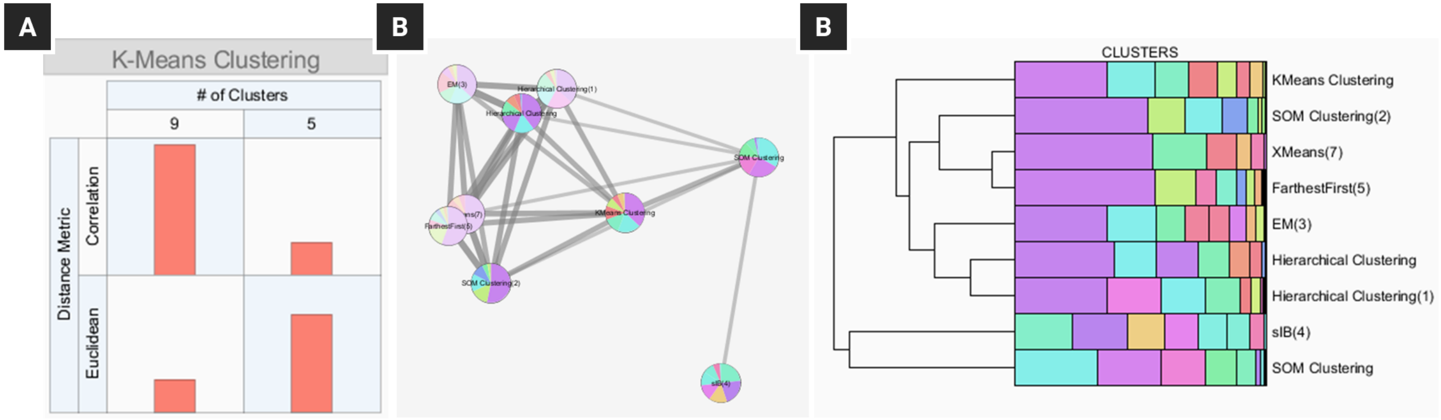


Figure 5.2: Three overviews supported in XCluSim. (A) The parameter information view provides the parameter settings used for the clustering results produced. The table in the parameter information view is for a clustering algorithm, and it shows a bar in each cell to represent the number of clustering results using the corresponding parameter setting. (B) The force-directed layout overview intuitively shows similarity among multiple clustering results with the distance between nodes representing similarity. (C) The dendrogram overview shows similarities between clustering results in a familiar dendrogram layout with a clustering result visualized at a terminal node.

multiple times even with the same parameter setting when the algorithm (e.g., K-means) works non-deterministically. For more details on clustering parameters, the user can also look into the visualization of individual clustering results.

To help users determine which results to select for detailed analysis, XCluSim provides scalable similarity overviews both at the cluster level and at the clustering result level using a force-directed layout (FDL) and a dendrogram view. In the next two sections, we present details of these two overviews.

Force-directed Layout (FDL) Overview

In the FDL overview, overall similarity relations among multiple clustering results are visualized in a force-directed layout, where more similar results are placed closer together and connected with thicker edges (Figure 5.1B. 5.2B). The similarity metric for calculating distances between nodes is F-measure [130], which is the harmonic mean of the precision and recall measure. Each of the precision and recall measures for the two clustering results is calculated by dividing the number of agreed pairs of items by the number of all pairs of items belonging to a clustering result. An agreed pair refers to two items that "agree" to be clustered together in both clustering results.

Since the FDL overview uses physical distance to visually encode similarity between clusters, it has a perceptual advantage in revealing similarity relations among them. In addition, a pie chart is embedded in each node to enable users to visually estimate the number of clusters and their sizes. Since the global color encoding scheme also helps users to grasp similarities among clusters, users can estimate which clusters remain stable across different clustering results. For the scalability of the FDL overview, nodes become smaller as more results are added to the view. Moreover, an edge between two clusters is displayed only when the similarity between the clusters exceeds a predetermined similarity threshold.

Dendrogram Overview

The overall similarity relations are also visualized in the dendrogram overview (Figure 5.1C, 5.2C) after running a HAC with all clustering results (i.e., each row or node represents a result). As in the FDL overview, we use the F-measure as the distance measure between a pair of results. However, the visual representation and its purpose are different from the FDL overview. While the FDL overview intuitively shows similarities using physical distance, the dendrogram overview uses a more familiar clustering visualization component (i.e., a dendrogram) to represent similarities between clustering results. Moreover, the dendrogram overview is more space-efficient so that users can see clustering results and cluster distributions more clearly without occlusion.

5.3.3 Visualization for Comparing Selected Clustering Results

When users identify clustering results of their interests in the overview of all results, they want to select them and perform more in-depth comparison with them. In the next two subsections, we introduce visualizations for comparing the selected clustering results: the enhanced parallel sets view and the tabular list view. When a user selects a result either in the FDL or dendrogram overviews, the selected result is added to the enhanced parallel sets view for more in-depth comparison. The tabular list view, located on the rightmost side of XCluSim, enables users to access detailed information of the selected clustering results with each result in a separate tab.

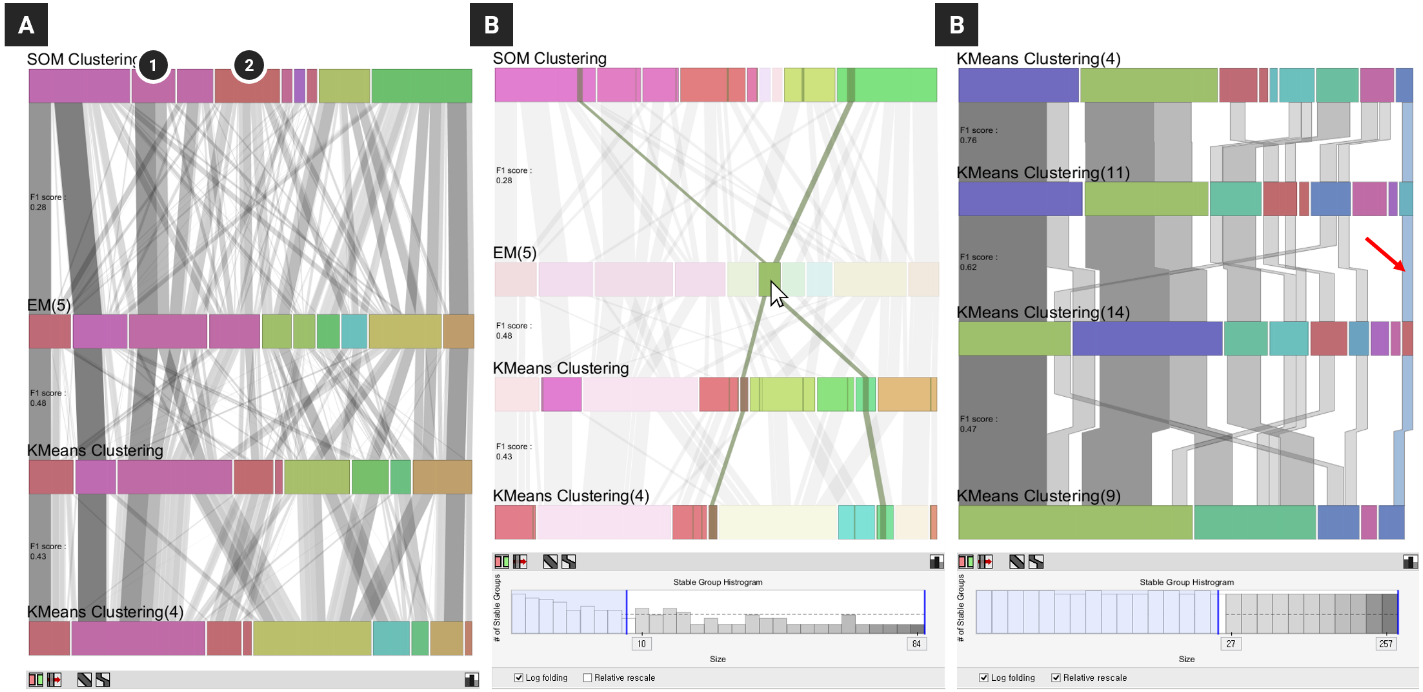


Figure 5.3: The enhanced parallel sets view with various user interactions for in-depth comparison. (A) The parallel sets view provides rearranging algorithms that minimize line crossings. (B) When users hover a mouse pointer over the node of a cluster, the stable groups contained in it are highlighted while other stable groups fade out to reveal flows more clearly. By using a filtering feature on the stable group histogram at the bottom of the parallel sets view, users can hide less interesting bands. (C) Moreover, by using common angle plot [47], users can compare the sizes of different bands more accurately.

Enhanced Parallel Sets View

To visualize the concordance and discordance of multiple clustering results in more detail, we utilized parallel sets [13]. We enhanced the parallel sets for effective clustering result comparison by designing more appropriate interactions and revealing more relevant information, that is a stable group (explained in detail later in this section). In the parallel sets view (Figure 5.1D, 5.3), each horizontal row of stacked bars represents a clustering result. A tiny gap is placed between each bar to assist users to correctly perceive a single cluster since adjacent bars can occasionally have similar colors when the Tree Colors scheme is used. Rows are arranged in such a way that the distance between adjacent rows encodes the dissimilarity between the corresponding clustering results. Each horizontal bar in a row represents a cluster in the corresponding result. We define a stable group of items as a set of items that are clustered together through all selected clustering results. A stable group is represented as a ribbon-like band across all rows. Since the parallel sets view only enables comparisons based on a linear ordering of results, users can interactively switch any two rows by dragging one over the other. When the vertical order of the rows is changed, all rows are replaced accordingly to reflect the similarity between new adjacent clustering results.

The aggregated band representation for links connecting items in a stable group significantly reduces visual clutter compared to the use of a single line representation to connect individual items. The width of a band is an important visual cue that encodes important information about a stable group (i.e., its size) in XCluSim. Users can easily recognize the largest groups of items that are clustered together across multiple clustering results as they spot thick bands. Moreover, users can visually estimate the stability of a cluster by looking at the width of each stable group in it. For example, since the average width of stable groups in ① is bigger than ② in Figure 5.3A, a user can infer that ① is a more stable cluster than ②. Cluster-similarity based on the color-coding of bars (i.e., clusters) helps to facilitate the comparison of multiple clustering results.

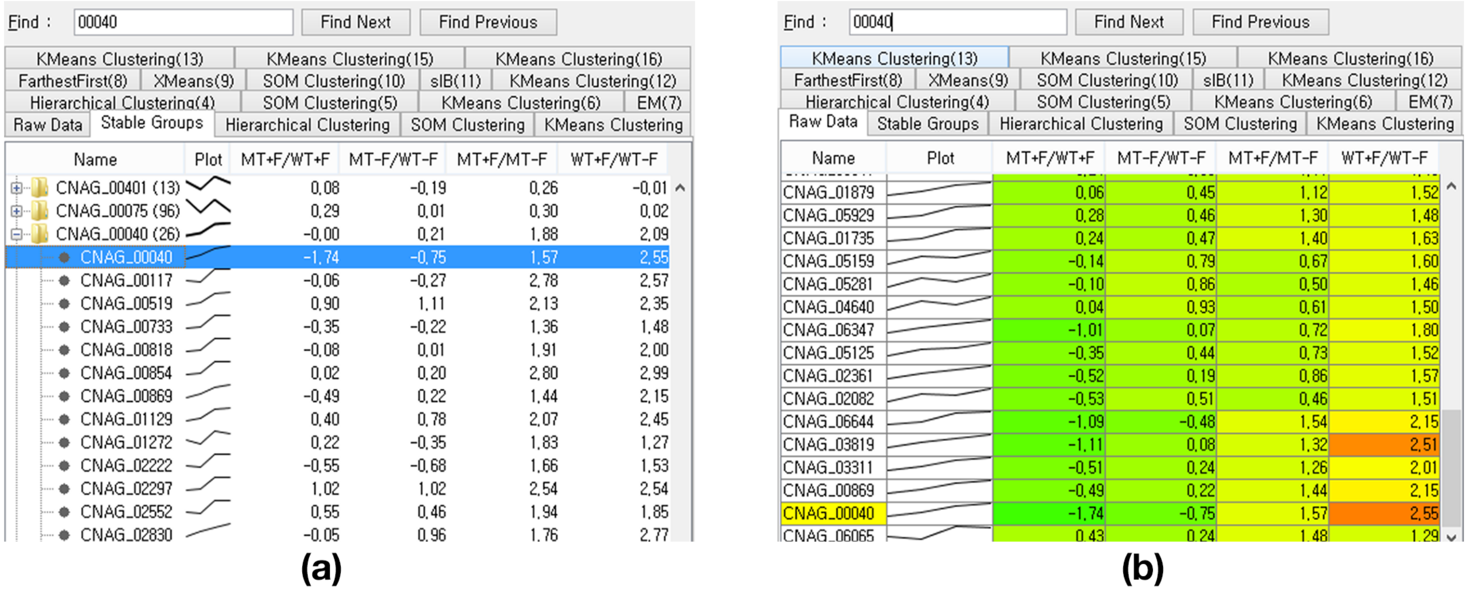
However, the aggregation method could still suffer from clutter due to band-crossings. We applied a rearrangement algorithm [150] to address this issue. To provide more flexible user interaction depending on a user's need, we divided the algorithm into two rearrangement features: rearranging clusters (i.e., bar rearrangement) and rearranging their members (i.e., band rearrangement). These features can be evoked by pressing on the button at the bottom of the enhanced parallel sets view (Figure 5.1D). When a user uses any of these two features, a smooth animated transition is supported to reduce the cognitive burden that accompanies users' attempts to trace the movement of bands or bars.

XCluSim provides more user interactions to overcome the cluttering problem. First of all, users can alleviate the visual clutter in the region of interest by rearranging the bars in a row. This involves dragging them horizontally. After manually rearranging bars (i.e., clusters), users can employ the band rearrangement feature to reduce the visual clutter of bands across multiple rows due to the current manual arrangement of bars in the row. Secondly, there is a band filtering feature similar to that in [113]. The stable group histogram at the bottom of Figure 5.3C shows the distribution of bands by size. There are two blue filtering bars on both sides. Users can filter out bands that are too small or too big from the parallel sets view by adjusting the position of the filtering bars. Finally, when the mouse pointer hovers over a cluster, it highlights the bands, allowing the clusters to show their flows across other clustering results clearly (Figure 5.3B). This can be helpful when a user is especially interested in stable groups that belong to a specific cluster.

The perception of a stable group's size could be distorted by a line width illusion [47]. Such an illusion causes humans to perceive the line width incorrectly at slanted angles. This distortion may disrupt the task of band size comparison. In order to prevent it, we adopt the common angle plot [47] idea (Figure 5.3C). By comparing the straight, vertical parts of bands, users can compare the sizes of the stable groups more accurately. However, since the common angle plot represents a single line as three connected straight lines, it may generate more clutter and occlusions. Thus, it is better to use this feature when only a small number of bands are displayed in the parallel sets view.

Tabular List View

Users can access detailed information concerning the selected clustering results with each result in a separate tab in the tabular list view (Figure 5.4). The tabular view provides detailed information in two different modes: the group-by mode and the heatmap mode. In the group-by mode, users can see the data grouped by stable groups or by clusters. A group is represented by a representative item in a single row with the number of group members between parentheses. Moreover, there is a line graph glyph in each row to show the overall average pattern of the corresponding group. In the heatmap mode, the tabular list view shows numerical details with each cell color-coded according to its value. There is a text search field on the top of the tabular list view so that users can directly access specific items. A user can export a selected subset of data (e.g., a specific stable group) as a CSV text file for further analysis.



Find : 00040 | Find Next | Find Previous

KMeans Clustering(13) | KMeans Clustering(15) | KMeans Clustering(16)
FarthestFirst(8) | XMeans(9) | SOM Clustering(10) | sIB(11) | KMeans Clustering(12)
Hierarchical Clustering(4) | SOM Clustering(5) | KMeans Clustering(6) | EM(7)
Raw Data | Stable Groups | Hierarchical Clustering | SOM Clustering | KMeans Clustering

Name	Plot	MT+F/WT+F	MT-F/WT-F	MT+F/MT-F	WT+F/WT-F
CNAG_00401 (13)		0.08	-0.19	0.26	-0.01
CNAG_00075 (96)		0.29	0.01	0.30	0.02
CNAG_00040 (26)		-0.00	0.21	1.88	2.09
CNAG_00040		-1.74	-0.75	1.57	2.55
CNAG_00117		-0.06	-0.27	2.78	2.57
CNAG_00519		0.90	1.11	2.13	2.35
CNAG_00733		-0.35	-0.22	1.36	1.48
CNAG_00818		-0.08	0.01	1.91	2.00
CNAG_00854		0.02	0.20	2.80	2.99
CNAG_00869		-0.49	0.22	1.44	2.15
CNAG_01129		0.40	0.78	2.07	2.45
CNAG_01272		0.22	-0.35	1.83	1.27
CNAG_02222		-0.55	-0.68	1.66	1.53
CNAG_02297		1.02	1.02	2.54	2.54
CNAG_02552		0.55	0.46	1.94	1.85
CNAG_02830		-0.05	0.96	1.76	2.77

(a)

Find : 00040 | Find Next | Find Previous

KMeans Clustering(13) | KMeans Clustering(15) | KMeans Clustering(16)
FarthestFirst(8) | XMeans(9) | SOM Clustering(10) | sIB(11) | KMeans Clustering(12)
Hierarchical Clustering(4) | SOM Clustering(5) | KMeans Clustering(6) | EM(7)
Raw Data | Stable Groups | Hierarchical Clustering | SOM Clustering | KMeans Clustering

Name	Plot	MT+F/WT+F	MT-F/WT-F	MT+F/MT-F	WT+F/WT-F
CNAG_01879		0.06	0.45	1.12	1.52
CNAG_05929		0.28	0.46	1.30	1.48
CNAG_01735		0.24	0.47	1.40	1.63
CNAG_05159		-0.14	0.79	0.67	1.60
CNAG_05281		-0.10	0.86	0.50	1.46
CNAG_04640		0.04	0.93	0.61	1.50
CNAG_06347		-1.01	0.07	0.72	1.80
CNAG_05125		-0.35	0.44	0.73	1.52
CNAG_02361		-0.52	0.19	0.86	1.57
CNAG_02082		-0.53	0.51	0.46	1.51
CNAG_06644		-1.09	-0.48	1.54	2.15
CNAG_03819		-1.11	0.08	1.32	2.51
CNAG_03311		-0.51	0.24	1.26	2.01
CNAG_00869		-0.49	0.22	1.44	2.15
CNAG_00040		-1.74	-0.75	1.57	2.55
CNAG_06065		0.43	0.24	1.48	1.29

(b)

Figure 5.4: The tabular list view enables users to access numerical details. (a) Users can see detailed information for each item grouped by cluster or stable group. (b) Users also can see raw data in a heatmap form. When a user wants to access an item or a group directly, he/she can use the search box provided on the top of the tabular list view.

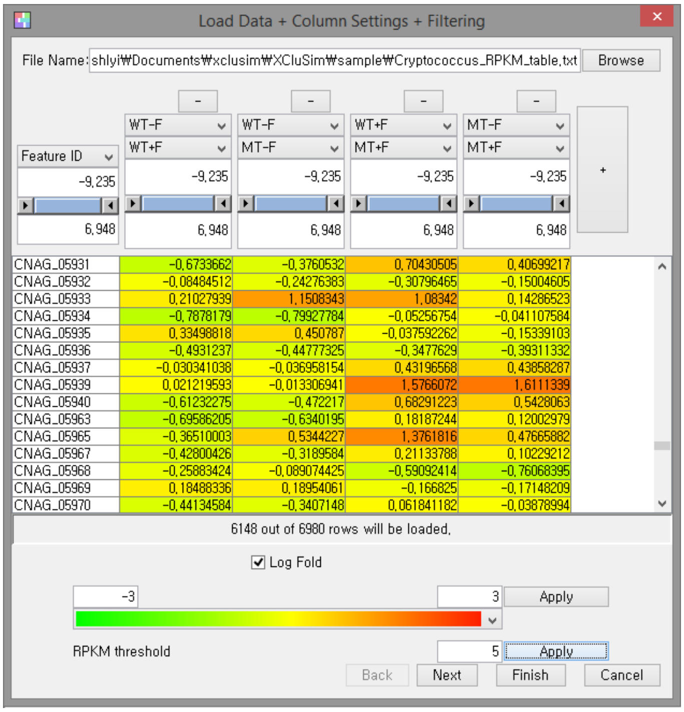
XCluSim provides brushing and linking among all visualization components. Thus, the tabular list view is coordinated with all visualization components in XCluSim. Thus, whenever a user selects a group of items in any visualization, they are highlighted in the tabular list view to help the user access detailed information about them. In addition, when the mouse pointer hovers over an item in a component, it highlights the item in white-blue color, and all related items on the other components are also highlighted. This could lead to additional meaningful insights. For example, hovering a mouse pointer over the title of a specific algorithm in the parameter information view results in the highlighting of all related clustering results in overviews and detail views (Figure 5.1). As a consequence, users are able to understand that K-means clustering can produce totally different clustering results depending on the clustering parameters chosen (e.g., compare "K-means clustering(10)" to "K-means clustering(11)" in the dendrogram overview in Figure 5.1).

Interactive Data Manipulation

Simple file formats such as comma-separated values (CSV) and tab-delimited text are used for XCluSim. XCluSim enables researchers to interactively manipulate the input dataset when loading it, prior to clustering it 5.5). Users can generate a ratio value by selecting two columns from the original dataset. XCluSim provides filters such as a range filter and RPKM threshold adjustment. It also provides features for calculating fold changes.

5.3.4 Visualization for Individual Clustering Results

To make XCluSim a more general visual analytics tool for comparing clustering results, we try to provide a wide variety of clustering algorithms. First of



Load Data + Column Settings + Filtering

File Name: shlyi₩Documents₩xclusim₩XCluSim₩sample₩Cryptococcus_RPKM_table.txt

Feature ID	WT-F WT+F	WT-F MT-F	WT+F MT+F	MT-F MT+F
-9,235	-9,235	-9,235	-9,235	-9,235
6,948	6,948	6,948	6,948	6,948
CNAG_05931	-0,6733662	-0,3760532	0,70430505	0,40699217
CNAG_05932	-0,08484512	-0,24276383	-0,30796465	-0,15004605
CNAG_05933	0,21027939	1,1508343	1,08342	0,14286523
CNAG_05934	-0,7878179	-0,79927784	-0,05256754	-0,041107584
CNAG_05935	0,33498818	0,450787	-0,037592262	-0,15339103
CNAG_05936	-0,4931237	-0,44777325	-0,3477629	-0,39311332
CNAG_05937	-0,030341038	-0,036958154	0,43196568	0,43858287
CNAG_05939	0,021219593	-0,013306941	1,5766072	1,6111339
CNAG_05940	-0,61232275	-0,472217	0,68291223	0,5428063
CNAG_05963	-0,69586205	-0,6340195	0,18187244	0,12002979
CNAG_05965	-0,36510003	0,5344227	1,3761816	0,47665882
CNAG_05967	-0,42800426	-0,3189584	0,21133788	0,10229212
CNAG_05968	-0,25883424	-0,089074425	-0,59092414	-0,76068395
CNAG_05969	0,18488336	0,18954061	-0,166825	-0,17148209
CNAG_05970	-0,44134584	-0,3407148	0,061841182	-0,03878994

6148 out of 6980 rows will be loaded.

☑ Log Fold

-3 3

RPKM threshold 5

Figure 5.5: Interactive manipulation of input data supported in XCluSim: derive a new column (ratio, fold change), change color mapping, filter items using a range filter and RPKM adjustment.

all, we implement frequently-used clustering algorithms in XCluSim. These include Hierarchical Agglomerative Clustering [29], SOM clustering [63], K-means clustering, and OPTICS clustering [8]. Moreover, all clustering algorithms from Weka [42] are also available in XCluSim. Users can also import any clustering results made by any other clustering algorithms that are not available in XCluSim.

Taxonomy of Visualization Techniques for Visualizing Clustering Results

Different clustering algorithms work on different principles. For example, there are three major categories of clustering algorithms: hierarchical, partitional, and density-based. Clustering algorithms in different categories need

Principle to Show Cluster Membership		Visualization Component	Clustering Algorithm		
Main	Secondary		Hierarchical	Partitional	Density-based
Similarity (color or size)	Proximity	Scatterplot + Color	△	△ [49]	△ [5]
		*Graph (vertex as item) + Color	△	△ [28]	△ [5]
	Enclosure	*Bar chart (Reachability Plot)	X	X	O [8, 42, 55]
		Colored shape	△ [96]	△	△ [64]
	.	*Parallel coordinates plot + Color	△ [150]	△	△
Proximity	Similarity and Enclosure	Bar chart (Silhouettes Plot)	△	O [28, 104]	O
Connectedness (line connection)	Similarity and Proximity	*Dendrogram	O [74, 98, 112]	X	△
		Normal tree	△ [12]	X	△
		Circular tree	△ [12]	X	△
Enclosure	Proximity	*Heatmap + Partitioning	O	O	O
	Similarity and Proximity	Treemap	△ [12]	X	X

Table 5.1: Taxonomy of visualization techniques for visualizing clustering results. Visualization components for visualizing clustering results use visual cues based on Gestalt principles of grouping [139] to represent cluster membership. We categorize the visualization components by principle and indicate how appropriate each visualization component is for showing clustering results by different types of clustering algorithms. (i.e. ”O“ for most appropriate, ”Δ“ for moderately appropriate, and ”X“ for not applicable).

different visualization techniques to effectively visualize their clustering results.

To suggest effective visualizations for each category of clustering algorithms, we first surveyed visual encoding techniques for visualizing the clustering results of various algorithms (Table. 1). Sedlmair et al. presented a related taxonomy of factors in visual cluster separation [110]. They evaluated the effect of each factor on visual cluster separation in scatterplots. Building upon this work, we consider the appropriateness of visual encoding techniques in representing the characteristics of each type of clustering algorithm. To broaden the perspective of our taxonomy, we further categorize the visual encoding techniques in terms of Gestalt principles of grouping [139]: similarity, proximity, connectedness, and enclosure.

Similarity: The similarity principle is the one most commonly used in cluster visualization. It helps users to perceive cluster membership by employing similar colors, shapes, or sizes. Among them, color is the most frequently used visual cue. However, using color as the main visual cue may not scale well because the use of human color perception to discriminate between classes is limited to a number of colors. Thus, it is often used in conjunction with visual cues such as in reachability plot [42] and silhouettes plot [28].

Proximity: This principle facilitates the perception of cluster membership by placing related items closer together. For example, in the silhouettes plot [104], bars belonging to the same cluster are placed next to each other. However, this principle is not used alone. It is typically used together with other visual cues. For example, the partitioned heatmap sometimes puts gaps between clusters to show their boundaries clearly [72, 74, 75, 112].

Connectedness: The connectedness principle helps users to identify groups by connecting related items using a visual artifact such as a line. Line connection is one of the most powerful visual cues among the Gestalt principles of grouping. However, it can confuse users when there are too many lines in a single view. The connectedness principle is especially used with hierarchical clustering results since hierarchy structures can best be demonstrated with line connections. For example, HCE [112], Matchmaker [74], and others use this principle to represent clusters in dendrograms.

Enclosure: The enclosure principle is adopted particularly when drawing a closed boundary containing items belonging to a cluster. For example, when a dataset contains spatial information, all items of a cluster are shown on a color-coded region with a solid boundary [64, 96]. Another typical technique based on this principle is the partitioned heatmap [72, 75]. It is a powerful way to display raw data while clearly specifying the boundary surrounding the members of each cluster.

In addition to these four Gestalt principles of grouping, there are some attempts to use abstract representations (such as glyphs or special shapes) for clusters without showing any individual items in clusters. The cluster graph [136] uses an abstract representation of a circular node for a cluster. Clusters derived from SOM clustering results are visualized in a hive-shaped grid view while each item is abstracted as a node [63]. As these attempts do not allow for the visualization of individual items, they are not a good fit for the classification based on Gestalt principles.

After reviewing and categorizing visual encoding techniques for visualizing clustering results, we selected visualization techniques appropriate for visualizing each of the three main kinds of clustering algorithms: hierarchical clustering, partitional clustering, and density-based clustering. In the next three subsections, we describe the visualization techniques in detail.

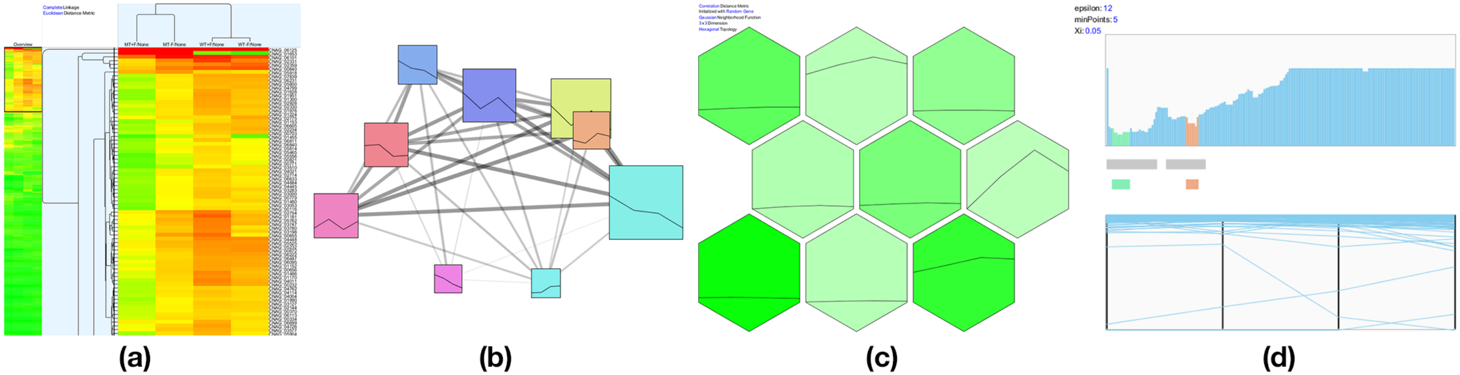


Figure 5.6: Visualization techniques for individual clustering results in XCluSim. (a) Dendrogram + heatmap visualization for hierarchical agglomerative clustering results. (b) Force directed layout for every partitional clustering result and imported clustering results. (c) Common hive-shaped visualization for SOM clustering results. (d) Reachability plot together with parallel coordinates plot for OPTICS.

Visualization Technique for Hierarchical Clustering

We visualized HAC results with the combination of a dendrogram and heatmap visualization (Figure 5.6a), where users could interactively compress/expand, flip, and swap sub-trees. The batch compression of sub-trees using the minimum similarity bar [112] is also possible. By adjusting the position of the similarity bar, users can dynamically determine the clusters. There is a compact bird's-eye overview using heatmap [73] in the left-most part which is tightly coupled with the dendrogram. By dragging a black-bordered rectangle that represents the current viewport (see the black rectangle in the top-left of Figure 5.6a) in the heatmap overview, users can efficiently navigate through the dendrogram+heatmap view.

Visualization Technique for Partitional Method

Partitional clustering results other than SOM clustering (e.g. K-means clustering, EM clustering, farthest first clustering, etc.), and all imported results are visualized in a force-directed layout (Figure 5.6b), where each cluster is represented as a rectangle whose size is proportional to the cluster size. The force between nodes is determined by the similarity between members of each cluster so that similar clusters are closely positioned and have thicker links between them. To show an overview of a cluster, XCluSim also visualizes the average pattern of all members of the cluster in a line chart, which is shown as a glyph in the cluster's node. XCluSim also supports semantic zooming to enable users to explore clusters in more detail. When a cluster is zoomed into, more details of its members are dynamically visualized in a parallel coordinates plot.

SOM clustering results are visualized using the typical hive-shaped visualization (Figure 5.6c), where each hexagonal cell represents a cluster. In

XCluSim, the background intensity of each cell represents the size of the corresponding cluster. As a visual summary of each cluster, XCluSim presents the average pattern of the cluster members in a line chart within each hexagonal cell. XCluSim also supports semantic zooming. Users can zoom into a cluster by double-clicking on the corresponding cell and look at the details of their members in a parallel coordinates plot in the same way they would in a force-directed layout.

Visualization Technique for Density-based Method

Density-based clustering algorithms calculate a kind of density-related information for each item during the clustering process. For example, OPTICS [8] calculates the reachability distance for each item. We believe that users can more intuitively understand a density-based clustering result when the density-related information is revealed. Therefore, a bar-chart-like visualization, with each item arranged on the horizontal axis and the density-related information on the vertical axis, can effectively visualize density-based clustering results. The conventional reachability plot for OPTICS is a typical example. In XCluSim, we enhance the plot for better cluster identification and for the improved examination of details (Figure 5.6d). To clearly show the position of each cluster, XCluSim places a horizontal bar from the start to the end positions of the cluster right below the reachability plot. The parallel coordinates plot at the bottom shows more details of cluster members. These two plots support brushing and linking between the cluster members. For example, when a mouse pointer hovers over a cluster in the reachability plot, the lines for the members of the cluster are highlighted in the parallel coordinates plot.

5.3.5 Implementation

XCluSim was developed using Java Standard Edition 7 (Java SE 7), which enables it to run on any platform with JRE version 1.7 or higher. We used the Piccolo 2D framework to implement visualization components and interactions. Weka's clustering algorithms were integrated into XCluSim using Weka SDK 3.6 [42].

5.4 Case Study

To evaluate the efficacy of XCluSim, we conducted two case studies with our collaborator in a major bioinformatics research laboratory. He is a senior research engineer and has years of experience in genome and transcriptome analyses.

5.4.1 Elucidating the Role of Ferroxidase in Cryptococcus Neoformans Var. Grubii H99 (Case Study 1)

This study was carried out in his laboratory for 80 minutes. Pre- and post-study interviews were conducted for 10 minutes each. The participant used XCluSim for 50 minutes after a 10-minute tutorial. We used a dataset containing normalized expression levels of 6,980 genes belonging to the Cryptococcus neoformans var. grubii H99 strain. The dataset had been prepared for his previous work [60].

His task was to elucidate the role of ferroxidase (cfo1) by knocking it out. He was interested in finding a meaningful set of genes whose expression would be influenced and in identifying the affected pathways. For the task, he tried to see the effect of fluconazole on two different strains: the wild type of Cryptococcus neoformans var. grubii H99 and the cfo1 mutant of the same strain. In the dataset, each gene has four expression levels: two different strains, each cultured in two conditions (i.e., wild-type strain and cfo1 mutant with and without fluconazole treatment).

When he loaded the data, he made four new data columns of ratio values, including the wild-type strain with fluconazole versus the wild-type strain without fluconazole treatment (WT+F/WT-F) and the cfo1 mutant with fluconazole versus the cfo1 mutant without fluconazole treatment (MT+F/MT-F). Subsequently, he adjusted the RPKM threshold and used log fold changes to filter out less interesting genes for more efficient analysis.

After data pre-processing, XCluSim showed the results of three clustering algorithms (i.e., HAC, SOM clustering, K-means clustering) in three independent views. Since he was most familiar with dendrogram and heatmap visualization, he examined the HAC results first. He was interested in genes that were highly expressed with fluconazole treatment. Among them, he found the gene named Erg11 (CNAG_00040). He said that this gene was reported to be associated with azole resistance.

Next, he tried to see which genes were stably grouped together across different clustering results. He tried to load as many clustering results as possible to see the differences between them. The parameter information view provided him with a good overview of all clustering results (clustering algorithms and their parameters). He was able to make diverse clustering results without generating any duplicate results.

After generating 15 different clustering results, he selected four diverse results from the FDL overview to find out which genes were clustered together with Erg11. However, he recognized that the stable groups were excessively thin because of the result named "FarthestFirst(6)." This had to do with the fact that it was the most dissimilar result to other selected clustering results (Figure 5.1). So he removed that result from the parallel sets view. Then he selected a more similar one named “KMeans Clustering(4)” (Figure 5.3A). He subsequently accessed the stable group with Erg11 directly, utilizing the search feature in the tabular list view. He was able to confirm that 17 other genes belonged to the stable group. After validating the members of the stable group with an enrichment analysis, he found that most of them (10 out of 18) belonged to the ergosterol biosynthetic pathway.

Once he had selected the stable group in the tabular list view, he was able to efficiently inspect the flow of the group across different clustering results in the enhanced parallel sets view (Figure 5.3B). While he looked into the flow of the stable group across all rows (the rightmost highlighted-band in Figure 5.3B), he also noticed that the clustering result from “KMeans Clustering(4)” had the tightest cluster, which included the stable group. However, there were no more genes outside the stable group in the cluster that belonged to the ergosterol pathway.

Then he tried to find the best algorithm and those of its parameters that gave the tightest cluster containing genes belonging to the ergosterol pathway. Since “KMeans Clustering(4)” had previously been the best clustering result among the selected results, he ran K-means clustering algorithms with different parameters to arrive at similar results. He then inserted three of the most similar results in the parallel sets view (Figure 5.3C). Again, he highlighted a stable group with Erg1 (the band indicated with a red arrow in Figure 5.3C). By checking the flow of the stable group crossing each result, he recognized that “KMeans Clustering(14)” gave the tightest cluster. This led to the conclusion that K-means clustering with the corresponding parameter configurations (i.e., Euclidean distance as the distance metric and 9 as the number of clusters) was the best result for the given dataset among all the results.

5.4.2 Finding a Clustering Result that Clearly Represents Biological Relations (Case Study 2)

A second case study was subsequently carried out with the same participant in his laboratory. The study was conducted for 150 minutes on a different day. Since the participant was already familiar with XCluSim, we skipped the tutorial. In the study, he relied on the gene expression profiles of 169 genes in Escherichia coli, which used a DNA microarray [59]. In the dataset, each gene contained 19 expression levels in order to investigate the effects of the perturbations on tryptophan metabolism. The expressions were measured under the following conditions: wild type growth with and without tryptophan (five conditions), wild type growth with and without tryptophan starvation (nine conditions), and the growth of wild type and a trp repressor mutant (five conditions).

Through the case study, the participant wanted to find a clustering result that clearly reflected biological relations in tryptophan metabolism. In the original paper [59], the authors used HAC to cluster the 169 gene expression profiles measured in the 19 conditions. It was indicated in the paper that genes showing similar expression responses did not necessarily fall into the same cluster. One example included the genes associated with aromatic amino acid metabolism.

He first wanted to see if the optimal algorithm and its parameters in the previous case study would work for another dataset. To determine this, he produced 11 clustering results in XCluSim, including the result produced using previous optimal settings: K-means clustering with Euclidean distance as

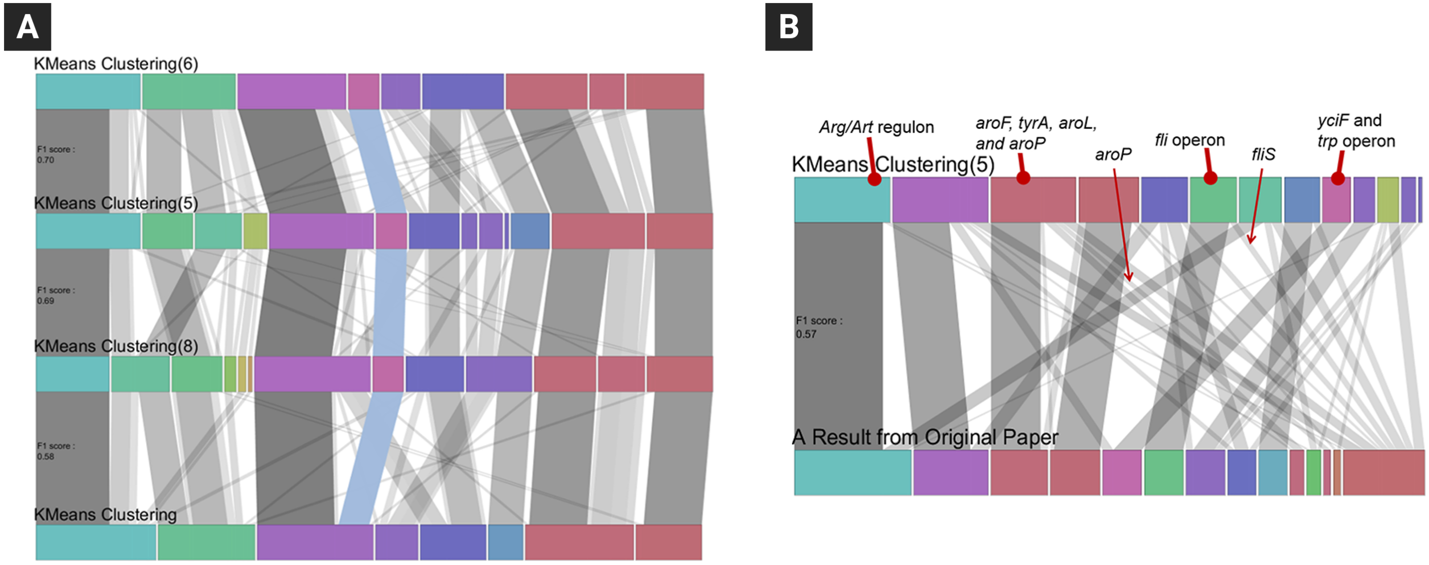


Figure 5.7: Results of the second case study are visualized in the enhanced parallel sets view. (A) The highlighted stable group contained the trp operon with yciF. (B) Visual comparison of two results: the best clustering result (“KMeans Clustering(5)”) derived from the case study and a result (“A Result from Original Paper”) presented in the original research paper [59].

the distance metric and 9 as the number of clusters. He validated each cluster in the result ("KMeans Clustering(6)" in Figure 5.7A) through an enrichment analysis using the DAVID website (http://david.abcc.ncifcrf.gov/). After validating each cluster, he concluded that most of the clusters were grouped well in the sense that they represented biological relations in pathways. However, he recognized two problems in the result. First of all, a cluster that had both Arg and Art regulons also contained a gene named tnaA that was considered to be noise. This was because tnaA showed a different expression pattern and was not highly related to other cluster members in biological terms. Secondly, one gene from the fli operon, fliS, fell into a different cluster from the other genes in the same operon while they had homogeneous expression patterns.

By utilizing visualizations in XCluSim, he wanted to find the clustering result that properly represented biological relations as "KMeans Clustering(6)" while the two problems were revisited. For this intended task, he selected all the similar results from the FDL overview: "KMeans Clustering(5)," "KMeans Clustering(8)," and "KMeans Clustering." Then he accessed the stable groups that contained tnaA and the Arg/Art regulon. He easily recognized that genes in both the Arg and Art regulons fell into the same stable group while tnaA was not stably clustered with them. The results, which separately clustered tnaA from the Arg and Art regulons, were "KMeans Clustering(5)" and "KMeans Clustering(8)." Similarly, by checking the flow of stable groups in each horizontal row, he easily recognized that two clustering results that used the correlation coefficient as a distance metric clustered two stable groups together: one with the fli operon and the other with fliS. The two results were "KMeans Clustering(5)" and "KMeans Clustering." As a consequence, "KMeans Clustering(5)," using the correlation coefficient as the distance metric and 13 as the number of clusters, was the most satisfying result for the dataset.

Additionally, our participant gained insight by seeing a stable group in XCluSim. Genes in the trp operon (i.e., trpE, trpD, trpC, trpB, and trpA) were stably clustered together with yciF through the four different results (see the highlighted stable group in Figure 5.7A). Since yciF was assigned to a putative function, he said that the gene might be closely related to tryptophan synthase as a trp operon.

After he found the best result, he compared it with a clustering result provided in the original work [59] to see if his result better represented biological relations (Figure 5.7B). The clustering result presented in the paper had been prepared prior to the study and was imported to XCluSim for visual comparisons. After comparing two results, he found that some of the genes involved in aromatic amino acid metabolism, aroF, tyrA, aroL, and aroP, were clustered together in our best result while only three of them fell into the same cluster in their original result. Moreover, their result did not cluster fliS with the other fli operon. These results suggested that the authors of the original work [59] could have generated more biologically meaningful results if they had used XCluSim in the first place.

5.5 Discussion

During the case studies, we received positive subjective feedback on XCluSim from the participant. He especially liked the ability to identify stable groups across multiple clustering results. Moreover, he was satisfied that he could select and run diverse clustering algorithms and interactively compare them by adding/removing a clustering result to/from the enhanced parallel sets view. He could quickly shift his attention to a more interesting set of results for more in-depth comparison. However, he also pointed out the limitations of XCluSim. Since filtering sets of items was only available at the data manipulation step, he said it would be helpful to allow users to interactively filter raw data in the visualization components as well.

We color-coded each cluster consistently across the whole system using the Tree Colors scheme after building a hierarchical structure of all clusters from multiple clustering results. With the help of this color coding, overviews became even more useful in XCluSim. While the color encoding was applied for a specific purpose in this work (i.e., for the visualization of clusters), we think it can also be applied to parallel sets applications in a more general and scalable way. For example, instead of distinguishing only a small number of categories while visualizing a categorical dataset, it might be possible to distinguish many more nodes in the parallel sets once a hierarchical structure of the nodes has been built in a similar manner to the one we employed in XCluSim.

We provided a taxonomy of visualization techniques for visualizing clustering results based on the Gestalt principle of grouping and the types of clustering algorithms. The design space defined by this taxonomy can help researchers to make design decisions for clustering results visualization. By thinking about visualization techniques in terms of the Gestalt principle, researchers can come up with better visual encoding without overlooking important features. For example, since the graph layout is used to visualize cluster memberships by color-coding each item [5, 28], one can also utilize the enclosure principle, such as GMap [35] and BubbleSets [21], to represent their membership more clearly.

5.5.1 Limitations and Future Work

At present, when a clustering algorithm does not assign all items to clusters, all un-clustered items are treated as a single cluster in XCluSim. OPTICS and DBSCAN clustering algorithms can give rise to results of this kind. XCluSim treats un-clustered items as a group of less interesting items as if it were a special cluster. Otherwise, it could make a huge number of stable groups since each un-clustered item will become a single stable group. This would make it hard for users to gain insight from visualizations. In the future, we plan to improve XCluSim to resolve this problem. For example, we can represent these kinds of groups with different textures in the parallel sets view to distinguish them from other normal clusters.

In this chapter, we concentrated mostly on supporting comparison tasks based on the concordance/discordance of multiple clustering results. However, since bioinformaticians' cluster analysis is highly integrated with the validation stage, it would also be valuable to provide a visual representation of cluster validity measures (e.g., internal cluster validity indices). For example, the grayscale intensity of each band (i.e., stable group) in the parallel sets view, which currently represents the size of a stable group, can be utilized to represent its internal validity measures. In such a case, stable groups provided by XCluSim will become more reliable information.

5.6 Summary

In this chapter, we presented XCluSim, a visual analytics tool that enables users to compare multiple clustering results. XCluSim provides three different overviews to help users grasp their overall similarity relationships in a more scalable and flexible way. Moreover, the enhanced parallel sets view enables users to detect differences among select clustering results even more clearly by using improved user interactions. We conducted case studies to evaluate the usefulness of XCluSim, and the participants gave positive feedback.

Chapter 6

Future Research Agenda

This chapter suggests the agenda for future research build upon this dissertation, such as designing visual comparison recommendations, understanding the perception of subtle difference in visualizations, and distinguishing InfoVis novices from experts.

6.0.1 Recommendation for Visual Comparison

Although visualization recommendation has been considered as one of the key supportive techniques for InfoVis novices, we still lack a recommendation system that specifically supports visual comparison tasks between multiple visualizations. In this dissertation, we have built the fundamental knowledge on how to design visual comparison recommendation in two aspects: (**recommendation model**) we have organized the usefulness of individual comparative layouts in diverse analysis contexts based on 104 research papers including eight papers with quantitative user studies in Chapter 3, and (**recommendation interface**) we have identified implications for designing understandable recommendation interface for novices based on the result of a qualitative user study ($N = 24$) in Chapter 4. We believe this knowledge can potentially be helpful tools for designing a visualization recommendation interface that supports visual comparison tasks. In the future, we will study on designing recommendation systems that support effective visual comparison to assist novices in organizing their insights gained from exploring individual visualizations during the visual exploration process. A promising recommendation in this area would be suggesting visualization layouts (e.g., juxtaposition or superposition) depending on visualization types, complexity, and tasks.

6.0.2 Understanding the Perception of Subtle Difference

In the future, it seems promising to further understand the effectiveness of individual comparative layouts in a wide range of study factors, such as visualization types and primitive visual channels (e.g., length in bar charts) to show the difference. For example, for local comparison tasks (e.g., comparing the length of bars), animated transition showed the best performance compared with chart-wise and item-wise juxtaposition [94]. Interestingly, however, from our user study in Chapter 4, participants commented that animated transition has not sufficiently showen the difference between visualizations. These contrary results can be explained by different study settings. Firstly, the amount of difference (e.g., SSIM [138]) in pairs of visualizations was different. For example, the recommendation for changing point size in a scatterplot with the small number of points would have made the animated transition harder to be noticed in our study. Secondly, visualization types in the two studies were different as well: scatterplots in our study and bar chart, line chart, and pie chart in the Ondov et al.'s work [94]. Lastly, primitive visual channels (e.g., color, length, and size) that were involved in the predefined difference between pairs of visualizations were also different.

In our study, diverse visual channels were varied, including color (Change Point Opacity) and size (Change Point Size). In contrast, Ondov et al. [94] used main visual channels in individual visualizations, such as length for bar chart and angle for slope chart and pie chart, which do not overlap with ours. One of promising future research would be investigating the effectiveness of animated transition for showing different visual channels between visualizations.

6.0.3 Distinguishing InfoVis Novices from Experts

When we understand the difference between novices and experts in terms of their ability in visual analytics (e.g., perception of visual difference) and are able to predict the expertise level of people based on their behavior patterns, we can provide more personalized and effective visualization designs that can complement their skills. In this dissertation, we identified diverse challenges that novices confront, which can be further studied to compare such challenges with experts. In Chapter 3, we were able to identify diverse novices' challenges in the real-world in literature, such as difficulty in using novel visual representations [77, 106, 118]. Through an observation study in Chapter 4, we empirically found their hurdles in using visual-encoding recommendations, such as recognizing subtle difference. In the future, we will focus on understanding and modeling people's diverging ability in visual analytics through observation studies. One promising study would be modeling just-noticeable difference (JND) for pairs of visualization in these two groups through user studies by employing deep features as perceptual metrics [147].

Chapter 7

Conclusion

In this dissertation, we presented the result of three studies to further build our understanding on designing information visualization (InfoVis) for novices: general people who are not familiar with visual representations and visual data exploration process. In Chapter 3, we presented the result of a literature survey on research papers that suggested novel comparative visualizations. Based on the result, we offered practical implications, such as actionable guidelines for using comparative layouts, as well as the lucid categorization of visualization designs for visual comparison. Identifying the major stages in the visualization construction process [17] where novices confront challenges with visual comparison tasks, we visited two main tasks—comparing visual mapping (encoding barrier) and comparing information (interpretation barrier)—with actual users in Chapter 4 and 5, respectively. Chapter 4 showed that people still rely on textual descriptions to compare the appropriateness of visual encoding suggestions. Moreover, we suggested implications for designing visualization recommendation interfaces that better help novices to compare and understand recommendations. In Chapter 5, we designed and implemented XCluSim, an interactive visual analytics tool for comparing multiple clustering results. Case studies with a bioinformatician showed that XCluSim enabled the analyst to easily evaluate the quality of clustering results, making him allowed to come up with a better result, more clearly reflecting biological relations, compared with the previous study [59].

Bibliography

[1] Juan Alcalde, Clare E Bond, and Charles H Randle. Framing bias: the effect of figure presentation on seismic interpretation. *Interpretation*, 5(4):T591–T605, 2017.

[2] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 483–492. ACM, 2013.

[3] Robert Amar, James Eagan, and John Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Pages 111–117. IEEE, 2005.

[4] Gennady Andrienko, Natalia Andrienko, Georg Fuchs, and Jo Wood. Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2120–2136, 2016.

[5] Gennady Andrienko, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, Dino Pedreschi, and Fosca Giannotti. Interactive visual clustering of large collections of trajectories. In *2009 IEEE Symposium on visual analytics science and technology*, pages 3–10. IEEE, 2009.

[6] Natalia Andrienko and Gennady Andrienko. State transition graphs for semantic analysis of movement behaviours. *Information Visualization*, 17(1):41–65, 2018.

[7] Natalia Andrienko, Gennady Andrienko, Jose Manuel Cordero Garcia, and David Scarlatti. Analysis of flight variability: a systematic approach. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):54–64, 2018.

[8] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod record*, volume 28 of number 2, pages 49–60. ACM, 1999.

[9] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A taxonomy and survey of dynamic graph visualization. In *Computer Graphics Forum*, volume 36 of number 1, pages 133–159. Wiley Online Library, 2017.

[10] Fabian Beck, Michael Burch, and Daniel Weiskopf. A matrix-based visual comparison of time series sports data. In *VMV*, 2016.

[11] Fabian Beck, Franz-Josef Wiszniewsky, Michael Burch, Stephan Diehl, and Daniel Weiskopf. Asymmetric visual hierarchy comparison with nested icicle plots. In *ED/GViP@ Diagrams*, pages 53–62, 2014.

[12] Michael Beham, Wolfgang Herzner, M Eduard Gröller, and Johannes Kehrer. Cupid: cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1693–1702, 2014.

[13] Fabian Bendix, Robert Kosara, and Helwig Hauser. Parallel sets: visual analysis of categorical data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Pages 133–140. IEEE, 2005.

[14] Jürgen Bernard, Anna Vögele, Reinhard Klein, and Dieter W Fellner. Approaches and challenges in the visual-interactive comparison of human motion data. In *VISIGRAPP (3: IVAPP)*, pages 217–224, 2017.

[15] Åsmund Birkeland, Cagatay Turkay, and Ivan Viola. Perceptually uniform motion space. *IEEE Transactions on Visualization and Computer Graphics*, 20(11):1542–1554, 2014.

[16] Fatma Bouali, Abdelheq Guettala, and Gilles Venturini. Vizassist: an interactive user assistant for visual data mining. *The Visual Computer*, 32(11):1447–1463, 2016.

[17] Mackinlay Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[18] H Carr, P Rheingans, and H Schumann. Similarityexplorer: a visual intercomparison tool for multifaceted climate data.

[19] Valentina Caruso, Alberto Cattaneo, and Jean-Luc Gurtner. Creating technology-enhanced scenarios to promote observation skills of fashion-design students. *Form@ re-Open Journal per la formazione in rete*, 17(1):4–17, 2017.

[20] Helen Chen, Sophie Engle, Alark Joshi, Eric D Ragan, Beste F Yuksel, and Lane Harrison. Using animation to alleviate overdraw in multiclass scatterplot matrices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 417. ACM, 2018.

[21] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. Bubble sets: revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[22] Michael Correll, Dominik Moritz, and Jeffrey Heer. Value-suppressing uncertainty palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 642. ACM, 2018.

[23] Lhaylla Crissaff, Louisa Wood Ruby, Samantha Deutch, R Luke DuBois, Jean-Daniel Fekete, Juliana Freire, and Claudio Silva. Aries: enabling visual exploration and organization of art image collections. *IEEE computer graphics and applications*, 38(1):91–108, 2017.

[24] António Cruz, Joel P Arrais, and Penousal Machado. Interactive and coordinated visualization approaches for biological data analysis. *Briefings in bioinformatics*, 2018.

[25] Tuan Nhon Dang and Leland Wilkinson. Scagexplorer: exploring scatterplots by their scagnostics. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 73–80. IEEE, 2014.

[26] Aritra Dasgupta, Susannah Burrows, Kyungsik Han, and Philip J Rasch. Empirical analysis of the subjective impressions and objective measures of domain scientists' visual analytic judgments. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1193–1204. ACM, 2017.

[27] Aritra Dasgupta, Jorge Poco, Yaxing Wei, Robert Cook, Enrico Bertini, and Claudio 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, 2015.

[28] Hao Ding, Chao Wang, Kun Huang, and Raghu Machiraju. Igpse: a visual analytic system for integrative genomic based cancer patient stratification. *BMC bioinformatics*, 15(1):203, 2014.

[29] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[30] Micheline Elias and Anastasia Bezerianos. Exploration views: understanding dashboard creation and customization for visualization novices. In *IFIP Conference on Human-Computer Interaction*, pages 274–291. Springer, 2011.

[31] Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[32] Microsoft excel. 2018. URL: `https://products.office.com/en-us/excel`.

[33] Leo Ferres, Petro Verkhogliad, Gitte Lindgaard, Louis Boucher, Antoine Chretien, and Martin Lachance. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 67–74. ACM, 2007.

[34] Stephen Few. *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press, 2009.

[35] Emden R Gansner, Yifan Hu, and Stephen Kobourov. Gmap: visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208. IEEE, 2010.

[36] Michael Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2017.

[37] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[38] David Gotz and Zhen Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 315–324. ACM, 2009.

[39] Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. A survey of visualization construction user interfaces. *EuroVis-Short Papers*:19–23, 2013.

[40] Lars Grammel and Margaret-Anne Storey. Poster: choosel-web-based visualization construction and coordination for information visualization novices. In *IEEE Information Visualization Conference, InfoVis*, volume 10, 2010.

[41] Lars Grammel, Melanie Tory, and Margaret-Anne Storey. How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952, 2010.

[42] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[43] Susan L Havre, Anuj Shah, Christian Posse, and Bobbie-Jo Webb-Robertson. Diverse information integration and visualization. In *Visualization and Data Analysis 2006*, volume 6060, page 60600M. International Society for Optics and Photonics, 2006.

[44] Jeffrey Heer and George Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[45] Florian Heimerl, Chih-Ching Chang, Alper Sarikaya, and Michael Gleicher. Visual designs for binned aggregation of multi-class scatterplots. *arXiv preprint arXiv:1810.02445*, 2018.

[46] Dominik Herr, Fabian Beck, and Thomas Ertl. Visual analytics for decomposing temporal event series of production lines. In *2018 22nd International Conference Information Visualisation (IV)*, pages 251–259. IEEE, 2018.

[47] Heike Hofmann and Marie Vendettuoli. Common angle plots as perception-true visualizations of categorical associations. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2297–2305, 2013.

[48] Tom Horak, Andreas Mathisen, Clemens N Klokmose, Raimund Dachselt, and Niklas Elmqvist. Vistribute: distributing interactive visualizations in dynamic multi-device setups. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 616. ACM, 2019.

[49] M Shahriar Hossain, Praveen Kumar Reddy Ojili, Cindy Grimm, Rolf Müller, Layne T Watson, and Naren Ramakrishnan. Scatter/gather clustering: flexibly incorporating user feedback to steer clustering results. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2829–2838, 2012.

[50] Lee Howorko, Joyce Maria Boedianto, Ben Daniel, et al. The efficacy of stacked bar charts in supporting single-attribute and overall-attribute comparisons. *Visual Informatics*, 2(3):155–165, 2018.

[51] Samuel Huron, Yvonne Jansen, and Sheelagh Carpendale. Constructing visual representations: investigating the use of tangible tokens. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2102–2111, 2014.

[52] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization'90*, pages 361–378. IEEE Computer Society Press, 1990.

[53] N Jardine, BD Ondov, N Elmqvist, and S Franconeri. The perceptual proxies of visual comparison. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[54] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. Touchpivot: blending wimp & post-wimp interfaces for data exploration on tablet devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2660–2671. ACM, 2017.

[55] Eser Kandogan. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 73–82. IEEE, 2012.

[56] Mandy Keck, Dietrich Kammer, and Rainer Groh. Visual version comparison of multidimensional data sets using glyphs. In *2018 IEEE Information Visualization (VIS)*. IEEE, 2018.

[57] Johannes Kehrer, Harald Piringer, Wolfgang Berger, and M Eduard Gröller. A model for structure-based comparison of many categories in small-multiple displays. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2287–2296, 2013.

[58] Alicia Key, Bill Howe, Daniel Perry, and Cecilia Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 681–684. ACM, 2012.

[59] Arkady B Khodursky, Brian J Peter, Nicholas R Cozzarelli, David Botstein, Patrick O Brown, and Charles Yanofsky. Dna microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in escherichia coli. *Proceedings of the National Academy of Sciences*, 97(22):12170–12175, 2000.

[60] Jeongmi Kim, Yong-Joon Cho, Eunsoo Do, Jaehyuk Choi, Guanggan Hu, Brigitte Cadieux, Jongsik Chun, Younghoon Lee, James W Kronstad, and Won Hee Jung. A defect in iron uptake enhances the susceptibility of cryptococcus neoformans to azole antifungal drugs. *Fungal Genetics and Biology*, 49(11):955–966, 2012.

[61] Kyungyoon Kim, John V Carlis, and Daniel F Keefe. Comparison techniques utilized in spatial 3d and 4d data visualizations: a survey and future directions. *Computers & Graphics*, 67:138–147, 2017.

[62] Younghoon Kim and Jeffrey Heer. Assessing effects of task and data distribution on the effectiveness of visual encodings. In *Computer Graphics Forum*, volume 37 of number 3, pages 157–167. Wiley Online Library, 2018.

[63] Teuvo Kohonen. Self-organizing maps: ophmization approaches. In *Artificial neural networks*, pages 981–990. Elsevier, 1991.

[64] Patrick Köthur, Mike Sips, Henryk Dobslaw, and Doris Dransch. Visual analytics for comparison of ocean model output with reference data: detecting and analyzing geophysical processes using clustering ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1893–1902, 2014.

[65] Robert Krueger, Guodao Sun, Fabian Beck, Ronghua Liang, and Thomas Ertl. Traveldiff: visual comparison analytics for massive movement patterns

derived from twitter. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 176–183. IEEE, 2016.

[66] Bum Chul Kwon, Ben Eysenbach, Janu Verma, Kenney Ng, Christopher De Filippi, Walter F Stewart, and Adam Perer. Clustervision: visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):142–151, 2017.

[67] Bum Chul Kwon and Bongshin Lee. A comparative evaluation on online learning approaches using parallel coordinate visualization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 993–997. ACM, 2016.

[68] Sehi L'Yi, Youli Chang, DongHwa Shin, and Jinwook Seo. Toward understanding representation methods in visualization recommendations through scatterplot construction tasks. In *Computer Graphics Forum*, volume 38 of number 3, pages 201–211. Wiley Online Library, 2019.

[69] Sehi L'Yi, Bongkyung Ko, DongHwa Shin, Young-Joon Cho, Jaeyong Lee, Bohyoung Kim, and Jinwook Seo. Xclusim: a visual analytics tool for interactively comparing multiple clustering results of bioinformatics data. *BMC bioinformatics*, 16(11):S5, 2015.

[70] Axel Larsen and Claus 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, 1998.

[71] Sukwon Lee, Sung-Hee Kim, and Bum Chul Kwon. Vlat: development of a visualization literacy assessment test. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):551–560, 2017.

[72] Alexander Lex, Hans-Jorg Schulz, Marc Streit, Christian Partl, and Dieter Schmalstieg. Visbricks: multiform visualization of large, inhomogeneous data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2291–2300, 2011.

[73] Alexander Lex, Marc Streit, Ernst Kruijff, and Dieter Schmalstieg. Caleydo: design and evaluation of a visual analysis framework for gene expression data in its biological context. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64. IEEE, 2010.

[74] Alexander Lex, Marc Streit, Christian Partl, Karl Kashofer, and Dieter Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1027–1035, 2010.

[75] Alexander Lex, Marc Streit, H-J Schulz, Christian Partl, Dieter Schmalstieg, Peter J Park, and Nils Gehlenborg. Stratomex: visual analysis of large-scale

heterogeneous genomics data for cancer subtype characterization. In *Computer graphics forum*, volume 31 of number 3pt3, pages 1175–1184. Wiley Online Library, 2012.

[76] Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. Embeddingvis: a visual analytics approach to comparative network embedding inspection. *arXiv preprint arXiv:1808.09074*, 2018.

[77] Daniel Limberger, Matthias Trapp, and Jürgen Döllner. In-situ comparison for 2.5 d treemaps. In *Proceedings of the 10th International Conference on Information Visualization Theory and Applications 2019 (IVAPP/VISIGRAPP)*, 2019.

[78] Dongyu Liu, Fangzhou Guo, Bowen Deng, Huamin Qu, and Yingcai Wu. Egocomp: a node-link-based technique for visual comparison of ego-networks. *Information Visualization*, 16(3):179–189, 2017.

[79] Xiaotong Liu and Han-Wei Shen. The effects of representation and juxtaposition on graphical perception of matrix visualization. In *Proceedings of the 33rd annual ACM conference on Human factors in computing systems*, pages 269–278. ACM, 2015.

[80] Marıa-Jesús Lobo, Caroline Appert, and Emmanuel Pietriga. Animation plans for before-and-after satellite images. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1347–1360, 2018.

[81] Marıa-Jesús Lobo, Emmanuel Pietriga, and Caroline Appert. An evaluation of interactive map comparison techniques. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3573–3582. ACM, 2015.

[82] M Loorak, Melanie Tory, and Sheelagh Carpendale. Changecatcher: increasing inter-author awareness for visualization development. In *Computer Graphics Forum*, volume 37 of number 3, pages 51–62. Wiley Online Library, 2018.

[83] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. Show me: automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 2007.

[84] Adrian Maries, Nathan Mays, MeganOlson Hunt, Kim F Wong, William Layton, Robert Boudreau, Caterina Rosano, and G Elisabeta Marai. Grace: a visual comparison framework for integrated spatial and non-spatial geriatric data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2916–2925, 2013.

[85] Gautier Marti, Philippe Donnat, Frank Nielsen, and Philippe Very. Hcmapper: an interactive visualization tool to compare partition-based flat clustering extracted from pairs of dendrograms. *arXiv preprint arXiv:1507.08137*, 2015.

[86] Matlab. 2018. URL: `https://www.mathworks.com/products/matlab.html`.

[87] Gonzalo Gabriel Méndez, Uta Hinrichs, and Miguel A Nacenta. Bottom-up vs. top-down: trade-offs in efficiency, understanding, freedom and creativity with infovis tools. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 841–852. ACM, 2017.

[88] Gonzalo Gabriel Méndez, Miguel A Nacenta, and Sebastien Vandenheste. Ivolver: interactive visual language for visualization extraction and reconstruction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4073–4085. ACM, 2016.

[89] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–24. IEEE, 2017.

[90] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. Formalizing visualization design knowledge as constraints: actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[91] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.

[92] Yuichi Naragino and Kazuo Misue. Visualization techniques for the comparative analysis of weighted free trees. In *2017 21st International Conference Information Visualisation (IV)*, pages 45–51. IEEE, 2017.

[93] Christina Niederer, Holger Stitz, Reem Hourieh, Florian Grassinger, Wolfgang Aigner, and Marc Streit. Taco: visualizing changes in tables over time. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):677–686, 2017.

[94] Brian Ondov, Nicole Jardine, Niklas Elmqvist, and Steven Franconeri. Face to face: evaluating visual comparison. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):861–871, 2018.

[95] Origin. 2018. URL: `https://www.originlab.com/Origin`.

[96] Eli Packer, Peter Bak, Mikko Nikkilä, Valentin Polishchuk, and Harold J Ship. Visual analytics for spatial clustering: using a heuristic approach for

guided exploration. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2179–2188, 2013.

[97] Gaynor S Paton and Jonathan Henderson. Visualization, interpretation, and cognitive cybernetics. *Interpretation*, 3(3):SX41–SX48, 2015.

[98] Alexander Pilhöfer, Alexander Gribov, and Antony Unwin. Comparing clusterings using bertin's idea. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2506–2515, 2012.

[99] Matthew D Plumlee and Colin Ware. Zooming versus multiple window interfaces: cognitive costs of visual comparisons. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(2):179–209, 2006.

[100] Zening Qu and Jessica Hullman. Keeping multiple views consistent: constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):468–477, 2017.

[101] R. 2018. URL: `https://www.r-project.org/`.

[102] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Hollerer, Eun Kyoung Choe, et al. Chartaccent: annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 230–239. IEEE, 2017.

[103] George Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. Animated visualization of multiple intersecting hierarchies. *Information Visualization*, 1(1):50–65, 2002.

[104] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[105] Puripant Ruchikachorn and Klaus Mueller. Learning visualizations by analogy: promoting visual literacy through visualization morphing. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1028–1044, 2015.

[106] Raja R Sambasivan, Ilari Shafer, Michelle L Mazurek, and Gregory R Ganger. Visualizing request-flow comparison to aid performance diagnosis in distributed systems. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2466–2475, 2013.

[107] Alper Sarikaya and Michael Gleicher. Scatterplots: tasks, data, and designs. *IEEE Transactions on Visualization and Computer Graphics*, (1):402–412, 2018.

[108] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. Revision: automated classification, analysis and redesign

of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 393–402. ACM, 2011.

[109] Johanna Schmidt, M Eduard Gröller, and Stefan Bruckner. Vaico: visual analysis for image comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, 2013.

[110] Michael Sedlmair, Andrada Tatu, Tamara Munzner, and Melanie Tory. A taxonomy of visual cluster separation factors. In *Computer Graphics Forum*, volume 31 of number 3pt4, pages 1335–1344. Wiley Online Library, 2012.

[111] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information visualization*, 4(2):96–113, 2005.

[112] Jinwook Seo and Ben Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, 2002.

[113] John Sharko, Georges G Grinstein, Kenneth A Marx, Jianping Zhou, Chia-Ho Cheng, Shannon Odelberg, and Hans-Georg Simon. Heat map visualizations allow comparison of multiple clustering results and evaluation of dataset quality: application to microarray data. In *2007 11th International Conference Information Visualization (IV'07)*, pages 521–526. IEEE, 2007.

[114] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier, 2003.

[115] Martin Smuts, Brenda Scholtz, and Andre P Calitz. Usability guidelines for designing information visualisation tools for novice users. In *Beyond development. Time for a new ICT4D paradigm? Proceedings of the 9th IDIA conference (IDIA2015)*, pages 148–162, 2015.

[116] Hyunjoo Song, Jeongjin Lee, Tae Jung Kim, Kyoung Ho Lee, Bohyoung Kim, and Jinwook Seo. Gazedx: interactive visual analytics framework for comparative gaze analysis with volumetric medical images. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):311–320, 2016.

[117] Florian Spechtenhauser and Harald Piringer. Diffpin: interactive specification of references for comparative small-multiple displays. In *EuroVis (Short Papers)*, pages 107–111, 2018.

[118] Arjun Srinivasan, Matthew Brehmer, Bongshin Lee, and Steven M Drucker. What's the difference?: evaluating variations of multi-series bar charts for visual comparison tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 304. ACM, 2018.

[119] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[120] Tableau software. 2018. URL: `http://www.tableau.com/`.

[121] Justin Talbot, Vidya Setlur, and Anushka Anand. Four experiments on the perception of bar charts. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2152–2160, 2014.

[122] Hagen Tarner, Veit Frick, Martin Pinzger, and Fabian Beck. Exploring visual comparison of multivariate runtime statistics. *COLLECTIONS*, 429:2, 2008.

[123] Martijn Tennekes and Edwin de Jonge. Tree colors: color schemes for tree-structured data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2072–2081, 2014.

[124] Christian Tominski. Comparing: reducing costs of visual comparison. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, pages 137–141. Eurographics Association, 2016.

[125] Christian Tominski, Camilla Forsell, and Jimmy Johansson. Interaction support for visual comparison inspired by natural behavior. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2719–2728, 2012.

[126] Edward R Tufte. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 2001.

[127] Christopher W Tyler. *Human symmetry perception and its computational analysis*. Psychology Press, 2003.

[128] Stef van den Elzen and Jarke J van Wijk. Small multiples, large singles: a new approach for visual data exploration. In *Computer Graphics Forum*, volume 32 of number 3pt2, pages 191–200. Wiley Online Library, 2013.

[129] Roy Van Pelt, Rocco Gasteiger, Kai Lawonn, Monique Meuschke, and Bernhard Preim. Comparative blood flow visualization for cerebral aneurysm treatment assessment. In *Computer Graphics Forum*, volume 33 of number 3, pages 131–140. Wiley Online Library, 2014.

[130] Cornelis Joost Van Rijsbergen. Foundation of evaluation. *Journal of documentation*, 30(4):365–373, 1974.

[131] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. Seedb: efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment*, 8(13):2182–2193, 2015.

[132] Ivan Viola and Tobias Isenberg. Pondering the concept of abstraction in (illustrative) visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(9):2573–2588, 2017.

[133] Athanasios Vogogias, Jessie Kennedy, Daniel Archambault, Benjamin Bach, V Anne Smith, and Hannah Currant. Bayespiles: visualisation support for bayesian network structure learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(1):5, 2018.

[134] Tatiana von Landesberger. Insights by visual comparison: the state and challenges. *IEEE computer graphics and applications*, 38(3):140–148, 2018.

[135] Jagoda Walny, Samuel Huron, and Sheelagh Carpendale. An exploratory study of data sketching for visual representation. In *Computer Graphics Forum*, volume 34 of number 3, pages 231–240. Wiley Online Library, 2015.

[136] Weixin Wang, Hui Wang, Guozhong Dai, and Hongan Wang. Visualization of large hierarchical data by circle packing. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 517–520. ACM, 2006.

[137] Yong Wang, Hammad Haleem, Conglei Shi, Yanhong Wu, Xun Zhao, Siwei Fu, and Huamin Qu. Towards easy comparison of local businesses using online reviews. In *Computer Graphics Forum*, volume 37 of number 3, pages 63–74. Wiley Online Library, 2018.

[138] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[139] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012.

[140] Graham Wills and Leland Wilkinson. Autovis: automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[141] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Towards a general-purpose query language for visualization recommendation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 4. ACM, 2016.

[142] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, (1):1–1, 2016.

[143] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager 2: augmenting visual analysis with partial view specifications. In *Pro-*

ceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pages 2648–2659. ACM, 2017.

[144] Loutfouz Zaman, Wolfgang Stuerzlinger, and Christian Neugebauer. Mace: a new interface for comparing and editing of multiple alternative documents for generative design. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, pages 67–76. ACM, 2017.

[145] Haipeng Zeng, Hammad Haleem, Xavier Plantaz, Nan Cao, and Huamin Qu. Cnncomparator: comparative analytics of convolutional neural networks. *arXiv preprint arXiv:1710.05285*, 2017.

[146] Qiuju Zhang, Aidan Slingsby, Jason Dykes, Jo Wood, Menno-Jan Kraak, Connie A Blok, and Rein Ahas. Visual analysis design to support research into movement and use of space in tallinn: a case study. *Information Visualization*, 13(3):213–231, 2014.

[147] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[148] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. Matrixwave: visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 259–268. ACM, 2015.

[149] Mingqian Zhao, Yijia Su, Jian Zhao, Shaoyu Chen, and Huamin Qu. Mobile situated analytics of ego-centric network data. In *SIGGRAPH Asia 2017 Symposium on Visualization*, page 14. ACM, 2017.

[150] Jianping Zhou, Shawn Konecni, and Georges Grinstein. Visually comparing multiple partitions of data with applications to clustering. In *Visualization and Data Analysis 2009*, volume 7243, 72430J. International Society for Optics and Photonics, 2009.

[151] Yingzhen Zhu, Jifang Yu, and Jiangqin Wu. Chro-ring: a time-oriented visual approach to represent writer's history. *The Visual Computer*, 32(9):1133–1149, 2016.

국문 초록

시각적 비교는 정보 시각화를 이용한 핵심적인 데이터 분석 과정 중 하나로써, 분산되어 있는 정보들을 사람들이 서로 정리, 평가, 병합할 수 있도록 돕는다. 예를 들어, 사람들은 시간의 흐름에 따른 데이터의 변화를 보거나, 서로 다른 출처의 데이터를 비교하거나, 같은 데이터를 여러 분석 모델들을 이용해 평가하기 위해 시각적 비교 과업을 흔히 수행하게 된다. 효과적인 시각화 디자인을 위한 여러 연구가 정보 시각화 분야에서 이루어지고 있는 반면, 어떤 디자인을 통해 효과적으로 시각적 비교를 지원할 수 있는지에 대한 이해는 다음의 제약들로 인해 아직까지 불분명하다. (1) 경험적 통찰들과 실용적 설계 지침들이 파편화되어 있으며 (2) 비교 시각화를 지원하는 방법을 이해하기 위한 사용자 실험의 수가 여전히 제한적이다.

본 논문에서는 시각화 초심자들에게 효과적으로 시각적 비교를 지원하기 위한 정보 시각화 디자인 방법을 더 깊이 이해하기 위해서 일련의 세 연구를 진행하고 이에 대한 결과를 제시한다. 특별히, 시각화 초심자들이 시각적 비교를 할 때 어려움을 경험할 수 있는 두 주요 시각화 단계를 확인함으로써, 본 연구에서는 시각적 인코딩 비교 (인코딩 장벽) 및 정보 비교 (해석 장벽) 과업들에 초점을 맞춘다. 첫째, 비교 시각화 디자인을 제시한 문헌들($N = 104$)을 체계적으로 조사 및 분석함으로써 시각화 연구자들이 사용자 실험과 시각화 설계 과정을 통해 얻은 실용적 통찰들을 정리하였다. 이 문헌조사를 기반으로 비교 시각화 설계에 대한 지침들을 정립하고, 비교 시각화를 위한 디자인 공간을 더 깊이 이해하고 탐색하는 데 도움을 줄 수 있는 시각화 분류 및 예시들을 제공한다. 둘째, 초심자들이 시각화 추천 인터페이스에서 어떻게 새로운 시각적 인코딩들을 서로 비교하고 사용하는지에 대한 이해를 돕기 위해 사용자 실험($N = 24$)을 수행하였다. 이 실험의 결과를 기반으로, 초심자들의 주요 어려움들과 이들을 해결하기 위한 디자인 지침들을 제

시한다. 셋째, 생명정보학자가 시각적으로 다수 개의 클러스터링 결과들을 비교 및 분석할 수 있도록 도와주는 시각화 시스템, XCluSim을 디자인하고 구현하는 디자인 스터디를 수행하였다. 사례 연구를 통해 실제로 생명정보학자가 XCluSim을 이용하여 많은 클러스터링 결과들을 쉽게 비교 및 평가할 수 있다는 것을 보였다. 마지막으로, 이 세 연구 결과들을 기반으로 비교 시각화 분야에서 유망한 향후 연구들을 제시한다.

주요어: 정보 시각화; 비교 분석; 시각적 비교; 시각화 초심자; 문헌 조사; 사용자 실험; 설계 연구

학번: 2013-23127

감사의 글

6년 반이라는 긴 기간 동안 저의 서울대학교 컴퓨터공학부에서의 경험은 헤아릴 수 없는 감사의 시간이었습니다. 공학도로서 필요한 수많은 지식을 쌓을 수 있었던 배움의 시간이었음과 동시에 다양한 일들을 접하고 맡아볼 수 있었던 훈련의 시간이었습니다. 또한, 저의 부족한 부분들을 채워주는 여러 은인을 만날 수 있었던 감사의 시간이기도 합니다. 이 글을 통해 제가 대학원에 입학하고 졸업하기까지 함께해 주신 분들에게 제한적이게 나마 감사하는 마음을 전하고자 합니다.

누구보다도 저의 지도교수님이신 서진욱 교수님께 가장 큰 감사를 드립니다. 학생들을 열정적으로 지도해주시고 사려 깊게 챙겨주시는 지도교수님을 만난 것은 저에게 이루 말할 수 없는 복이었습니다. 교수님께서 저를 믿고 받아주신 덕분에 제가 바라던 분야의 연구를 시작할 수 있었고, 교수님의 따뜻한 배려와 인내로 인해 연구자로서 부족했던 제가 계속 성장할 수 있었습니다. 또한, 학업에 대해서 뿐만 아니라 삶에 대한 진심 어린 조언과 가르침을 주심으로 제가 많은 것들을 배우고 졸업할 수 있었습니다. 저도 언젠가는 교수님을 본받아 진실된 마음으로 학생들을 지도해주는 훌륭한 스승이 되길 소망해 봅니다. 저와 여러 연구를 함께 하셨던 김보형 교수님께도 큰 감사를 드립니다. 논문 쓰는 법과 연구 방향 설정에 대한 교수님의 꼼꼼한 지도 덕분에 연구를 어떻게 수행해야 하는지에 대해 배울 수 있었습니다.

많은 시간을 저와 동고동락한 연구실 분들에게도 감사의 마음을 전합니다. 먼저, 함께 연구하며 저의 부족함을 채워주신 공저자분들에게 감사를 표합니다. 연구실 선배이신 Kyle Koh형은 대학원 초반에 저와 연구 프로젝트를 같이 하며 열정적으로 저를 지도해 주셨고, 이때의 경험이 제가 연구를 하는데 중요한 밑바탕이 되었습니다. 조재민은 저의 연구실 동료이자 기숙사 룸메이트로서 오랜 시간을 함께 밀접하게 지내며 연구 내외적으로 정말 많은 것들을 배울 수 있게 해주었고,

여러 방면에서 값진 도움을 주었으며, 함께 즐거운 추억들도 많이 쌓았습니다. 신동화와는 긴 기간 연구과제에 함께 참여하며 슬픔과 기쁨을 같이했고, 힘든 일이 있을 때 서로 의지하며 취미생활도 공유하며 즐거운 시간을 보냈습니다. 이제는 미국에서 연구하고 있을 장유리와는 대학원의 처음과 끝을 함께 연구하며 많은 연구 얘기들을 나눌 수 있었습니다.

저와 대학원 생활을 함께해준 연구실 분들에게도 감사를 드립니다. 제가 연구실에 잘 적응 할 수 있도록 도와주신 이형민 형, 온화한 카리스마를 가지고 조언을 주시던 송현주 형, 함께 유럽에서 자전거를 타며 좋은 추억을 만든 정대경 형, 초기 연구실 세팅에 도움을 준 고봉경, 같이 운동하며 체력의 중요성을 일깨워준 박헌진 형, 육아 중에도 단기간 만에 연구 실적을 내고 졸업하신 멋진 최고은 누나, 같이 미국 서부 여행을 하며 즐겁게 지낸 전재호, 항상 연구실에서 묵묵히 연구하시던 모습이 큰 자극이 되었던 유승훈 형, 동갑내기 동네 친구로서 함께 연구실을 오가며 재밌는 대화를 나눈 김영호, 일본에서 늦은 저녁까지 학회 발표 준비에 도움을 준 김이은, 항상 밝고 유쾌함을 잃지 않는 황정민, 연구실에서 새로운 연구 분야를 개척하고 졸업한 김원재, 노련함이 돋보이는 민구봉 형, 취미 생활을 공유하며 재밌는 대화를 나눈 이용석 형, 적극적으로 발표 및 연구에 대한 건설적인 의견을 공유해준 한구현, 연구실 짝꿍으로써 연구 얘기를 편히 나눌 수 있어서 좋았던 채한주 형, 연구실에서 맡은 일들을 항상 꼼꼼하게 진행하던 점을 본받고 싶은 복진욱, 산책하며 인생 얘기를 나눌 수 있어서 좋았던 김영택 형, 긴 기간 연구실을 이끌어가게될 김재영, 연구실 분위기를 한껏 밝게 만들어준 재간둥이 김준회, 조용히 그러나 정석적으로 연구를 수행하며 좋은 실적을 내고 졸업하는 김민지, 연구와 삶에 대한 뜨거운 열정을 본받고 싶은 최길웅, 바쁜 와중에도 기꺼이 영어 대화 연습에 도움을 준 Dung Ho, 연구를 위해 필요한 새로운 지식들을 금방 배우고 써먹는 박관모, 앞으로 연구실을 이끌어갈 시각화 연구자 정석원, 엄청난 개발 결과들을 짧은 시간 안에 만들어 내는 점이 놀라운 안단테, 연구에 대한 열정을 보이며 성실히 노력하는 모습이 보기 좋은 고형권, 이 모든 분들 덕분에 저의 대학원 생활이 즐거웠고, 더불어 제가 많은 것들을 배우며 성장할 수 있었습니다.

우리 가족들에게도 감사의 마음을 전합니다. 저에 대한 가족들의 변함 없는 지지와 후원을 통해 제가 힘들고 어려울 때에도 꿋꿋이 연구를 이어갈 수 있었습니다. 우선 저의 부모님께 가장 깊은 감사를 드립니다. 두 분의 전폭적인 신앙적, 재정적 지원을 통해 제가 편안한 마음으로 대학원 생활에 집중할 수 있었습니다. 또한, 제가 어떤 상황에 있던지 '할 수 있다'는 자신감을 잃지 않을 수 있었던 것은 두 분께서 항상 저를 위해 열심히 기도해주신다는 사실을 제가 잘 알고 있기 때문입니다. 이제는 물리학도로서 연구를 하는 저의 동생, 이철희에게도 고마움을 표합니다. 동생의 지속적인 응원을 통해 자신감을 잃지 않았고, 이따금 재밌는 시간을 함께 보내며 순간의 힘든 일들을 잊을 수 있었습니다.

저의 새로운 부모님이신 장인 장모님께도 깊은 감사를 전합니다. 항상 저를 반갑게 맞아주시는 두 분 덕분에 외로울 수 있는 대학원 생활이 풍요로울 수 있었습니다. 저의 상황들을 세심하게 신경 써 주시고 물심양면으로 지원해주심에 여러 바쁜 일정들을 평안하고 건강히 소화할 수 있었습니다. 저의 처제와 동서, 이윤혜와 신재식에게도 감사의 뜻을 표합니다. 바쁜 상황에서도 서슴없이 도와주고 진심으로 응원해주는 두 분 덕분에 대학원을 마무리하는 기간 동안 마음이 든든할 수 있었습니다. 지켜보는 것만으로도 마음의 위로가 되어준 귀여운 두 조카, 신주담과 신윤아에게도 고마움을 표합니다.

마지막으로 저의 사랑하는 아내, 이지혜에게 감사의 마음을 전합니다. 광야와 같은 대학원 생활에서 그녀를 만난 것은 하나님께서 제게 주신 선물이었습니다. 그녀는 저의 긴 대학원 생활을 가까이에서 함께해주며, 같이 즐거운 시간을 보내며 힘든 일들을 잊게 해준 저의 가장 친한 친구이자, 지겨울 수도 있는 저의 각종 고민거리를 진실되게 들어주고 고민해준 참된 조언자이며, 저를 위해 열심히 기도해주는 믿음의 동반자였습니다.

2019년 12월

이세희 올림