Utilizing Color for Perceptually-Driven Data Visualization

By

Danielle Nicole Albers Szafir

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2015

Date of final oral examination: 07/07/2015

The dissertation is approved by the following members of the Final Oral Committee:
Michael Gleicher (Chair), Professor, Computer Sciences
Steve Franconeri, Associate Professor, Psychology, Northwestern University
Bilge Mutlu, Associate Professor, Computer Sciences
Kevin Ponto, Assistant Professor, Computer Sciences & Design Studies
Robert Roth, Assistant Professor, Geography

To Dan for his unwavering patience and support.

ACKNOWLEDGMENTS

The work in this dissertation would not have been possible without the mentorship and support of many amazing people. I first want to thank my committee for their guidance and feedback over the course of this work. I would especially like to thank my advisor, Michael Gleicher, for teaching me so much over these last six years and encouraging me to pursue a Ph.D. His guidance and thoughtful mentorship has made this work possible.

I would also like to thank my colleagues in the Visual Computing Lab, especially Eric Alexander, Michael Correll, Nathan Mitchell, Adrian Mayorga, Alper Sarikaya, and Brandon Smith, for always providing at least one extra pair of eyes whenever needed, for the many unforgettable discussions, and for providing the occasional much-needed distraction.

None of the work discussed in this dissertation was conducted in isolation. I would like to thank my coauthors both on the work discussed here and on the other publications I have had the pleasure to be a part of. I would especially like to thank Michael Correll for our many hours working together to understand visual aggregation, Steve Franconeri for his enthusiasm for bridging perception and visualization, and my mentors at Tableau, Maureen Stone and Vidya Setlur, whose efforts and advice were significant factors in many of these projects and who also taught me that sometimes the best way to understand a study is to stop agonizing over details and just run it. Many thanks to the collaborators who worked with me to provide many of the case studies discussed in this dissertation, especially Colin Dewey, Catherine DeRose, and Robin Valenza, who all worked with to help broaden my understanding of their domains. Specifically, Chapters 3 through 8 were written with Michael Gleicher, Chapter 4 with Michael Correll, Chapter 5 with Colin Dewey, Alper Sarikaya, and Julie Mitchell, Chapter 6 also with Alper, Chapter 7 with Maureen Stone, and Chapter 8 with Maureen and Vidya Setlur.

I am grateful to my parents for their support and encouraging me to study technology from an early age, and to my brothers, Zach and Jace, for always reminding me to balance work and play. I would also like to thank my friends, especially Tiffany Lowe, Jason Power, Emily Kawaler, Ben Farley, and the folks in the HCI lab, and teammates, especially Kajsa Jackson, Charley Henckler, Caylie O'Neil, and Allie Danielson, for keeping me sane and for many unforgettable memories during my time in Madison. I am grateful for my feline companions who kept me company when I found myself working late and for sitting on the keyboard

to force much needed breaks. I especially would like to thank my husband, Dan, for his unwavering support, for keeping our life moving during deadline rushes, for our inadvertent late night discussions about experimental design, for his patient feedback, and for always knowing how to make me laugh no matter the circumstances.

The work in this dissertation was generously funded by BACTER and NSF grants IIS-1162037 and CMMI-0941013.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

UTILIZING COLOR FOR PERCEPTUALLY-DRIVEN DATA VISUALIZATION

Danielle Nicole Albers Szafir

Under the supervision of Professor Michael Gleicher
At the University of Wisconsin-Madison

Visualization allows viewers to explore large collections of data. Effective visualizations must support viewers in understanding data both at high-levels to investigate "big picture" statistics, patterns, and trends, and at low-levels to examine individual values. Visualization design guidelines currently focus on how designs can support low-level tasks, such as determining if one value is larger than another, but far less is known about designing for high-level tasks. High-level tasks require viewers to aggregate information across multiple datapoints, such as estimating the average value of a set of points. Systems can explicitly compute these values, but must know the task and data that viewers are interested in in advance to do so. Instead, viewers frequently need to *visually aggregate* information across multiple datapoints in a visualization. However, designs that are effective for low-level tasks may not support visual aggregation, especially as datasets increase in size and complexity. To remain effective at scale, visualizations must consider how designs can support estimates both across multiple values (visual aggregation tasks) and between individual values (low-level tasks).

This dissertation describes a set of experiments, metrics, and techniques that allow visualizations to more effectively support both high- and low-level tasks by using color. To support high-level tasks, I identify limitations that inhibit visual aggregation in existing visualization designs and introduce novel designs using color to overcome these limitations. I show how different decisions made in creating a visualization can support visual aggregation. I embody these results in visualization systems that increase the size of datasets analysts can explore for three different domains. I address challenges of using color for low-level tasks by generating metrics and guidelines for color encoding design tailored to visualization. I first show how visualization designs can improve perceptions of shadowed colors in surface visualizations. I then model how two factors of visualization viewing (viewing enviroment and mark size) influence color encoding perceptions in practice and show how these models can be used to guide effective encoding design. The main technical contributions of this dissertation include a method for task-driven aggregation for one-dimensional data, novel visualization

systems for analyzing data in genomics, text analysis, and structural biology, and a data-driven method for modeling perceived color differences.

Michael Gleicher

ABSTRACT

Visualization allows viewers to explore large collections of data. Effective visualizations must support viewers in understanding data both at high-levels to investigate "big picture" statistics, patterns, and trends, and at low-levels to examine individual values. Visualization design guidelines currently focus on how designs can support low-level tasks, such as determining if one value is larger than another, but far less is known about designing for high-level tasks. High-level tasks require viewers to aggregate information across multiple datapoints, such as estimating the average value of a set of points. Systems can explicitly compute these values, but must know the task and data that viewers are interested in in advance to do so. Instead, viewers frequently need to *visually aggregate* information across multiple datapoints in a visualization. However, designs that are effective for low-level tasks may not support visual aggregation, especially as datasets increase in size and complexity. To remain effective at scale, visualizations must consider how designs can support estimates both across multiple values (visual aggregation tasks) and between individual values (low-level tasks).

This dissertation describes a set of experiments, metrics, and techniques that allow visualizations to more effectively support both high- and low-level tasks by using color. To support high-level tasks, I identify limitations that inhibit visual aggregation in existing visualization designs and introduce novel designs using color to overcome these limitations. I show how different decisions made in creating a visualization can support visual aggregation. I embody these results in visualization systems that increase the size of datasets analysts can explore for three different domains. I address challenges of using color for low-level tasks by generating metrics and guidelines for color encoding design tailored to visualization. I first show how visualization designs can improve perceptions of shadowed colors in surface visualizations. I then model how two factors of visualization viewing (viewing enviroment and mark size) influence color encoding perceptions in practice and show how these models can be used to guide effective encoding design. The main technical contributions of this dissertation include a method for task-driven aggregation for one-dimensional data, novel visualization systems for analyzing data in genomics, text analysis, and structural biology, and a data-driven method for modeling perceived color differences.

1 INTRODUCTION

Visualization allows analysts to directly see structure in information. As the amount of data available to analysts grows, visualization has become an increasingly critical part of the analyst's toolbox. It supports dynamic exploration and pattern finding in large and complex datasets without necessitating a priori questions or hypotheses. However, current data visualization tools have often limited scalability. Many systems and approaches are bound in terms of how much data they can communicate at a given time, the complexity of individual datapoints, or the number of questions (or *tasks*) that analysts can address.

How we design a visualization directly informs which of these dimensions of scale a visualization can address. Studies in graphical perception, such as those of Cleveland and McGill [1984], experimentally generate guidelines for designing visualizations by measuring how accurately different visual encodings of data support viewers in completing different tasks. Most of these guides focus on tasks comparing individual values (e.g. how much larger is value a than value b). It is unclear how well this kind of point-level task supports analysts in examining millions of datapoints. It is inefficient to explore every datapoint in a display, and analysts are limited in their ability to remember complex datapoints [Alvarez and Cavanagh, 2004]. Instead, analysts first need to develop an understanding of the data in aggregate in order to locate interesting values to explore in greater detail. For visualizations to remain effective at scale, they need to consider both aggregate judgments that communicate the "big picture" and point judgments that support specific comparisons within the data. In this dissertation, I will show how designers can effectively use color to support analyses at both of these scales.

Aggregate analyses generally require the analyst to combine information from multiple data values. This aggregation can be done in two ways: computationally or visually. *Computational aggregation* uses statistics to compute the aggregate value of interest, such as the average or line of best fit across collection of points, and then visualizes the computed value. This approach requires the viewer to know ahead of time which values and statistics they are interested in. Alternatively, *visual aggregation* occurs when the viewer visually combines data to estimate aggregate information. For example, viewers can visually estimate the average position [Gleicher et al., 2013b] and correlation [Rensink and Baldridge, 2010] between points from a scatterplot. Visual aggregation can be done flexibly (the

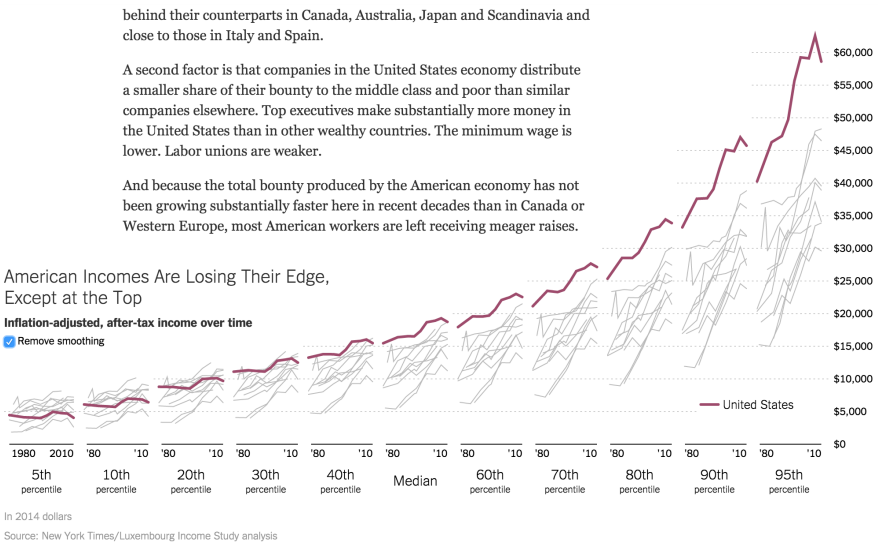
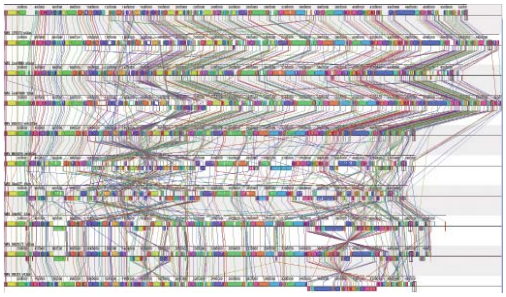


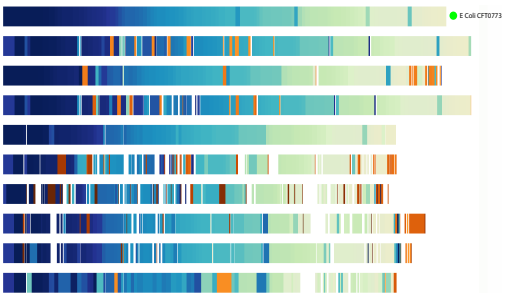
Figure 1.1: Mean income for the United States (red) and 11 other countries (grey) between 1980 and 2010[1]. Comparisons between the average orientation and position for the grey countries and the U.S. represent global changes in economic status. While line graphs encode change over time well for one country, they quickly become cluttered for 11 and make visual aggregation difficult.

viewer does not need to explicitly specify points or tasks of interest) and efficiently (the viewer does not need to input any information and many values can be estimated at a glance).

Here, I focus on visual aggregation as a way to inform visualization designs that help analysts make sense of data as the dataset scales. At a high-level, visualization design can support different *visual aggregation tasks*—tasks that combine information across multiple datapoints, such as estimating numerosity, averages, or outliers. These tasks are commonly used in data analysis, but designers do not necessarily take advantage of them when creating a visualization. They often choose encodings that designed for point judgments, without considering how they will be effective for aggregate tasks. For example, data journalists often use visualization to pursuade readers. Visual aggregation come into play when comparisons between different collections of data provide evidence of a point. Figure 1.1 is intended to show that the incomes for middle class families in the United States are not growing as quickly as in the rest of the world. The supporting visualization compares rates of change in income between the U.S. (red) and the average rate of change of other countries (grey). To compare these values, viewers must visually aggregate the vertical position and orientation of each set of grey lines and compare the results to the orientation and position of the red lines. In



(a) Eight *E. Coli* genomes visualized using Mauve (Darling et al., 2004)



(b) The same data visualized using color (Section 5.1)

Figure 1.2: Biologists want to understand how genetic material moves around between organisms to understand high-level structural and functional patterns. (a) Connection supports this task at small scales, but (b) color better supports this task at larger scales.

this example, a design suitable for point tasks break down for visual aggregation. While line graphs may encode data for individual countries effectively, the display becomes cluttered for 11 countries, complicating visual averaging.

Visual aggregation is also important for exploring data in specialized domains. For example, a biologist might want to understand what genetic material is common across a set of genomes. They often explore this data using visualizations that draw a line to connect matching genes (Fig. 1.2a). Connection is very good for encoding relationships between small numbers of datapoints. When the amount of data scales up, this encoding breaks down quickly: it is difficult to make aggregate estimates of how genetic material is shared between genomes. By instead using color, designs can better support aggregate judgments about the same data (Fig. 1.2b).

Choropleth maps can encode high-level geographic patterns using color. For example, Figure 1.3 helps viewers explore poverty geographically. The map uses semantic zoom to support precise judgments for individual geographic regions. The overview encodes information by county. Analysts can visually detect outlier counties and estimate averages across different regions (e.g. poverty is higher in the Southeast than the Upper Midwest). However, this aggregation removes potentially important details. For example, cities often have interesting internal variations (as seen in the thumbnails below the map) that are lost at the county level. It also limits our ability to understand the extent of these patterns. A county with high poverty at its center but low poverty on its outside will appear identical to one with uniform average poverty. Encoding data at a smaller granularity, such

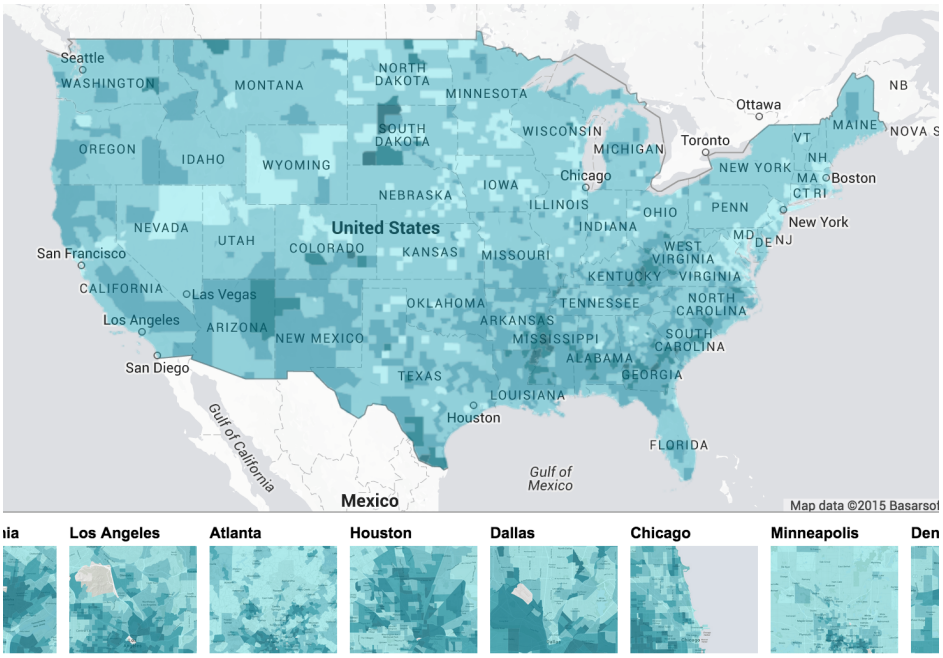


Figure 1.3: A choropleth map[2]showing the percetage of the population living below the poverty line in the United States. Color is mapped to county, which simplifies precise judgments at that scale, but hides potentially important patterns at smaller scales. Reducing the granularity of the data would reveal these patterns and may still support (or even improve) performance for visual aggregation tasks.

as that shown in the city views, would allow viewers to explore patterns at a finer level of detail but may decrease viewers' abilities to precisely estimate individual values. In this dissertation, I will show that this trade-off may be worthwhile for understanding data at a high level: the visual system can effectively aggregate color to make sense of large, dense datasets.

In the first half of this dissertation, I consider how designers can use color to create visualizations that support aggregate tasks at scale. I outline how recent findings from visual perception can help designers understand how different visualizations may (or may not) support visual aggregation for large datasets. These findings suggest that designs that are effective for precise judgments between a small number of points may not be effective for visual aggregation tasks. Color provides a promising alternative for designing visualizations for visual aggregation. However, color is often disregarded as being ineffective for quantitative data visualization. This guidance comes from studies of precise judgments on small datasets, where it underperforms encodings such as position and size. The ability of the visual system to readily summarize and detect aggregate structure from

Figure 1.4: Three five-step color ramps generated according to different metrics. All ramp colors should appear subtly distinct according to these metrics. Ramps based on controlled viewing conditions (top: CIELAB ΔE and center: empirical JNDs from colorimetry [Mahy et al., 1994a]) underestimate these differences. In this dissertation, I will introduce metrics that model encoding perceptions for visualization that are robust in practice for digital displays (bottom).

color has been used by artists for centuries for reasons perception can help us understand. This ability allows designers to reconsider to the effectiveness of color in visualization, with a specific focus on visual aggregation tasks.

The use of color for visual aggregation represents a trade-off. Effective visualizations must support both aggregate and point tasks—analysts generally need to understand the big picture and also examine key datapoints. To support point judgments, such as comparing two individual datapoints, encodings that represent different values should be visually distinct. Using color in visualization complicates these kinds of judgments: the visual system can only distinguish between a limited set of colors. While designers cannot increase the number of colors viewers can distinguish, they can increase the difference between colors.

Most metrics for measuring the perceived difference between colors come from colorimetry. These purpose of these metrics is to express the sensitivity of the visual system to small differences in color. As a result, they do not consider many of the complexities of viewing visualizations, such as variability in viewing environment and aspects of visualization design (e.g. context and mark size), that may impede perceptions of color-coded data. Current metrics often *underestimate* how distinct two colors will appear when applied to visualization as a result of these complexities (Fig. 1.4[3]).

[3]These metrics are designed for digital displays rather than print media. While the methods here could be used to build a robust model for print, printing may degrade apparent differences in many of the presented examples.

Some designers hand-tune color encodings to try to create encodings that have sufficiently different colors (e.g. Tableau, Brewer et al. [2003b], Samsel et al. [2015]). These approaches require extensive expertise to accomplish effectively and may also distort perceptual differences between colors. Instead, I propose a series of experiments and data-driven models that generate guidelines for using color effectively for point tasks.

In summary, current design guidance for visualization is largely based on how well viewers perform precise point judgments. In order to remain effective at scale, visualizations must consider how designs can support both estimates across multiple values (visual aggregation tasks) and between individual values (point tasks). This is challenging because designs that are effective at one scale may not be appropriate for the other. I propose novel designs using color to address this challenge that take advantage of how color is processed by the visual system to support visual aggregation. I then introduce empirical guidelines and metrics that model color difference perception under some of the complexities of visualization viewing by modeling for anticipated, rather than precise, factors in viewing visualizations.

It is the thesis of this dissertation that **color encodings in data visualization support visual aggregation and therefore increase visualization scalability, and that many limitations of using color in visualization can be mitigated through design guidelines and encoding metrics that consider encoding perception in practice**. I will prove this thesis using a combination of quantitative experimental data and qualitative insights from domain experts using visualization systems I have designed.

This dissertation will consist of two parts. In the first, I use theories from visual perception to hypothesize how color and other visual cues might help support visual aggregation tasks in practice and where limitations arise in existing designs. I will then discuss the design and evaluation of several aggregate encodings that leverage color to support these aggregate judgments at scale. Finally, I will discuss three systems I have developed that embody these ideas in practice to support data analysis at substantially larger scales than previous approaches.

In the second part, I will explore how we can mitigate limitations of color encodings for point analysis tasks by modeling practical color perception for visualization. Designers can approach this problem in two ways. First, they can manipulate aspects of a visual design to support a given color ramp. For example, the minimum mark size can be increased to support robust color comparisons.

Alternatively, color encodings can be designed to conform to the constrains of a display. For example, color differences can be increased to support a desired range of mark sizes.

I will discuss a series of studies that explore visualization design from both angles. First, I will consider how the design of a surface visualization can improve color identification in shadow. I will then introduce a method for tailoring color encoding design to different viewing populations. I extend this method to modeling color differences as a function of mark size and show how this model can be applied to improve effective visualization design.

Unless otherwise specified, this dissertation will consider color as a method that encodes data using a set of color values that vary in lightness, hue, and/or saturation. Discussions surrounding color will generally focus on encoding quantitative data using either sequential (a set of colors with an intuitive ordering) or diverging encodings (a pair of sequential encodings that share a common midpoint). The second part of this dissertation could also readily be applied to designing color encodings for categorical data.

1.1 Contributions

In this dissertation, I identify how visualization designs can use color to support both aggregate and point analysis tasks. My approach will identify relevant theory from perception to inspire novel design solutions. I will then take an application-centered approach to validate these solutions for visualization. This leads to several specific theoretical, experimental, and systems-level contributions. These contributions are as follows:

- Color for Visual Aggregation:
  - **Perception for visual aggregation:**
    I provide a high-level organization of findings from perceptual psychology that can be used to reason about visual aggregation at scale. I apply these findings to two common encodings in one-dimensional data. This organization helps identify limitations of existing encodings for visual aggregation and suggests reasons why color might be an effective alternative for visualizing information at scale. While this survey is by no means exhaustive, it provides a framework for considering percetual constraints in designing for visual aggregation tasks.

  - **Task-oriented aggregation techniques:**
    Scalability in visualization is often limited by the number of pixels available to represent data. To help overcome this limitation, I introduce a method for representing one-dimensional data to support different forms of visual aggregation. This method operates in screen space to provide the analyst with direct control over aggregation. The aggregate data is represented through perceptually-inspired glyphs designed to support specific comparison tasks. Two of these glyphs (event striping and color weaving) represent novel designs for visualizing aggregate data.
  - **Design guidelines for visual aggregation in times series data:**
    I evaluate how three design decisions that go into creating an aggregate visualization (how data are visually encoded, what data are encoded, and how data are mapped to encoding) influence performance for two forms of visual aggregation. These studies consider how the depicted data, statistical granularity, and visual variables used to represent data all contribute to viewers' abilities to accomplish six visual aggregation tasks (maxima, minima, range, mean, variance, and outlier numerosity).
  - **Systems using color to support visual aggregation at larger scales:**
    I introduce three systems I have developed that support visual aggregation at scale for different domains (whole genome alignments, text analysis, and classifier predictions for structural biology). These systems support analysts in exploring data at greater scales than previously possible.

- Engineering Color for Point Tasks:

  - **Guidelines for using color in surface visualization:**
    Color is commonly used to represent data value on molecular surfaces. It may also be obscured by shading and shadows used to communicate surface features. I present a series of experiments that show how visualization design influences how well viewers can interpret data values encoded as shadowed colors. The results of these studies point to specific trade-offs in surface visualization design.
  - **Modeling color difference perception in practice:**
    Conventional quantitative metrics for using color in visualization are based on controlled laboratory studies that do not consider the complications of display variability and other imperfections in visualization

viewing. I introduce a data-driven method for modeling color perceptions over a target audience that gives designers control over color discriminability. I apply this method to construct color metrics for web-based visualization using crowdsourced viewers.

– **Model of color difference perception for different mark sizes:** The appearance of a color encoding depends on the context of a visualization design. It is difficult to predict many aspects of a visualization context a priori, but often size is well constrained and significantly influences viewers abilities to distinguish between encoded marks. I present a model of color discriminability as a function of size. I show how this model can be used to design effective color encodings in visualization.

2 BACKGROUND

This chapter will briefly touch on prior work relevant to this dissertation. The intention of this chapter is to contextualize the topics discussed in this dissertation and motivate the importance of the dissertation topic to visualization. Additional relevant work will be described in the ensuing chapters.

2.1 Addressing Scale in Visualization

As datasets grow in size and complexity, visualization systems must evolve to better scale to these new challenges. They must support new data analysis workflows while enabling serendipitous insights into patterns and trends. This scalability presents an interesting challenge for visualization designers: how can a visualization help people make sense of large and complex data.

One was to simplify data analysis is by explicitly computing and visualizing statistics. Statistics provide compact summaries at the expense of raw data. They also rely on understanding a priori what aspects of a dataset are interesting to the viewer. Visualization, alternatively, allows the viewer to dynamically explore large collections of data. Statistical quantitaties can then be visually estimated. These visual aggregation judgments—judgments that integrate information across multiple points—provide new questions for designers to consider when building a visualization.

A visualization must show the relevant data to the analyst in order for visual aggregation to be useful. Screen space is a barrier to this: there are a limited number of pixels that can a visualization can use. Advancements in gigapixel displays offer scalability in terms of screen space, but massive displays may be challenging for viewers to interpret [Andrews et al., 2011]. Perceptual challenges also significantly limit the effectiveness of visualization design at scale—guidelines for design may fail as the data grows [Fekete and Plaisant, 2002]. One suggestion for overcoming these challenges is using overview visualizations with simple designs that provide details and excess dimensionality only on demand. While overview visualization has been extensively explored (see Hornbæk and Hertzum [2011] for a survey), how to design overviews that support visual aggregation tasks is largely unexplored.

A primary focus of this dissertation is understanding how color can facilitate visualization at scale. There is an established tradition of using color for overview

visualization (see Keim [2000], Shneiderman [2008] for examples). Heatmaps and choropleth maps (e.g. Fig. 1.3) are two of the most common examples. Pixel-oriented visualization techniques [Endert et al., 2011, Keim, 2000, Keim et al., 2002] are also commonly used for overview because they map data compactly—individual data values to pixels—allowing visualizations to maximize their use of screen space. Other systems (e.g. Chromogram [Wattenberg and Viegas, 2010] and Lasagna Plots [Swihart et al., 2010]) use colorfields to convey change across distinct groups over time.

While these approaches scale to millions of datapoints, they break down when the number of data points exceeds the available pixels on the screen. Approaches have been proposed to address this issue. For example, Keim et al. [2007] use importance functions to spatially filter data at large scales; however, this approach relies on knowing and quantifying *a priori* what regions of the display are "important" and provide little transparency into what data is lost and how that data might influence viewers abilities to accomplish certain tasks. Several approaches computationally reduce the dataset (e.g. [Keim et al., 2007, Papadimitriou et al., 2013]), but these methods also require knowing and quantifying relevant data a priori. Other methods collapse data based on well-defined structures, such as abstracting cliques into glyphs [Dunne and Shneiderman, 2013] or abstracting data at different levels of hierarchy [Elmqvist and Fekete, 2010]. However, these methods provide little insight into aggregating information without well-defined structural forms, such as ranked lists. In this dissertation, I show how visualization can leverage the strengths of color for overview as datasets scale beyond the bounds of the display (Chapter 3) and still support higher-level, visual aggregation tasks (Chapters 4 and 5).

2.2 Graphical Perception

Visualization has a rich tradition of incorporating knowledge from perception into design (see Healey and Enns [2012a] for a survey). Most guidelines for effective visualization design come from graphical perception—the study of how well viewers can interpret different encodings of data. For example, Cleveland et al. [1985] measured how accurately viewers could estimate values encoded using different visual channels, showing the benefits of size and position over color for quantitative judgments. Healey et al. [1996] explored how pop-out could help viewers find target datapoints, showing the value of size, orientation, and color for highlighting

specific values. Heer et al. [2009] showed how different designs for time series data help viewers find and compare specific values. Haroz and Whitney [2012] evaluated how grouping like items can improve viewers' abilities to find target values.

These experiments provide a wealth of grounded guidance for designing effective visualizations at small scales. They also provide methodologies for reasoning about the effectiveness of different visualization approaches. However, they focus on tasks involving small sets of well-defined datapoints. There is a broad range of analysis tasks that a visualization might support. Understanding the space of visualization tasks helps designers understand user needs and how design insights might transfer across domains. For example, Shneiderman [1996] outlines seven abstract tasks for visualization. Andrienko and Andrienko [2006] provides a more detailed, hierarchical construction of task.

These taxonomies also provide means for reasoning about individual tasks. For example, Schulz et al. [2013] consider how five dimensions of a task might inform how visualizations might address it effectively. Roth [2012] organizes existing concepts of task based on the goals, user actions, and characteristics of data that might influence how a user interacts with a visualization to accomplish a given task. Brehmer and Munzner [2013] shows how designers might consider *why* and *how* a viewer might approach an abstract task to help tailor general design ideas across specific domains.

This dissertation characterizes tasks differently than previous approaches, dividing the space into point tasks answered using specific data values, such as searching for a well-define target or comparing individual values, versus visual aggregation tasks that combine data over a range of values, such as identifying outliers or determining the average of a set of points. This division is selected as it characterizes a broad set of tasks that are seldom studied in visualization. While the graphical perception literature discussed above applies primarily to the first class of tasks, less is known about how visualizations can support the second.

Recent work has evaluated performance for specific visual aggregation tasks, such as estimating numerosity [Correll et al., 2013, Healey and Enns, 1998], correlation [Rensink and Baldridge, 2010], and mean [Gleicher, 2013]. However, these studies evaluate how well a specific visualization design support these tasks (e.g. scatterplots and tagged text). Alternatively, some studies measure the relative performance of different visualization designs for tasks such as correlation [Harrison et al., 2014] and trend [Fuchs et al., 2013]. These studies focus on

evaluating complete designs rather than lower-level components of design. As a result, it is unclear how these results translate to designing visualizations that support visual aggregation tasks.

Research in visual perception might provide some clues as to how visualization designs can support visual aggregation. Perception research focuses on how the visual system processes abstract low-level features, such as color and position. Chapter 3 will survey a number of results from perception to show how they might inform designs that support visual aggregation at scale.

Of explicit relevance for visual aggregation is a collection of recent work on ensemble statistics. These studies show that the visual system can readily summarize visual features such as size Ariely [2001a], orientation Choo et al. [2012a], and luminance Bauer [2010]. These visual summaries may serve as a scaffold for visual aggregation tasks by providing low-level statistical summaries of relevant data.

While this work can inspire theories about effective visualization, it is difficult to use these findings directly for several reasons. Perception experiments often test judgments about a small collection of simple stimuli that are presented for very brief (often < 1 second) durations under highly controlled conditions. Visualizations, conversely, are often complex with lots of data, unlimited exposures, and are viewed under variable conditions. As a result, visualizations drawing on visual perception for design inspiration must consider how well the cited studies translate to visualizations. Designs frequently need to be evaluated in studies specific to visualization in order to truly understand their effectiveness.

In this dissertation, I will survey results from visual perception to hypothesize about effective design for visual aggregation (Chapter 3). I will evaluate these designs in the context of six visual aggregation tasks (Chapter 4) and demonstrate how they can support analysis across different visualization domains (Chapter 5).

2.3 Using Color in Visualization

Color is just one of a number of *visual variables* that can be used to encode data values in visualization. The canonical set of variables are size, texture, orientation, shape, position, value, and color, where value refers to how light or dark a color is and color refers to its hue Bertin [1983]. Work since has added additional variables, such as motion Carpendale [2003], and considered their effectiveness for different kinds of information, such as communicating uncertainty MacEachren et al.

[2012] or cartographic data Garlandini and Fabrikant [2009]. Three principle visual variables are components of the color of a mark: lightness, saturation, and hue. In visualization, hue is principally used to encode categorical values whereas lightness and saturation can be used to represent quantitative data Rogowitz and Treinish [1998]. However, color encodings frequently introduce variations in all three components to improve visual appeal Brewer et al. [2003b]. Thoughout this dissertation, the term "color" will refer to encodings that may vary across any of these three components unless otherwise specified.

While color encodings are promising for visual aggregation tasks, they are known to be less effective for performing point tasks on quantitative data [Cleveland et al., 1985]. Color suffers from a number of limitations for point tasks. For example a limited number of colors can be distinguished at a given time [Ware, 2000], simultaneous contrast can alter the appearance of color [Mittelstädt et al., 2014], and effective color ramp design is challenging [Silva et al., 2011]. Some of these biological limitations—a visualization cannot increase the number of discernable colors the viewer can perceive—but others might be addressible through encoding design.

Visualizations can address some of these issues though color encoding design. Many approaches have been proposed for effective encoding design (see Silva et al. [2011] for a survey). For example, Tominski et al. [2008] propose methods for visualizing colors based on data distributions to increase the effectiveness of color encodings for different datasets. PravdaColor [Rogowitz and Treinish, 1998] provides guidelines for ramp design based on task. ColorCAT provides color encoding design tools for supporting color-blind users [Mittelstädt et al., 2015]. ColorBrewer [Brewer et al., 2003b] provides hand-designed ramps based on best practices in cartography that are commonly used in visualization. All of these solutions can support effective encodings, but they do not provide encodings where data value correlates with perceived color difference.

Perceptual metrics for color have some utility for creating encodings that correlate differences in color to differences in value. A common metric used in visualization design is CIELAB. CIELAB represents color using three axes—lightness (L^*), red-green a^*, and blue-yellow (b^*)—that model the three opponent processes the visual system uses to detect color. CIELAB is an approximately perceptually uniform color space, meaning Euclidean distances within CIELAB should correspond to the perceived differences between colors. One unit of Euclidean distance approximately maps to one just-noticeable color difference

(JND) under calibrated conditions. Designers have used CIELAB to try to control the apparent difference between colors to try to guarantee that certain values will be discriminable and that perceived differences map to value. For quantiative data, CIELAB allows visualization designers to try to divide color space effectively: dividing a ramp too finely makes it difficult to distinguish between values while dividing too coarsely reduces the number of values that can encode data. The ability to identify a difference between colors can support visualization tasks that require viewers to compare individual values.

CIELAB has been used to guide encoding design in visualization. For example, MagnaView [Wijffelaars et al., 2008] generates perceptually-linear ColorBrewer ramps by interpolating curves between control colors in CIELCH (CIELAB expressed in polar coordinates). However, CIELAB only approximates perceived color differences—perceived differences vary across different parts of the color space [Luo et al., 2001]. More complex models provide more accurate insight into color difference (see Robertson [2007] for a survey), but the simplicity of CIELAB makes it a popular choice for visualization tools (e.g. [Cao et al., 2010, Kaski et al., 1999, Livingston et al., 2011, Wang et al., 2008]). Some encodings account for these imperfections by hand (e.g. Samsel et al. [2015]), but this process does not scale well nor does it provide quantitative guarantees of effectiveness.

A more significant limitation to using CIELAB in visualization is that it was not designed for use in visualization. CIELAB was created to gauge the sensitivity of the eye to color. It models difference perceptions under laboratory conditions: lighting, display parameters, viewing angle and distance, and surround all were highly controlled [L'Eclairage, 1978]. However, factors such as increased direct or ambient lighting [Brainard and Wandell, 1992, Oicherman et al., 2008, Rizzo et al., 2002], background and surrounding colors [Mullen, 1985, Stokes et al., 1992], the size of a mark [Carter and Silverstein, 2010], and display device [Krantz, 2001, Sarkar et al., 2010] all can substantially degrade the differentiability of colors in practice. This complicates using CIELAB directly in visualization: visualizations operate under uncontrolled conditions and the design of a visualization (and therefore the size of a mark and surrounding color composition) is generally dependent on the data. In practice, a visualization designer cannot tune their design on the fly to account for this variation. Instead, effective encoding design must anticipate the expected variation and design color encodings that are robust to those variations.

Some studies have started to consider color perception for visualization. For

example, Mittelstädt et al. [2014] uses a post-process to correct for luminance contrast in visualization design. Crowdsourced studies have explored color under variable viewing conditions. Heer and Stone [2012] models crowdsourced color naming metrics and shows their applicability to qualitative encoding design. Zuffi et al. [2009] explore how to design colors combinations that are legible online. However, the efforts provide little generalizable guidance for considering small color differences. Degradation in small color differences is especially hard to measure—errors are more likely to make a light and midgreen indistinguishable than a green and an orange. Small color differences are key for encoding quantitative values, a common case in visualization. Effective quantitative encoding design is challenging as color differences must be subtle, yet sufficiently distinct, especially for ordinal or interval data, where datapoints are broken into discrete groups.

The second part of this dissertation explores methods for mitigating limitations of color encodings in practice. It provides experiments exploring how designers can construct visualizations that better support color identification tasks (Chapter 6). I will also introduce data-driven metrics for visualization design that allow designers to account for the effect of viewing enviroments (Chapter 7) and mark size (Chapter 8) for quantitative color encodings with minimal modifications to CIELAB as currently used in visualization.

Part I

Designing for Visual Aggregation

3 UNDERSTANDING PERCEPTION FOR VISUAL AGGREGATION

As the amount of visualized data increases, our ability to *visually aggregate* information from multiple data values becomes increasing important to conducting aggregate analysis tasks. Little is known about how visualizations can support visual aggregation in practice. While recent evaluations have begun to consider visual aggregation tasks, such as estimating averages [Gleicher et al., 2013b] or correlation [Harrison et al., 2014, Rensink and Baldridge, 2010], these efforts focus on evaluating performance for specific designs. Designers can use these findings to match complete designs to tasks, but it is unclear how well these results can be used to guide new designs or to understand the limitations in different designs, especially as viewers aggregate over larger collections of values.

The visual perception literature offers a starting point for understanding how different components of visualization design might support visual aggregation. In this chapter, I identify relevant theory from visual perception to understanding visual aggregation in design and how different designs might break down as datasets grow. I then derive design inspiration from these perceptual theories to address limitations in existing visualization designs using color.

To guide this discussion, I focus on two model problems in one-dimensional data analysis (comparing gene sequences and time series data). The surveyed findings show limitations in existing designs and point to color as an effective channel for supporting aggregate judgments across large data collections. One practical limitation for using color to support visual aggregation at scale is that displays only have a fixed number of pixels that can be used to encode data. For example, genomes often contain more genes than there are pixels available to represent those genes. To address this limitation, I will present a method for aggregating one dimensional data informed by findings from visual perception. This method tailors how aggregated data is represented to support different visual aggregation tasks.

3.1 Overview

Visual aggregation tasks require viewers to combine information across multiple data points in order to complete a task. For example, in a scatterplot, tasks, such as mapping symbols to a key or knowing whether a particular data value was lower or higher than another, operate over individual values. But other types of

task, such as the approximate mean or size of an entire cloud of points, compute information about an entire *set* of points. Visualization allows an analyst to complete these tasks using their visual system as opposed to explicitly computing aggregate values.

Little is known about how accurately and efficiently a viewer can complete different visual aggregation tasks. Findings from perception offer some insight into the statistical quantities that people can visually estimate from different visual features. However, these findings are generally based on small collections of simple visual objects (e.g. circles, rectangles, oriented lines). As datasets grow, perceptual challenges not considered in these studies might arise from visualization design. These challenges might influence how well viewers can complete visual aggregation tasks for many reasons, such as requiring viewers to aggregate information across larger amounts of data. By understanding how the different visual processes might contribute to visual aggregation over large collections of datapoints, designers can reason about what encodings might be most effective for certain tasks.

In this chapter, I use one-dimensional data analysis as a model problem for understanding visual aggregation at scale. One dimensional data analysis is commonly used across a number of domains. For example, time series data has nearly ubiquitous applications for showing how things change or behave over time. Advances in genomic sequencing technologies provide biologists with an ever-expanding collection of data. Scientists can understand the similarities and differences between genetic sequences by comparing the relationship between genes in different sequences to, for example, to understand evolution, to infer common function, or identify differences. Exploring ranked data provides insight into how performance or popularity change across collections. One dimensional data provides a useful model problem for understanding visual aggregation at scale. Not only is one dimensional data used in a breadth of domains, visualizations of one dimensional data must support visual aggregation as datasets grow both in terms of the number of data sequences and the length of those sequences. I anticipate that many of these findings will be relevant to other applications and will provide preliminary evidence of this generalizability in Chapter 5.

Statistics has explored how designers can computationally analyze one dimensional data. However, these analyses are often difficult to comprehend and generally focus on questions about the data that the analyst must formulate in advance. Visual exploration provides a greater degree of flexibility in exploring information, but it is currently an analytical bottleneck—while current approaches

are useful, they have difficulty scaling to the larger datasets becoming available. Understanding how viewers perceive information in a visualization can highlight scalability limitations in existing designs and inform new systems that better scale to modern datasets. In this section, I generate an understanding of how the visual system might process a data visualization, focusing on applications in time series and whole genome alignment data.

Sequence comparison is a fundamental task in the biological sciences. Whole genome alignment supports sequence comparison by matching sets of related genes across a collection of genomes. Scientists can use this matching to understand the similarities and differences between genetic sequences by comparing the relationship between genes in different sequences. These tasks allow scientists to, for example, understand evolution, to infer common function, or identify differences. Sequence tasks may also be computed at different scales, many of which require visual aggregation to complete. Because the sequences are too long for manual examination, scientists rely on alignment tools that automatically identify subsequences that match between the sequences being compared. Tools for visualizing these alignments are commonly used in performing sequence comparison.

Whole genome alignment comparison provides an interesting model problem for understanding visual aggregation as it relies on categorical data (comparing genes). A variety of approaches for displaying and exploring alignments exist, and have been incorporated into a wide variety of tools (see Procter et al. [2010] for a survey). These designs generally fall into three categories: dot-plots comparing two genomes, synteny views comparing a handful of genomes to a reference, and connected alignment visualizations.

I focus on connected alignment visualizations as they offer the greatest scalability of these approaches in terms of both number of sequences and number of supported tasks. These visualizations represent whole genome alignment data by drawing physical connections between corresponding genes, with genomes arranged as parallel tracks (Fig. 1.2a). Connectivity explicitly encodes the relationship between two genomes, creating oriented lines that encode how these genes change position and value in the dataset similar to a parallel coordinates plot. However, these displays only allow biologists to explore up to ten genomes simultaneously. New designs are required to support the larger datasets that are of increasing interest to biologists.

Time series analysis is an extremely common form of data analysis and provides

a model problem for visualizing quantitative one-dimensional data. While many methods have been proposed for visualizing time series data (see [Aigner et al., 2008] for a survey), line graphs and their derivatives remain the canonical method of visualizing time series data. Line graphs encode quantitative data using position. For this discussion, I focus on juxtaposed line graphs due to known limitations in using superimposed lines for visualization ([Javed et al., 2010]).

One issue with position-based (e.g. line graphs) or connection-based (e.g. connected alignment visualizations) is that they tend to become too complex to be effective as the data grows. At some point, without substantial reductions in the data domain, these methods no longer support effective analysis. For example, connection-based whole genome alignment approaches are limited to roughly ten or fewer genomes at once before the display is too cluttered to read effectively. In line graphs, reducing the height of a line reduces the fidelity with which data can be represented and compress the data enough to reduce legibility.

In this chapter, I identify relevant theories from perception to understand these limitations for position and connection in one-dimensional data analysis and use these theories to drive four novel designs using color to support aggregate visual analysis tasks. To do so, I first organize literature on how complex displays are interpreted into four components of perception for visualization design (pop-out, summarization, visual search, and visual clutter) and discuss their ramifications for visualization. I describe how these principles are relevant in current visualization approaches, either in terms of how these encodings might facilitate visual aggregation tasks or in identifying scalability limitations in current approaches. From these limitations, I explore the potential benefits of using color for facilitating visual aggregation and scalability in one-dimensional data analysis and propose a method for supporting scalable visual aggregation in one-dimensional data.

While these components by no means represent all aspects of perception, for example there is no discussion of attentional selection or crowding, I believe they are sufficient to characterize the scalability limitations in existing one-dimensional visualization approaches and provide a theoretical grounding for the utility of color for visual aggregation tasks. This is by no means the first application of findings from perception to inform visualization—surveys (e.g. [Healey and Enns, 2012b]) have outlined potentially useful findings and frameworks (e.g. [Rensink, 2014]) theorize how these findings may be applied—but it is the first to use perception to consider how designs might support visual aggregation as datasets scale. The intention of this section is not to identify the optimal encodings to support different

visual aggregation tasks. While the discussion here might help guide such an exploration in the future, it is beyond the scope of this work. Instead, I explore where existing approaches break down and why color might provide a promising alternative for overcoming these limitations.

3.2 Considering Perception for Visual Aggregation

The study of human perception has had wide ranging impact on the design of effective visualizations (see Chapter 2 for examples). In particular, perception helps designers understand what the visual system can and cannot do with different kinds of visual information. The perception literature helps us understand limitations in visual processing, design cases that avoid them, or work most efficiently within them.

Here, I focus on ideas from perception that are directly relevant to creating scalable one-dimensional data visualizations that facilitate visual aggregation tasks. I have organized these ideas into four categories of visual processing relevant to data visualization. Each of the following sections identifies a category of visual processing, considers how relevant findings may explain limitations in the model designs in sequence comparison (connected alignment) and time series analysis (juxtaposed line graphs) discussed in the previous section, and explains how color may better facilitate processing in each category at scale. These explanations are supplemented by an example image showing a connected alignment visualization of small set of genomes (generally two) and colorfield visualization that encodes a similar structure over a larger number (seven to one hundred) of genomes.

3.2.1 Pop-Out

The visual system processes much of the information in a visualization before a viewer actively searches a display. The visual processing that occurs at first glance allows a viewer to rapidly identify targets in cluttered environments and also ascertain the gist—roughly the summary visual structure—of a scene to help guide active visual search. The visual system is able to process this preattentive visual information extremely efficiently [Wolfe, 2001].

These processes allow certain visual features to “pop-out” of a display. Pop-out effects have been explored in detail by the perception community (see [Healey and

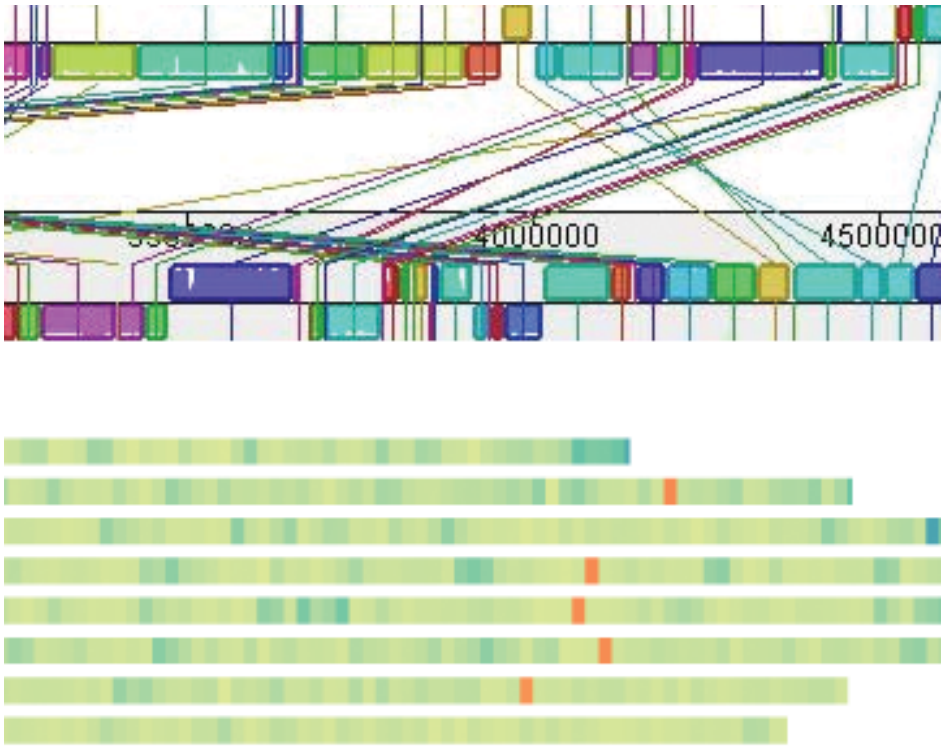


Figure 3.1: Pop-out processing helps more readily distinguish highly conserved regions when they are mapped to bright colors than a series of orthology lines.

Enns, 2012a] for a summary), and are often used in the design of visualizations [Ware, 2008]. Pop-out is of great utility for visualization as it can be leveraged to direct viewers' attention to important information (i.e. the viewer will not necessarily have to actively search to find important values).

Visualization designers can take advantage of this process by mapping important information to salient visual features. The visual system collects structural information to form a "spatial envelope" that encodes the most salient aspects of a display. High contrast and irregular regions are generally thought to be salient, implying that these regions are more likely to be the targets of attention and more likely to pop-out [Alvarez and Oliva, 2006]. It then becomes far easier to find desired objects in the scene, reducing visual search times and cognitive load [Wolfe, 2001], especially as the amount of data visualized increases. Creating and leveraging pop out in data visualization has been extensively explored, and is often used for tasks such as highlighting important data values. However, as the size of the data increases, effectively manipulating preattention can make it easier to locate important information, such as outliers.

There are many limits to the kinds of visual queries that can pop-out preatten-

tively. For example, the ability to distinguish between conjunctions of properties or find multiple targets are limited, but these features stand out more as the contrast between the target datapoints and remaining data increases [Duncan and Humphreys, 1989]. Further, increasing the saliency of certain data values increases the likelihood that specific data values will be identified preattentively [Nothdurft, 1993], but can also lead to search asymmetries: finding salient targets becomes efficient, but finding nonsalient values becomes more difficult (see Wolfe [Wolfe, 2001]).

Application to One-Dimensional Visualization

Pop-out can help analysts more readily find interesting elements, structures, or sequences in one-dimensional visualization. In connected sequence alignment visualizations, there are many possible ways pop-out might be able to facilitate visual aggregation by helping identify how large sections of material moves about in different sequences. For example, certain kinds of line crossings can pop-out and correspond to interesting features in the data, such as large regions where the order of genes has been reversed ([Enns, 1986, Fiorentini, 1989], Fig. 3.3). Unique orientations of the connecting lines can also pop-out to reveal locations where a motif (repeated pattern of genes) has been disrupted [Fiorentini, 1989, Haroz and Whitney, 2012]. However, this pop-out effect arises serendipitously—orientations and line crossings are determined by the data rather than the designer—and does not scale well. Pop-out in connected alignment visualization is limited by the number of elements compared—it can easily be canceled out by other connections crossing the same space as the preattentive connections. These residual crossings may clutter the salient connection or junction, inhibiting pop-out (see Section 3.2.4).

In time series analysis, high peaks or low valleys in line graphs may be salient due to their relative position and size. These features can aid visual aggregation tasks that involve identifying extreme values within the data. However, this effect likely does not scale well to larger numbers of sequences—as more sequences are compared, the space dedicated to a graph will likely shrink accordingly. As a result, the differences in position and size decrease in magnitude, making the pop-out less salient. Increased numbers of sequences may also introduce variability in defining these “peaks and valleys,” potentially reducing their ability to pop out despite salient local contrasts.

With color, salience can be achieved relatively robustly. Color mappings can

be selected to make certain groups of genes or values of a certain magnitude preattentively stand out. While, as in the line graph case, size can reduce the contrast of a color value, color facilitates a wider degree of control over salient differences—a designer can simply design a color ramp with a larger dynamic range. Current systems often take advantage of preattentive pop out via highlighting and matching color maps to data distributions [Tominski et al., 2008].

Color pop-out can also be leveraged to detect large scale pattern changes. Since attention capitalizes on spatial regularities in the data, irregular patterns in spatial organization can attract initial attention toward these regions of the visualization [Fiorentini, 1989, Haroz and Whitney, 2012]. While these spatial effects can be manipulated to highlight certain relationships within the data, including regions where data patterns are inverted (e.g. a gradient reversal) or where continuous patterns are interrupted (e.g. a gradient disruption).

However, the use of controlled schemes must be done with caution. Mauve [Darling et al., 2004] employs a color scheme that causes pre-attentive association of unrelated regions: color can reinforce the orthology shown by connectivity, but does not encode it completely. Unintended saliency can slow visual search for other features [Nothdurft, 1993], such as tracing edge crossings. This can be overcome by choosing color mappings that map semantically related content to similar colors.

Further, a designer does not often know a priori what data is significant to the viewer. Making on piece of information salient can introduce search asymmetries—search for salient data values becomes easier, while finding non-relevant data items becomes more difficult. However, with color, these asymmetries can often be overcome by allowing the analyst to specify the information they are interested in, whereas with connection and position, preattentive features generally arise as a result of the data, not of the visual mapping.

3.2.2 Summarization

Early visual processes provide additional value for visualization beyond pop-out. For example, the visual system can create rapid statistical summaries of a scene at first glance. Prior to attention actively engaging, the visual system creates a coarse representation, or gist, that identifies regions of interest that attention will be directed to in the search. The gist provides a set of prior probabilities that guide navigation when viewers actively explore a scene [Oliva, 2005]. In this process, the

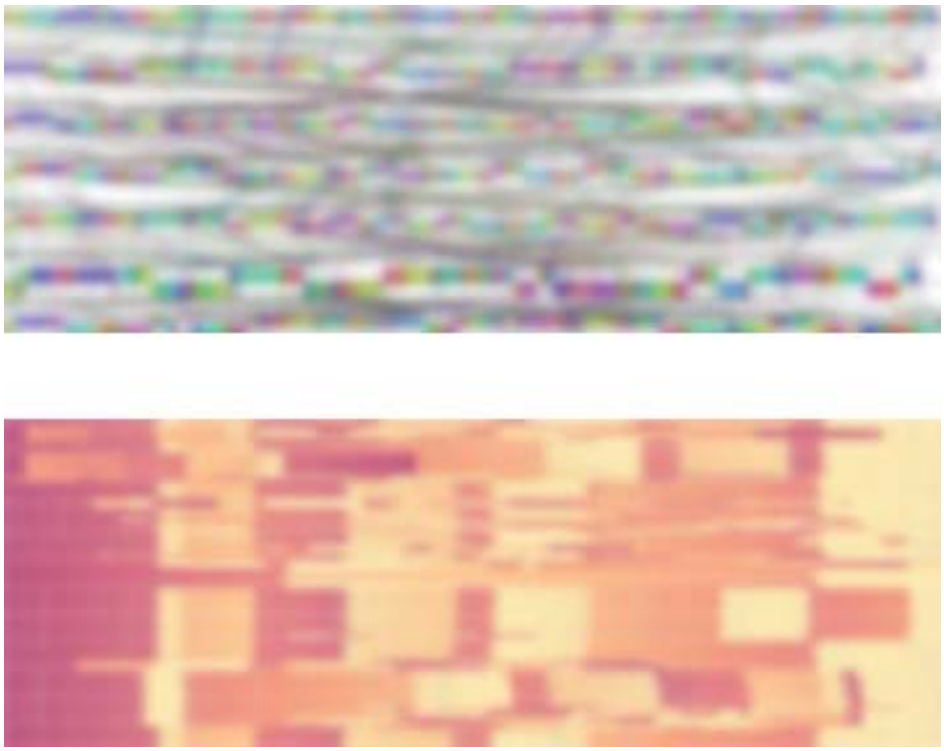


Figure 3.2: Visualizations that support the low-resolution processing of visualized data can orient the viewer as to the overall data trends without requiring their explicit attention. For example, large scale patterns in data are more meaningfully averaged when data is encoded using color (top) instead of connection (bottom).

visual system is able to aggregate collections of data together [Balas et al., 2009]. These collections are internally represented as a set of *ensemble statistics*, such as the mean and variance, that describe summary properties of the collection [Ariely, 2001b]. These statistical summaries are collected both before attention is engaged (e.g. during gist processing [Oliva and Torralba, 2006]) and also in regions outside of the focus of attention (e.g. in the periphery [Freeman and Simoncelli, 2011]).

If a display is designed correctly, these ensemble statistics allow the viewer to access low-resolution summary information even without having to explicitly attend to all regions of the display. Summarization, if supported, estimates many aggregate statistical properties of a dataset, such as the mean [Ariely, 2001b] or variance [Morgan et al., 2008] of a group of visual features. These statistics can be used to make sense of large collections of information. For visualization, summarization processes can estimate coarse information about a visual encoding and likely helps in many visual aggregation tasks, such as computing the relative layout of a collection of points [Gleicher et al., 2013b], and some basic statistical properties such as mean [Correll et al., 2012b], variance,

and correlation [Harrison et al., 2014, Rensink and Baldridge, 2010]. This is valuable as often a scientist needs the context, not the details, of objects outside their immediate focus. Summarization facilitates visual aggregation by grouping related collections of items together into a series of summary values.

Applications to One-Dimensional Visualization

To effectively leverage summarization in visualization, displays must be designed so that their low-resolution summaries are meaningful. For example, summarization can be crudely approximated by blurring a dataset to generate rough local averages. When blurred (which is similar to what summarization effectively does), a connected alignment visualization becomes a gray mass, while colorfields retain useful features, such as large color gradients (see Figure 3.2). However, designs leveraging color average to meaningful structures.

Many aspects of connected alignment visualizations may be detrimental to visual summarization. At a low level, the visual system is adept at generating summary information about orientation (e.g. [Choo et al., 2012b, Morgan et al., 2008]). However, in connected displays, oriented lines overlap with one another, which may complicate this averaging. The large numbers of small lines connecting matches in synteny and parallel alignment views simply vanish or merge when considered at low resolution. Edge bundling might help, but other, more summarization-friendly designs might be needed.

Further, with connected displays, connections are often drawn only between adjacent sequences. This means that any summary judgments that can be inferred from the orientation of the different connecting lines (e.g. estimating how genes move around on average or how many genes are conserved between sequences) are entirely dependent on the ordering of the sequences in the visualization. Inferences comparing sequences that are not colocated require the viewer to effectively "daisy-chain" the results of a visual aggregation task across all of the sequences between the compared sequences.

For time series, line graphs might also challenge summarization mechanisms. Line graphs rely on different visual features to encode information: individual points are connected to form shapes and changing in the height of the shape encode values. The visual system is capable of computing ensemble statistics over individual points efficiently (see [Alvarez and Oliva, 2009] for perceptual evidence and [Gleicher et al., 2013b] for evidence from the visualization community). However, the complex shape of line graphs are processed in higher-level visual

areas that may not be readily summarized across multiple graphs in a useful way. There is no evidence that the visual system can efficiently average across height per se. In fact, prior evidence suggests that global shape perception, necessary for summarizing line graphs, is perceptually inefficient [Wolfe and Bennett, 1997].

Line graphs may instead rely on the visual system's abilities to summarize size [Ariely, 2001a, Chong and Treisman, 2003] as a proxy for height. To make use of this ability, the visual system must segment out relevant visual features from the global shape of the graph. However, this segmentation operates over rigid constraints, and the observer may not be able to arbitrarily summarize data in a line graph [Franconeri et al., 2009, Singh and Hoffman, 1997]. Chapter 4 evaluates how well line graphs support tasks involving summarization.

The ability of the visual system to readily summarize color [Bauer, 2010] makes it a promising encoding for supporting visual aggregation tasks. The visual system's ability to summarize dense fields of color into meaningful aggregates is something that artists (e.g. pointilism [Kleiner, 2013]) and display designers (e.g. pixels in a Bayer mosaic) have leveraged for decades and longer. More recent evidence suggests that some aspects of color can be meaningfully summarized for colors of larger areas [Bauer, 2010] and that these summaries can provide useful structural information in early visual processing [Graham et al., 2009].

While color can be readily summarized by the visual system, some care must be taken to compose these summaries correctly. For example, the visual system summarizes this data in across both the vertical and horizontal dimensions [Graham et al., 2009]. The rows and columns in a traditional heatmap may be averaged together. In one-dimensional data comparisons, each data sequence is a unique entity, but summaries may lump together data across sequences in a heatmap. Separating different sequences into independent tracks using empty space can significantly alleviate this issue [Balas et al., 2009].

Perceived similarity of different areas of a display might be a function of local averaging [Graham et al., 2009]. If local averages are meaningful, such as for colors whose perceptual differences align with value differences, then ensemble statistics can help viewers compare these clusters. If the averages are not meaningful, for example, if color differences do not align with value difference, local averages may not provide a useful comparison.

Additionally, color mappings should take advantage of the visual system's abilities to summarize color. As summaries effectively statistically aggregate colocated visual features, colors that are perceptually close should encode similar

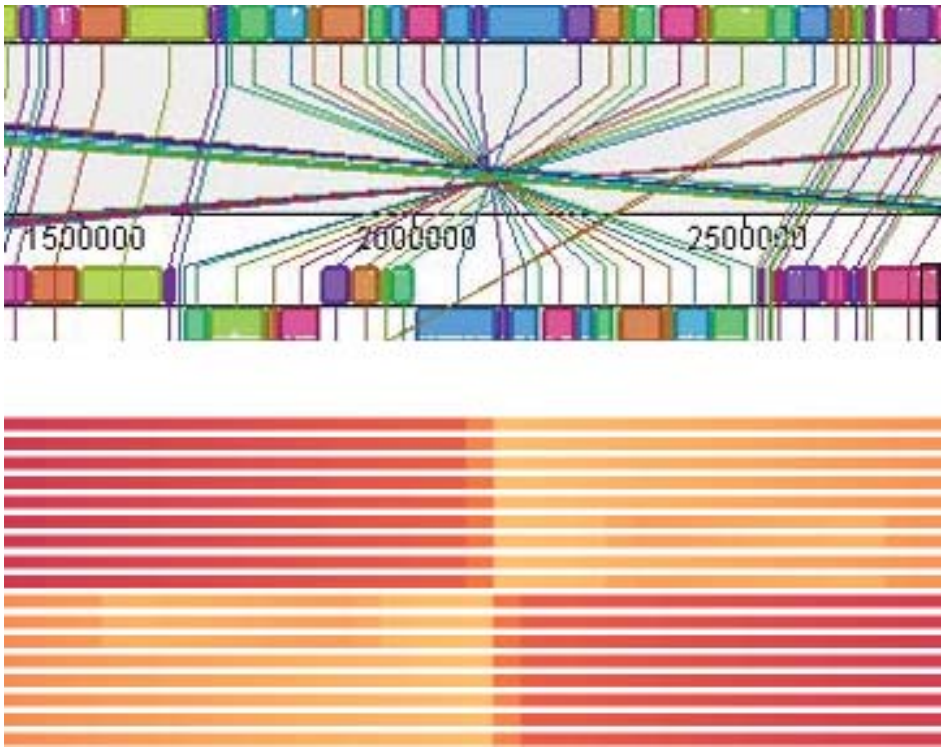


Figure 3.3: Connections between related data impose a non-linear search order to the data, whereas a conventional reading order supports a more natural search pattern and allow large component color fields to be associated preattentively.

values to effectively leverage summarization for large collections of data. This constraint supports pre-attentive pattern finding and summarization: large fields of colors can be matched and texture patterns suggest relevant visual relationships (see Figure 3.1). For example, reversal of color gradients when color difference is proportional to value difference readily communicates inversions. Summaries of categorical colors, where color difference does not map to data, may help identify the relative numerosity of various values [Healey et al., 1996], but the average of these colors is meaningless.

3.2.3 Visual Search

When a viewer is looking for a specific datapoint (or set of data points) that do not readily pop-out, they must instead search for them. Visual search occurs when a viewer must scan their attention over the scene to search for targets. Without perceptual aid, search tasks can be cognitively demanding and time-consuming [Alvarez et al., 2007]. By designing visualization systems that cooperate with perceptual search mechanisms, viewers can more easily process the display for

more rapid and efficient visual search.

Search is initially guided by pop-out and structural summarization—these processes identify areas of potential interest and attract attention prior to search. Visual search is key in constructing effective visualizations as supporting visual search makes it easier for the viewer to actively find interesting information. If the search target is known a priori and easy to formula, interactive searches can be used. However, in visual aggregation tasks, the search target is defined relative to other values in the collection, such as identifying co-located clusters of values with high averages or searching for the local maxima of a subset of data values.

Effectively leveraging summary processes can facilitate visual aggregation tasks involving search by establishing a low-resolution contextual map of visual encodings. This ensures that the visual system can readily identify objects of importance during early visual exploration. For example, large discontinuities can pop-out, helping the viewer determine where to direct their attention (Fig. 3.3). When the structure of a visualization promotes familiar search strategies, such as preserving a left-to-right reading order [Arnheim, 1976] or using a regular layout [Alvarez et al., 2007], viewers are generally more comfortable and more efficient at finding information in a visualization.

Applications to One-Dimensional Data Visualization

Many interaction techniques, such as text-based search and highlighting, can facilitate visual search when a target is known and well-defined. However, in visual aggregation tasks, a search target might be a single value defined based on its relationship to other points in a collection, such as the median value of a collection, or might be a collection of values, such as a spatial cluster with a high or low average value. Visualization can support effective visual search for these tasks by using regular layouts, visual features that are readily summarized, and methods that colocate meaningful data.

In connected alignment visualization, as mentioned previously, deriving useful summaries is difficult (see §3.2.2 for details). However, connected lines also impose a non-linear reading order to trace the movement of genes throughout a dataset. This may significantly inhibit search tasks by creating a secondary search structure that does not follow a traditional reading order even for salient connected patterns, such as the regular crossing pattern in Figure 3.3.

Line graphs can preserve a single left-right reading configuration, allowing the viewer to methodically scan over the data in a logical ordering, thereby reducing

the cognitive cost of visual search. However, the presentation of data in a line graph is fairly rigid—the order in which the data is arranged is firmly coupled to the data axis that is being visualized. The visualization is limited in its ability to reorganize data along the x-axis to create new clusters, impairing the types of visual queries a viewer is able to make by constraining the proximity between data points.

Color mappings are not necessarily bound by positional constraints (other than that they must have a relatively unique position). While finding specific values mapped to color may be less efficient for low-level search tasks [Cleveland and McGill, 1984] (a limitation addressed in the second part of this dissertation), the efficiency with which color can be summarized coupled with flexible positions are extremely beneficial for search tasks involving visual aggregation.

To efficiently leverage color for visual search in one-dimensional data applications, color tracks should follow a conventional left-right or top-bottom reading order. Dividing data sequences using white space, as discussed in §3.2.2, can reinforce this order.

To help form new clusters that facilitate more flexible aggregate search tasks, the ordering of colors within a data sequence should be somewhat flexible ([Slingsby et al., 2009]). While this flexibility allows for complex visual queries, it also can potentially inhibit search by placing data in an unconventional order. Using coordinated views or animated transitions for rearrangement will reduce the number of new searches associated with changes in viewpoint or display. Flexible orderings to the data can also benefit from the pairing of conventional and novel views (e.g. the approaches described in [Meyer et al., 2009, Peeters et al., 2004]) and facilitate comparison by colocating values of interest ([Wickens and Carswell, 1995]). Since analysts know how to navigate in conventional models, they can more quickly identify points of interest, which can then be used to map to unfamiliar views. This reduces the cognitive load of search when data is presented in novel clusters. By providing visual coordination mechanisms, exposure to existing tool can help simplify search in novel viewpoints.

3.2.4 Visual Clutter

Visual clutter occurs when item quantity, encoding, or layout hinders performance in search tasks [Rosenholtz et al., 2005]. Clutter impairs the perceptual system by bogging down cognitive processes and slowing visual search. In data-processing

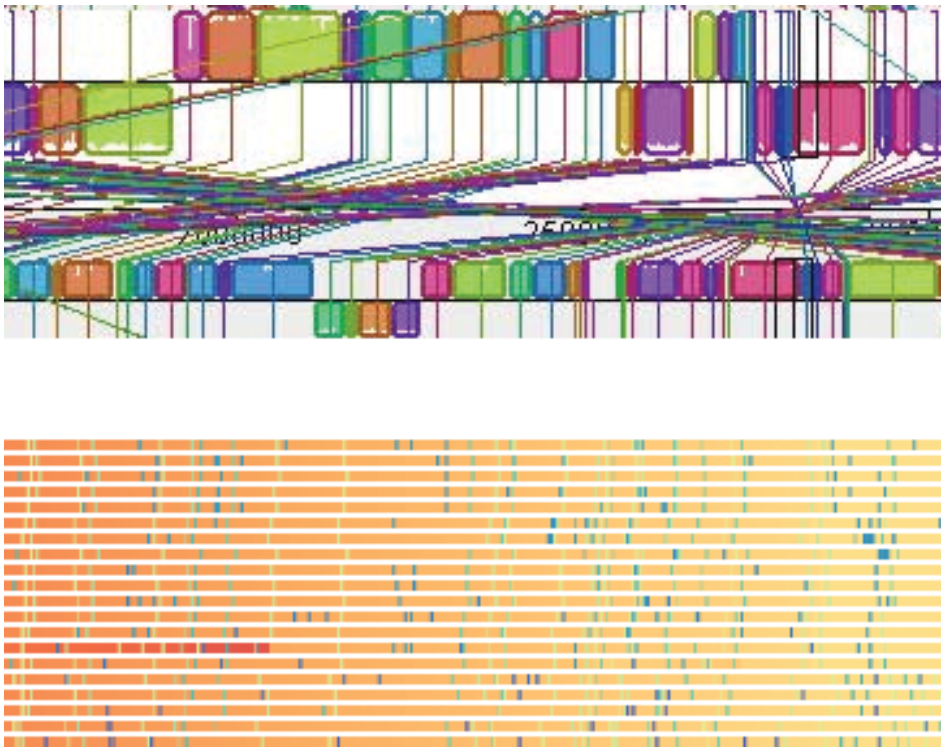


Figure 3.4: Visual clutter can significantly inhibit perceptual processing by adding additional visual objects to a scene. Orthology lines quickly become cluttered, with multiple lines crossing in unstructured ways. Clutter in color instead forms a dense texture in regions with high numbers of sequence events.

tasks like sequence comparison, clutter reduction by adjusting semantic data granularity (e.g. visualizing the data for weeks instead of days) often proves far more effective than simply removing data and still preserves the overall dataset [Rosenholtz et al., 2007].

In addition to slowing visual search, clutter can impede how accurately viewers conduct different visualization tasks. Clutter can lead to errors in estimating visualized values and also increase the confidence with which values are estimated [Baldassi et al., 2006]. This misestimation may lead to problems when visually aggregating information across a cluttered display.

Application to One-Dimensional Visualization

Clutter is a common concern in visualization, so much so that models of clutter in visualization have been developed [Lohrenz et al., 2009]. As the amount of data a design must visualize increases, clutter becomes increasingly important. More data generally correlates with a larger number of marks. How the data is

encoded likely plays a significant role in how quickly visual clutter accumulates. Connected alignment visualizations and juxtaposed lines graphs can become cluttered in different ways. However, with color, the visual system's abilities to readily summarize color may be beneficial for managing clutter.

Clutter is a substantial problem for connected visualizations. As the number of connections between adjacent genomes increases, the amount of visual clutter is likely to increase. Greater numbers of connections quickly form tangled webs that make connections difficult to trace (Fig. 3.4). While techniques like edge-bundling can help reduce clutter [Meyer et al., 2009], they substantially reduce the fidelity of represented data and their effectiveness is heavily dependent on the regularity of the connections—bundles generally preserve connections of similar orientation.

Alternatively, juxtaposed line graphs do not necessarily suffer from the same clutter issues as connected visualizations. However, line graphs become increasingly cluttered as the amount and variability of data in a single sequence increases—the line converges to, in essence, a dense scribble of data as points are pushed closer to one another. Low-pass filters and other kernel-based processing can smooth the data to reduce such issues. These reductions operate in data-space, making it difficult to control the amount of visual simplification introduced for a given visualization.

Color encodings, like juxtaposed line graphs, also benefit from a lack of the overlapping encodings which cause clutter in connected displays. However, in color, the visual system's ability to readily aggregate information can actually leverage color to form data "textures" that are readily parsed. Large amounts of data clutter form textures that suggest regions with potentially interesting variations.

However, as in with line graphs, representing one dimensional data using color is still problematic once the density of the dataset exceeds a certain level (e.g. once there is more data than pixels to visualize that data). The flexibility of color encodings makes it possible to overcome this limitation—we can manipulate the position and representation of encoded data values in order to better fit the visualized data on the screen. In the next section, I will discuss a technique that addresses this limitation for one-dimensional data visualizations using color. This technique builds upon these four processes discussed in this section (pop-out, summarization, visual search, and visual clutter) to facilitate visual aggregation tasks at scale.

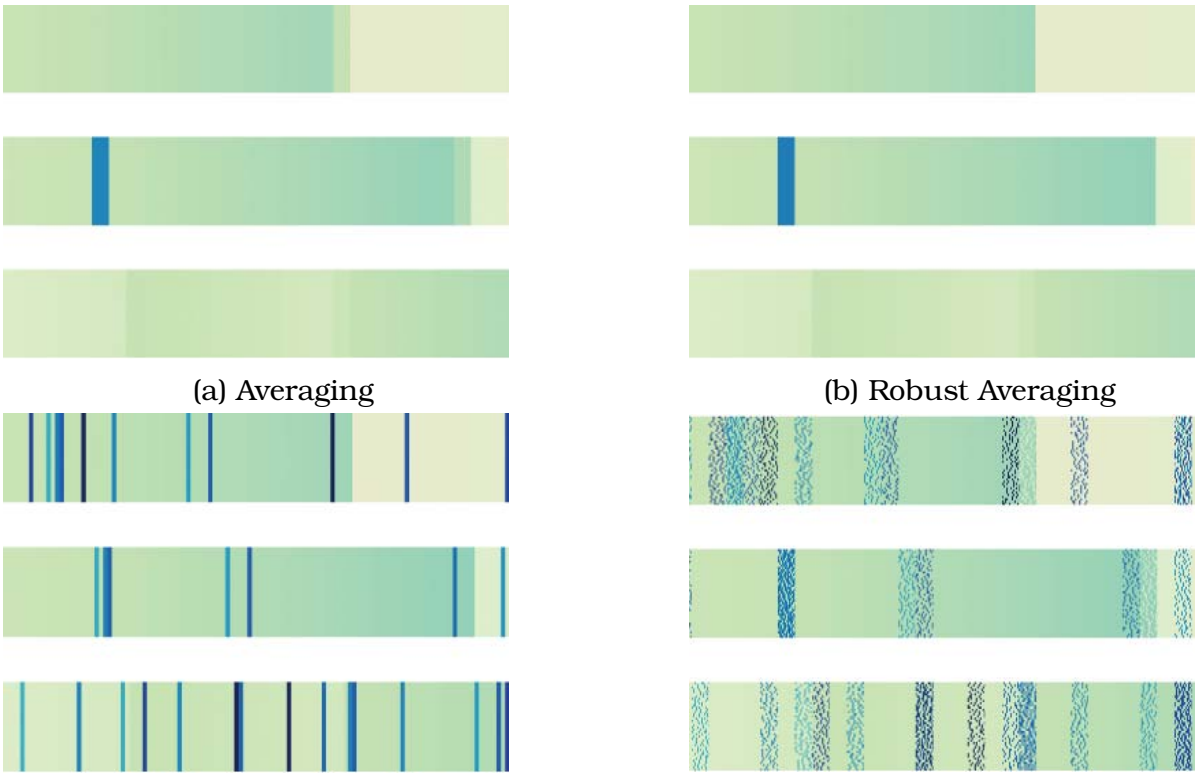
3.3 Designing Aggregate Visual Encodings

The amount of available data continues to grow, creating demands for visualization tools that can scale to larger datasets and the challenges they bring. Constructing such tools will require facing a number of challenges, including the engineering issues of handling immense amounts of data. The visual designs used by such tools must also be carefully planned to remain effective as the data grows more complex.

The previous section discusses how perception can inform methods for representing data that better facilitate visual analysis at scale, especially when those analyses must consider multiple values simultaneously. This discussion suggests that color is extremely useful for facilitating data analysis at scale. However, many datasets in one-dimensional analysis have sequences or series that are longer the horizontal spans of pixels on a conventional display. Even if we could fit all of the data on the display, maximally dense displays may be cluttered and difficult to interpret, and the visual system can only process a limited amount of information at any given time. Using color alleviates some of this complexity, but is still limited by the number of pixels on the display.

To manage this complexity for one-dimensional data visualization, visualizations must explicitly aggregate in the horizontal direction. Several approaches to overcome screen space restrictions involve computationally reducing the dataset, as in [Dupont and Plummer, 2003, Keim et al., 2007, Lampe and Hauser, 2011, Papadimitriou et al., 2013]. However, these methods require knowing and quantifying information relevant to the task *a priori*. These techniques do not offer much control over what specific information is thrown away in the course of the aggregation—they operate over the data abstractly to facilitate data reduction, not *semantically* to support a given set of analysis tasks. Alternatively, aggregation approaches that reduce data based on semantic structures, such as cliques in graphs [Dunne and Shneiderman, 2013] and different levels of hierarchy [Elmqvist and Fekete, 2010], abstract structural information from the dataset into compact glyphs. Time series visualizations often use this approach to aggregate information over meaningful time spans ([Lammarsch et al., 2009]). Visualization techniques from biology allow the biologists to collapse large regions within the sequence, encoded as the dominant value within that region ([Slack et al., 2004, Vehlow et al., 2011]).

Data in these techniques is aggregated over a predefined region and condensed



(a) Averaging (b) Robust Averaging

(c) Event Striping (d) Color Weaving

Figure 3.5: The different aggregation schemes available in Sequence Surveyor. (a) Averaging reveals high-level trends in the blocks. (b) Robust averaging removes the influence of outliers from the average, resulting in smoother color fields conveying the dominant trends in the data. (c) Event Striping highlights outliers in the data. (d) Color weaving depicts the distribution of genes in the blocks.

into a single, predefined value. As a result, data that may support other analysis tasks is abstracted away. Here, I propose a screen space method using color to aggregate one-dimensional data based on task (Fig. 3.6). This method first maps the data in a sequence to visual space, creating a fixed mapping of pixels to data values. Then contiguous datapoints are grouped across fixed pixel windows, creating a series of screen-space data "blocks" of roughly uniform size. Data within these blocks is then reduced locally based on the task specified by the analyst and mapped to a glyph that uses color to communicate the reduced data values. The glyphs have been designed to support the specified task based on the perceptual processes discussed in the previous section.

We provide four aggregate glyphs that encode blocked data, shown in Figure 3.5: averaging, robust averaging, event striping, and color weaving. By providing different aggregation filters, different properties of the data can be explored at the

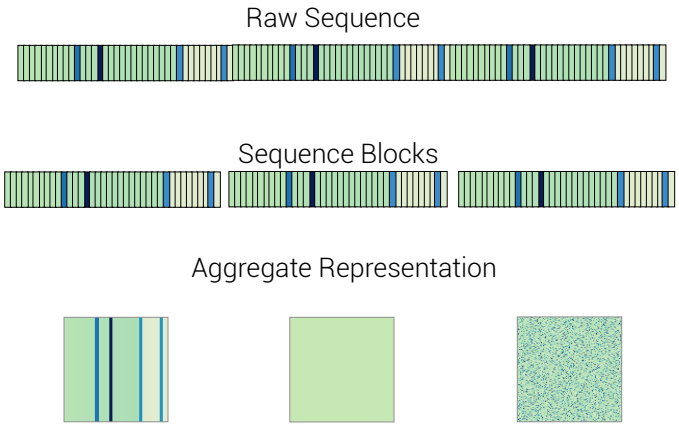


Figure 3.6: Blocking first maps data to visual space, and divides the corresponding data into uniform, screen-space bins. Once these bins are composed, the data within each bin is reduced locally and encoded using a perceptually-inspired glyph designed to support a specific type of visualization task.

overview level without having to recompute the display properties of the entire set. Drilling deeper into these aggregated blocks can be accomplished with zooming (§5.1.6). Each of these glyphs is defined as follows:

- *A*veraging (Figure 3.5a) colors blocks by the mean of the component gene color values. This glyph summarizes overall trends in each block, as seen in Figure 5.11b, using a single color. This encoding minimizes visual clutter by greatly reducing the overall visual complexity of the dataset, allowing the viewer to readily infer summary information about overall dataset and helping areas of unusual data values pop-out.

- *R*obust averaging (Figure 3.5b) provides a statistically more robust visualization of the data. This technique averages data within a block more intelligently by removing the visual contribution of outliers. This encoding accomplishes many of the same perceptual goals as averaging, but does focuses on the most dominant patterns in the data.

  I compute the robust average by averaging the color values within one mean absolute deviation of the inner quartile range of the block. While I define outliers as values outside of a block's IQR in this application, any definition of outliers can be used. In practice, designers should choose whatever method best fits the data and domain.

- *E*vent striping (Figure 3.5c) flags outliers and changes to trends in a block as "events." Events are drawn as pixel-wide vertical stripes at the relative

location of each event within the block. This prioritizes outliers within the data by physically enlarging these regions to highlight their existence, which may otherwise be lost to more dominant trends like in Figure 5.10b. Event striping helps to make outliers more salient and more likely to pop-out.

When there are more values in a sequence than pixels to represent those values, the space between subsequent events encodes the average of values between the events. This both contextualizes the outliers in the active gradient and helps support aggregate summarization tasks. If there are enough pixels to represent all values in a sequence, every value in the block is drawn as a single stripe. This allows viewers access to all of the data in a sequence.

- *C*olor weaving (Figure 3.5d) randomly permutes data and maps the resulting information at the pixel level, similar to the overlay technique presented by [Hagh-Shenas et al., 2007]. The resulting glyph is capable of displaying as many data values as the pixel area of the block (as opposed to the pixel width of the block in more conventional approaches). Randomization helps avoid misleading striping artifacts that may be introduced through repeating ordered data for blocks where the number of data values is less than the pixel area of the block.

  This approach takes advantage of the visual system's abilities to summarize data values to support multiscale inference into the distribution of values within a block. In many cases, all of the information within the block is communicated, but breaking local structures may facilitate visual averaging. Actively attending to a block can then act as a form of zoom—the visual system can extract ensemble statistical values from unattended blocks, while focusing on data within the block can reveal more precise, pixel-level information.

Screen-space grouping allows a great deal of flexibility in how data is ordered within a sequence. Aggregation is independent of the horizontal data dimension and also provides a fine degree of control over how finely the aggregation is applied. The width of a block can be determined in two ways: by a user-defined parameter and by gaps in data sequence. Allowing the user to define the maximum block width within a sequence in pixels is similar to specifying a "bin size" parameter in a histogram. This provides the analyst with fine-grained control over the visual complexity of a display—smaller blocks mean less data reduction.

However, these windows only operate over contiguous data. In some cases, data at a certain position in screen space might simply be missing. For example, a data sequence might be ordered such that column positions map to values, like when comparing genomes to a reference. A gap can arise due to an absence of that value in a sequence. Alternatively, the data might be incomplete and no measure exists for a particular position or time point. Naturally-occurring gaps in the data positions that are at least one pixel wide prematurely break a block grouping, creating a visible gap in the encoding. This induced irregularity allows the viewer to see significant gaps in the data at an aggregate level while not overemphasizing small gaps which would otherwise be perceptually indistinguishable in an unaggregated overview. This gapping helps support data clustering by treating physically separate clusters as independent blocks, preserving patterns local to each block.

Further, by aggregating information locally, the process preserves interesting local variations—all reduction criteria is done agnostic of the global data distribution. As a result, data is tuned to the local neighborhood of the data, preserving interesting pockets of variation. This also supports more semantically complex reductions, such as the ability to define an outlier with respect to the local sequence neighborhood. This helps to identify and highlight variations that break local gradients, but may otherwise conform to the global data distribution.

3.4 Discussion

This chapter presents a theoretical basis for using color to support visual aggregation tasks using model problems from one-dimensional data analysis. In this chapter, I have introduced an organization for leveraging perception to reason about scalability and visual aggregation, outlined a perceptual argument for color as an effective encoding for visual aggregation tasks, and presented a visual method for task-driven aggregation of one-dimensional data. These contributions collectively illustrate how color can be used to design visualizations that are both scalable and readily support analyses that combine information from multiple datapoints simultaneously.

While I use these contributions to motivate color as an effective channel for visual aggregation, this discussion represents a starting point for considering the perceptual basis for visual aggregation tasks. The survey of perceptual processes is by no means exhaustive, but instead intended to represent processes that

are critical to aggregate analyses of scalable visualization, at least one of which (summarization) had not been explicitly considered by visualization designers prior to this work.

Additionally, the arguments presented here are based on a mapping of theories from perception which do not consider many of the nuances of data visualization tasks. Experimental validation is critical for verifying how well these inferences translate to visualization. Some of this validation will occur in the next chapter.

The aggregation methods presented in §3.3 do not necessarily support all possible visual aggregation tasks. I instead see these glyphs as representative of an important class of statistical tasks. A broader consideration of visual aggregation tasks as well as their links to psychology are outlined in on-going work.

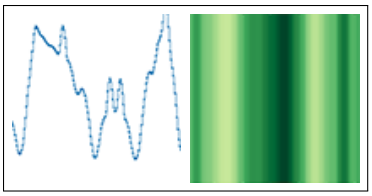
4 TASK-DRIVEN AGGREGATION FOR SEQUENCE DATA

The previous chapter introduced a method for task-driven aggregation of one-dimensional data. This method and the accompanying task-driven glyph designs (averaging/robust averaging, color weaving, and event striping) are inspired by findings from perceptual psychology. While perception explains how the visual system might process encoded data, these explanations are derived to understand the mechanisms of the visual system. The results generally describe how the visual system processes simple stimuli over short durations and under heavily controlled conditions. These experiments are not reflective of visualizations, which often have large amounts of complex data explored over arbitrary lengths of time and under a plethora of conditions.

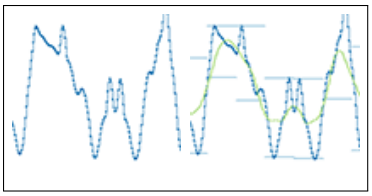
In this chapter, I validate the design considerations and method presented in Section 3.3 for visualization. This study considers how color and other properties of visualization design help viewers perform different statistical visual aggregation tasks. I combine prior results from perceptual science and graphical perception to suggest a set of design variables that influence performance on various aggregate comparison tasks and describe how choices in these variables can lead to designs that are matched to particular tasks. I focus on statistical tasks for exploring time series data as it is more familiar for the general viewer than genomics analyses. Aggregate designs are blocked semantically (at the level of months). I use these variables to assess a set of eight different designs, predicting how they will support a set of six aggregate time series comparison tasks. A crowdsourced evaluation confirms these predictions. These results not only provide evidence for how the studied visualizations support different tasks, but also provide design guidance into how the discussed design variables can guide new visualizations well suited to various tasks.

4.1 Overview

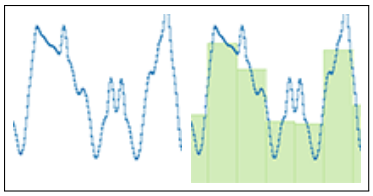
Visualizations can support judgments over collections using two strategies: they may present raw data requiring the viewer to determine the aggregate properties, or they may compute these aggregate properties and present the derived data. For example, if the designer knows that the viewer is trying to find the maximal value in a series, they may either explicitly compute and encode the maximum, or choose a design that facilitates visual search for the maximum. While such computational



(a) **Visual variables** can influence task performance - positional encodings (left) help viewers make point comparisons, but color encodings (right) help viewers make summary comparisons over regions of a series.



(b) **Mapping variables** can influence task performance - overlaying statistical quantities explicitly on the original series (right) is beneficial for tasks where extracting the quantities visually would be difficult without assistance.



(c) **Computational variables** can influence task performance - dividing a series into discrete, task-relevant blocks (right) is beneficial for aggregate summarization tasks.

Figure 4.1: We can infer how well a particular encoding support a given task by examining the interplay of visual variables (what visual channels are used to encode value), mapping variables (which raw or derived quantities are visualized), and computational variables (how these quantities are computed).

aggregation can be precise, it also requires a number of specific preconditions to be met. For example, it requires knowing *a priori* which properties are relevant to the task. In contrast, visual aggregation relies on the capabilities of the viewer's visual system, necessitating visual encodings that allow for relevant properties to be determined effectively. Computational and visual aggregation are compared more thoroughly in Chapter 2. Both strategies require a good match between design and task. However, aside from specific examples of designs that apply to specific tasks, there has been little exploration of the trade-offs in how various design elements may apply to different visual aggregation tasks. By understanding how aggregation strategies combine with other design elements, we can better guide the design and selection of visualizations to support aggregate comparison tasks.

The previous chapter outlines a number of perceptual theories that support speculation about designing visualization to support visual aggregation, but this speculation is not grounded in visualization, but rather in a more abstract understanding of the visual system. In this chapter, I identify three key variables in the design of visual displays, and explore their effect on viewers' ability to carry out various statistical aggregation tasks. **Visual variables** [Bertin, 1983] refer to the visual channels used to represent the data values, such as color, position, or orientation. **Mapping variables** refer to the selection of which particular properties

of the data to display, for instance choosing to emphasize local outliers or creating a derived dimension from existing data. **Computational variables** describe the methods used to compress the data, such as whether a statistic is computed continuously or segmented over discrete regions of the series. Since no one choice of encoding will be appropriate for all tasks, and the tasks to be completed may not be known *a priori*, understanding the relationship between these three design variables and different types of aggregate comparisons provides guidance into the design of effective visualizations for broad sets of aggregation tasks.

As in the previous chapter, I base this exploration in the model domain of time series analysis. I will present an evaluation (conducted with collaborators in computer science and psychology) that evaluates how each of these variables influences performance on six different statistical tasks—three *point comparisons* requiring a comparison of individual points within different subsets of the data and three *summary comparisons* that require viewers to combine data within each subset. My results show that all three variables offer robust predictions about performance for these tasks. Figure 4.1 shows how consideration of these variables might lead to different design choices for different tasks.

4.2 Informing Design through Task

These experiments focus on performance for a special form of visual aggregation task, aggregate comparison tasks, which require comparisons between ranges of points. This experiment measures performance for two specific classes of aggregate comparison task: point comparisons and summary comparisons. *Point comparisons* require viewers to identify and compare points drawn from specific subsets of the data, such as monthly ranges, whereas *summary comparisons* compare values computed from entire ranges of the data, such as monthly averages.

I analyze performance on these tasks as a function of three design variables that, based on the discussion in Section 3, I believe offer predictive insight for matching task and encoding: visual variables, mapping variables, and computational variables. These variables arise from the types of choices a designer must consider when creating a visual encoding meant to deal with information in the aggregate. While these variables do not attempt to define the full breadth of encoding choices made by a designer, I believe that these design variables help characterize the tasks an encoding supports and, by understanding the relationship between variable and task, visualizations can be tailored to better

support different visual aggregation tasks.

Visual variables refer to the choices in low-level visual properties used to represent data, such as position and color [Bertin, 1983]. While graphical perception results suggest what encodings may provide the most precise extraction [Cleveland and McGill, 1984], results on visual aggregation suggest that different visual variables may be better for statistical summarization [Correll et al., 2012a]. Color is particularly promising for summary comparisons (see Section 3.2.2 for a justification), whereas trade-offs in position are promising for point-value comparisons (see Section 3.2.3 for a justification).

Mapping variables refer to *which* aggregate properties are computed and presented. For example, a visualization may show the raw data, averages, or extrema. The use of such computed aggregates allows the visualization to do work that would otherwise need to be done by the viewer, and can offer a degree of precision that cannot be achieved mentally. However, these computed statistics are task specific: the system must know which statistics are relevant to the viewer's goals, and avoid overwhelming the viewer with too many irrelevant ones. Mapping variables are more nuanced than simply encoding the "right" answer for a given task, a statistic that is not directly relevant may still help the viewer by serving as a *benchmark* for a related task.

Computational variables refer to *how* these aggregate properties are computed. For example, a given statistic, such as mean, may be computed over discrete ranges of the data or as a continuous moving average. For example, in Section 3.3, choosing to explicitly encode local averages reduces visual clutter in complex data while still supporting aggregate inference in to the data. Some of these choices allow the computation to fit the task, for example by blocking in groups relevant to the task, but this requires foreknowledge of the task. Interaction is commonly used to adjust computational variables to support tasks at different scopes. Mapping variables provide a direct way to manage the visual clutter of a scene by controlling the granularity of information shown.

These design variables make explicit the choices in designing a visualization that will affect the visualization's applicability to specific tasks. They allow a designer some predictive insight into how a proposed design may fit a set of tasks. These variables conceptually align with the filtering and mapping stages of the visualization pipeline [Card and Mackinlay, 1997] used to characterize visualization designs. However, our approach differs as we seek to inform design using task by characterizing explicit design choices rather than to more generally characterize

visualization approaches. Further, the distinction between the granularity at which each encoded statistic is computed (computational variables) and how these statistics are encoded (visual variables) is important for constructing designs that support different varieties of aggregate tasks at different granularities within a series.

The next section outlines a range of existing designs (shown in Figure 4.2) for displaying times series data using these variables to predict each design's appropriateness for a range of tasks. This provides both a validation of the predictive power of these variables, as well as a better understanding of a set of known tasks and encodings. Further, this provides some empirical grounding for the claims made in Section 3.3—three of the proposed visualization designs are derived from the aggregate glyphs designed to support visual aggregation at scale.

4.3 Hypotheses and Examples

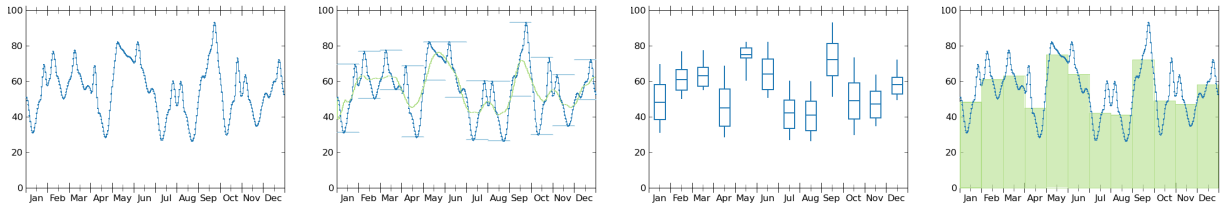
Considering how each design variable is processed visually may help predict how different encodings support different visual aggregate tasks. In particular, each design variable independently allows us to make predictions about the performance of different visual encodings for various tasks:

H1: Visual variables that support preattentive summarization, such as color, will better support summary comparisons for designs where aggregation is not done computationally, whereas visual variables with higher perceptual fidelity, such as position, will better support point comparisons.

H2: Mapping variables that explicitly convey relevant statistics (either the exact task statistic or a benchmark statistic, such as the mean when estimating variance) will support more accurate comparisons, but will still be limited by how each statistic is computed and visualized.

H3: Computational variables that provide task-aligned discrete aggregation will support more accurate aggregate comparisons than variables which are encoded continuously by reducing the visual clutter present in the design.

I confirm these predictions by evaluating viewers' abilities to accomplish six aggregate comparison tasks for eight encodings for time series data. For each task, I performed a between-subjects experiment to compare viewer accuracy for each encoding. The tasks, detailed in the Section 4.4 section, include three point comparison tasks (identifying the month with the largest value, smallest value, and largest range) and three summary comparison tasks (identifying the

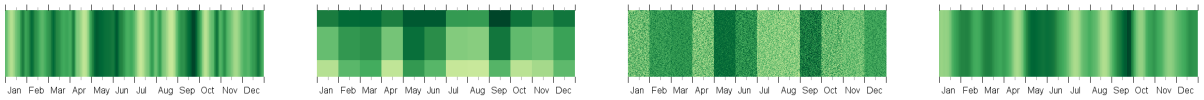


(a) Line graph—vertical position encodes value.

(b) Modified Stock Chart—line graph with monthly highs and lows (horizontal bars) and 30-day moving average (green line).

(c) Box Plot—monthly interquartile range (box), mean (horizontal line), and extrema (whiskers) are explicitly encoded.

(d) Composite Graph—bar chart encoding the monthly average is overlaid on a line graph of the raw data.



(e) Colorfield—a vertical stripe of color encodes daily values (darker green means higher sales).

(f) Color Stock Chart—each month has 3 color blocks encoding the maximum (top), mean (middle), and minimum (bottom).

(g) Woven Colorfield—the colorfield's pixels are randomly permuted within each month, creating discrete value "blocks."

(h) Event Striping—smoothed data plotted as a colorfield with outliers overlaid as vertical color bands (e.g. end of Sep, dark green).

Figure 4.2: Visual designs explored in this experiment. The first two rows of encodings use position to encode value; the bottom two use color. Conditions 4.2d, 4.2b, 4.2c, 4.2g, 4.2f, and 4.2h calculate and display different statistics at the per-month scale, which requires prior task knowledge (e.g. that the tasks will be performed at the scale of months).

month with the highest average, spread, and outlier numerosity). The encodings, detailed in the next sections, vary with respect to each design variable: primary visual variable (position versus color), the set of mapping variables (value statistic explicitly encoded, benchmark statistics explicitly encoded, and no explicit task statistics), and the computational variable defining the continuity of the encoding (continuous versus discrete). Figure 4.3 summarizes the performance predictions made by each design variable for each encoding.

4.3.1 Position-Based Encodings

Line graphs (Figure 4.2a) are the canonical approach for visualizing time series data using position. Position encodings support extracting exact values from a

visualization [Cleveland and McGill, 1984]. However, prior theory suggests that their ability to support summary tasks, such as comparing local averages, is limited [Correll et al., 2012a]. A more detailed theoretical evaluation of line graphs for visual aggregation is presented in Section 3.

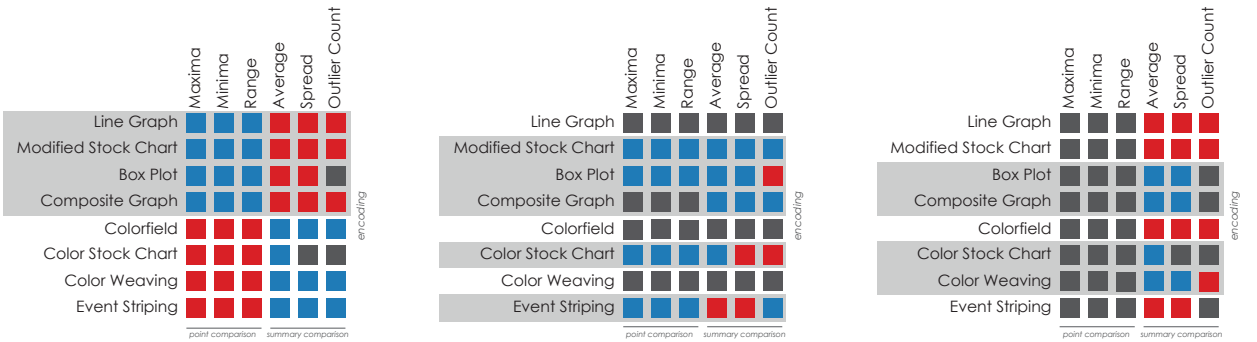
Modified Stock Charts (Figure 4.2b) supplement summary judgments in line graphs by layering a moving average over the original series. Extrema of discrete regions are encoded using range bars. I anticipate that the presence of the moving average will help with summary comparisons, albeit the continuous mean aggregation may still limit value extraction from discrete regions. The increased saliency of the extrema as discrete range bars will better afford minimum, maximum, and range comparisons. However, the amount of information encoded by the chart may cause issues of visual clutter.

For some comparison tasks, summary statistics may sufficiently summarize the necessary information in a series. **Box plots** (Figure 4.2c) discretely compute and visualize the range, interquartile range (IQR), and mean of the series for each temporal region. The explicit encoding of these statistics may better afford comparisons of the encoded statistics and reduces the amount of visual clutter in the display, but does so at the expense of the raw data—it is impossible to derive inferences beyond those explicitly encoded by the plot.

Composite graphs (Figure 4.2d) layer a line graph over a bar chart representing averages of discrete subregions. By explicitly mapping the mean value aggregated over each month, this approach may enhance the viewer's ability to extract averages from the visualization without inhibiting their ability to extract point-level information from the original series. Visually encoding the average may also provide a benchmark statistic for comparisons requiring average extraction, such as spread (average distance from the average).

4.3.2 Color-Based Encodings

Recent work demonstrates that color encodings, such as those used in **colorfields** (Figure 4.2e), may better support average comparisons than position encodings [Correll et al., 2012a]. Colorfields map each datapoint within a series to a point on a color scale, creating a one-dimensional heatmap, and adhere to the design constraints discussed in Sections 3.2.2 and 3.2.3. I anticipate that the perceptual system's ability to preattentively summarize color will support summary comparisons; however, colorfields will likely be less effective for point comparisons due to



(a) Visual Variable Predictions - Position encodings (grey rows) will better support point comparisons, whereas color encodings enable summary comparisons.

(b) Mapping Variable Predictions - Explicitly mapping statistics relevant to a task (grey rows) will better support comparisons, but limit the breadth of possible comparisons.

(c) Computational Variable Predictions - Discrete aggregation (grey rows) will aid summary comparison. Outlier counting is in part a hybrid task leading to different predictions.

Figure 4.3: We consider the design variables of a visualization in order to make predictions about how it supports different aggregate comparison tasks. We analyzed 8 time series visualization techniques using 3 variables, considering how each variable aligns with task requirements to hypothesize about their performance for 6 tasks. Blue squares indicate the variable aligns with the task, red show misalignments, and grey indicate no prediction.

the limited perceptual fidelity of color.

Color Stock Charts (Figure 4.2f) explicitly map the local extrema and average of each temporal range using color (average in the center, with top and bottom runners representing local maxima and minima respectively). This approach is comparable to the averaging aggregation discussed in Section 3.3. It simplifies the visual computation required to extract point values from a colorfield while preserving some high-level statistics from the series; however, the performance benefit of this mapping may be limited by the ability of the color encoding to communicate each statistic. Further, encoding only these tasks statistics sacrifices the ability to extract data about local features or other distributional information.

Color weaving [Albers et al., 2011, Correll et al., 2012a] (Figure 4.2g) breaks local structures in a colorfield by randomly permuting data values at the pixel-level within each month (see Section 3.3 for more information). This technique encodes a series as task-blocked woven glyphs whose pixel-level distribution mirrors the distribution of values in each month. Prior studies have shown that by breaking this local structure, color weaving improves the perceptual system's ability to summarize the encoded values [Correll et al., 2012a]. The

enhanced visual structures of color weaving may better afford average and spread comparisons; however, the increased difficulty of extracting a particular datapoint may complicate point comparisons using color.

Event striping [Albers et al., 2011, Correll et al., 2011] (Figure 4.2h) highlights outliers in the dataset by representing outlier values as broad "stripes" drawn over a smoothed colorfield representation of the original series (see Section 3.3 for more information). Explicitly mapping outlier values within the series visually boosts unusual values while the smoothed colorfield preserves the context of the series. Event striping provides an example of an encoding designed specifically for a given task. Its visual design is very similar to colorfields; however, the design choices made to support outlier identification thought increased visual saliency may influence how well the encoding supports other tasks, such as averaging, by distorting the visual distribution of data.

4.4 Methods

A series of experiments, one for each of six tasks (discussed below), compared the performance of viewers asked to make comparative judgments from time series data across the eight different visual encodings (described above). The experiments shared some common features across both tasks and encodings that we describe here. §4.5 describes the specifics of each experiment along with their results for clarity.

Each experiment focused on one aggregate comparison task. The encoding used to visualize the time series data was a between-subjects factor (see Figure 4.2). Accuracy (number of correct answers in a forced-choice setting) was the principle measure. Participants were instructed to be as accurate as possible and were allowed as much time as they needed to respond. I chose accuracy, rather than response time, as our performance metric, as accuracy allowed us to present stimuli that were more difficult, and thus more generalizable to real world datasets (see [Gleicher et al., 2013a] for more discussion of this choice).

Piloting revealed a small a learning effect—participants performance improved with greater exposure to the different visualization types. To partially counteract this, participants first completed an set of four stimuli designed to show the heterogeneity of difficulties present in the task and also to help participants develop an initial understanding of the task and encoding. These initial "training" stimuli were excluded from analysis. I also randomly interspersed stimuli that

were intentionally "easy" to serve as validation questions to gauge both validity of responses and participant understanding of task. For each task, a minimum acceptable accuracy on validation questions was determined based on piloting. Additional participants were recruited to replace participants failing to reach this level. Validation stimuli were otherwise excluded from analysis. Each participant saw a total of 44 stimuli (4 training, 6 validation, and 32 experimental) and was paid $1.00.

I recruited all participants using Amazon's Mechanical Turk infrastructure. Each participant saw a brief tutorial explaining the encoding they were going to see as well as the statistical property they were meant to compare. After the tutorial, participants saw a series of individual graphs which were exposed for 20 seconds, and after which were hidden. Participants could submit their answer at any point after the exposure of the graph. In very few cases (less than 4%) participants responded after the graph was hidden. I informed participants whether or not they got the previous question right, and gave complete feedback at the end of the task. After the participant completed the study, I collected demographics data.

Previous research has shown that Turk offers a reliable and diverse participant pool and provides a mechanism for rapidly recruiting a large number of participants [Buhrmester et al., 2011]. While there are known limitations to using Turk, with proper care in experimental design, Turk studies have proven to be a reliable source of human subjects data for understanding the efficacy of designs for information visualization.

4.4.1 Tasks

For each experiment, participants compared different statistical values, rather than calculating them precisely. For instance, rather than answering "what is the highest number in this time series?" (an extraction of a particular value), participants would answer "in which month does the highest number occur?" (an extraction and then comparison amongst values). I had no knowledge of our participants' statistical backgrounds, so the specific task questions had to be carefully phrased. For instance, range is the difference between the local minimum and maximum whereas spread sounds similar but considers variation amongst all points.

Little research exists exploring how to effectively ask lay audiences about outliers and spread. For these experiments, I needed to determine effective ways

of asking about these statistics. I generated candidate wordings by consulting the Simple English Wikipedia and evaluated these candidates in a pilot study on Mechanical Turk, asking participants to asses their comprehensibility and accuracy. For the studies reported here, I asked participants to compare the following statistical properties, using the following phrasing:

1. **Maxima**: Which month had the day with the highest sales for the year?
2. **Minima**: Which month had the day with the lowest sales for the year?
3. **Range**: Which month had the largest range of values?
4. **Average**: Which month had the highest average sales for the year?
5. **Spread**: Look at the average sales from each month. Which month had the sales which were the most spread out from their monthly average?
6. **Outliers**: Which month had the most unusual (outlier) sales days?

For all experiments, I presented time series of sales data for a fictional company over the course of a 12 month, 360 day "year" (to ensure months of equal length). For each task, participants were asked to make comparisons on the scale of months (e.g. which month had the highest average sales?). I believe this scale of data is substantial enough to make explicit calculation impossible given the 20 second exposure time available to participants, but sufficiently long for participants to comfortably complete each task.

For all of these tasks, since the viewer's specific goal was known to the designer, the answer could have been given directly. However, the goal of this study is to understand how visualizations work in settings where the designer may not know the exact goal of the viewer, or the viewer may have multiple goals.

4.4.2 Stimulus Generation

To help increase the validity of this experiment, data needed to be carefully controlled. The data needed to have a sufficient balance of apparent randomness so that it appeared realistic but did not adhere to a particular pattern. Additionally, I needed to control task difficulty and to vary the correct answer. To ensure that the participant responses correctly aligned with the task (and not with a related statistical property), we needed to explicitly decorrelate the correct answer from other statistics. For example, unless care is taken, the month with the highest

average often contains the highest single point. Explicitly decorrelating these statistics discourages strategies that may provide the right answer, but to the wrong question.

These constraints made it impractical to use real-world data. Therefore, I developed procedures to synthesize stimulus data. For all tasks, the data was created by blending together signals created by structured random noise [Perlin, 1985] that gave control over perceived noisiness and allowed for local adjustment to create variation. A set of applied constraints ensured that the resulting signals met the requirements of decorrelation and specific difficulties. The synthesizer fit each signal to these constraints, while minimizing the adjustment from the initial random signal. The final signals were created either by solving the corresponding constraint optimization problem or by locally adjusting signals to achieve the correct properties. The data was pregenerated for each experiment, and a post-hoc analysis verified that the data met the appropriate constraints. The same set of data signals was used for all encoding conditions within each experiment.

The stimuli for each experiment were generated from the data as pre-rendered images. Stimuli were presented to the viewer as losslessly compressed images to avoid variation in browser display. Color encodings used a green-yellow ColorBrewer sequential ramp [Brewer et al., 2003a].

Hardness Parameters

For each task, a set of parameters were associated with task difficulty derived from either past research or piloting. I leveraged three main dimensions of hardness: Δ, the difference in value between the correct month and the next highest months (lower Δ meaning more difficult to discriminate between months), the number of distractor months (the number of months with the value $x - \Delta$, where x is the correct highest value), and a qualitative dimension of noise. In each experiment, there were two levels of noise (“smoother” and “noisier” levels) and between one and four distractor months. Acceptable levels of Δ were highly dependent on the task and were modified for each experiment based on piloting. Each participant saw an equal number of each level of $\Delta \times$ noise, while the number of distractors was randomly sampled across all stimuli.

In our experiments and in piloting, each hardness parameter was highly correlated with performance overall, although different encodings could reduce or eliminate this correlation. For example, two box plots encoding signals with equal variation and extrema look identical regardless of the frequency of the underlying

signal, so noise would likely not impact task difficulty for box plots. Difficulty levels were considered in our statistical analyses.

4.5 Experiments and Results

This section detail each experiment (one per task) and the corresponding results. For each experiment, I performed an Analysis of Covariance (ANCOVA) to determine the effect of encoding type on accuracy. The model also tested interaction effects between encoding type and the aforementioned hardness parameters (Δ, distractor count, and noise level). Hardness parameters generally had highly significant effects in the expected direction (noisier signals underperform smoother signals, smaller Δs are more difficult, etc.). As a result, I omit these factors from reporting unless unusual. For significant results, I performed Tukey's Test of Honest Significant Difference (HSD) with $\alpha = 0.05$ to assess relative performance of the encodings. I also performed post-hoc mean squared contrast tests to verify significant differences within these clusters. Figure 4.4 summarizes these findings.

Including piloting and the main tasks, a total of 582 participants were recruited (306 male and 276 female, μ_{age}=31.3, σ_{age}=10.3). A Student's t test found no significant differences in performance across gender (μ_f=60.1%, μ_m=64.4%, $p = .0938$). For each experiment, 8 participants were recruited per encoding, totalling 64 participants for tasks evaluating all eight encodings, 56 for the spread experiment (which excluded color stock charts), and 48 for the outlier experiment (which excluded box plots and color stock charts). If a participant failed to achieve acceptable performance on validation stimuli, I discarded their data and recruited additional participants for that condition. Across all experiments, 37 additional participants were recruited for this reason. 397 total participants were recruited for the main experiments. Although accuracy was the performance metric, I also tracked response time for each task and found the longer a participant spent on a particular question, the *more* likely they were to be incorrect ($b = -1.6\%$ accuracy/sec, Pearson's $r = 0.83$).

Maxima

For this task, participants were asked to locate the month containing the day with the highest absolute sales. Maxima within the series were created by amplifying the natural peak in the base random series and constraining all remaining values

		Maxima	Minima	Range	Average	Spread	Outliers
LG		87.5%	78.9%	74.2%	47.7%	48.8%	36.7%
MSC		88.7%	96.1%	91.8%	56.3%	39.7%	34.0%
BP		75.0%	93.8%	88.5%	68.8%	85.0%	X
CG		93.0%	88.3%	77.0%	85.9%	53.8%	33.6%
CF		59.4%	56.6%	48.8%	60.5%	57.8%	31.3%
CSC		69.9%	73.4%	64.8%	70.3%	X	X
WC		43.0%	45.7%	38.7%	77.7%	71.3%	23.0%
ES		61.7%	59.4%	44.1%	52.3%	42.2%	66.8%

Figure 4.4: A summary of our experimental results. All measures are in accuracy across all participants. Gray rows indicate position encodings; white indicate color encodings. Gray columns indicate summary comparison tasks; white columns indicate point comparison tasks. An "X" indicates that the encoding does not afford that task. and so no experiment was conducted for this combination of task and encoding. Since performance is not strictly comparable across tasks, cell color encodes the number and direction of standard deviations from the task mean: $\leqslant -1$, (-0.5,-1) , [0.5,-0.5] , (1,0.5) , $\geqslant 1$.

to be at least Δ less. Picking the month with the highest average sales could be a confounding strategy, especially in the color conditions where detecting individual points is difficult. As a result, the month with the highest average sales was decorrelated from the month with the highest absolute sales. I sampled evenly across Δs of 1,2,3,4, with validation stimuli at $\Delta = 20$.

Results: Encoding had a significant main effect ($F(7, 2016) = 45.8$, $p < .0001$). Generally, position encodings outperformed color encodings, with one exception. Box plots significantly underperformed all other positional encodings ($F(1, 2016) = 24.5$, $p < .0001$), and were not statistically significantly different from the color stock chart ($F(1, 2016) = 1.70$, $p = .1930$). The remaining color encodings performed significantly worse than the color stock charts ($F(1, 2016) = 28.8$, $p < .0001$) and the position encodings as a group.

These results support **H1**—as this was a point comparison task, we expected

position encodings to outperform color encodings, which are not as accurate for extracting exact values. There is partial support for **H2**—color stock charts, which were the only color encoding to explicitly encode the maximum value in each month, outperformed other color encodings. However, box plots, which were one of two position encodings to explicitly encode maximum values, undeperformed the other position encodings. This may be due to biases arising from visual properties of box plots that have been shown to impact the perception of whisker values [Behrens et al., 1990]. An alternative explanation is that viewers were unused to whiskers encoding range; however, overall performance, especially on validation stimuli, suggests that this is unlikely.

Minima

For this task, participants were asked to locate the month containing the day with the lowest absolute sales. This task was functionally identical to the Maxima task—questions about "highest" were changed to "lowest" and the stimuli were derived using the same constraints as the Maxima task. Despite the similarities in the tasks, prior work [Sanyal et al., 2009] suggests that there are differences in performance between the two and that different encodings may be appropriate for detecting minima versus maxima.

Results: Encoding had a significant main effect ($F(7, 1984) = 59.1$, $p < .0001$). Within groups, line graphs significantly underperformed the rest of the position encodings ($F(1, 1984) = 25.5$, $p < .0001$), and were only marginally better than color stock charts ($F(1, 1984) = 2.76$, $p = .0966$). The remaining color encodings proved significantly worse than the color stock charts ($F(1, 1984) = 46.1$, $p < .0001$), and also the position encodings as a group. Unlike other experiments (even the Maxima experiment reported above), the noisiness of the signal had no significant effect on accuracy ($F(1, 1984) = 0.18$, $p = .6725$).

As in the Maxima experiment, these results support **H1**—position encodings tended to outperform color encodings. **H2** was more strongly supported than in the Maxima experiment—box plots and modified stock charts, which both explicitly encode monthly minima, outperformed line graphs, and color stock charts outperformed all other color encodings.

Range

For this task, participants were asked to locate the month with the largest range of sales—the largest gap between the maximum day and the minimum day. Initial piloting showed that participants frequently confounded the range with the maximum. To avoid confounds with the maximum and the statistically related measure of spread, we explicitly decorrelated these three quantities. This task proved more difficult for participants than either of the extrema tasks as it required participants to compare the difference between two points. To avoid floor effects, stimuli were sampled from Δs of 4, 7, 10, and 15, with validation stimuli at $\Delta = 20$.

Results: Encoding had a significant main effect ($F(7, 1984) = 59.3$, $p < .0001$). The color encodings all significantly underperformed the position conditions. Encodings which explicitly encoded extrema performed significantly better than the other encodings of their type: color stock charts outperformed the other color encodings ($F(1, 1984) = 45.8$, $p < .0001$), and box plots and modified stock charts outperformed the other positional encodings ($F(1, 1984) = 28.9$, $p < .0001$).

As the range task is a pairwise point comparison task, these results support **H1**—position encodings afford greater fidelity in extracting point values than color encodings. **H2** is also supported. Box plots, modified stock charts, and color stock charts all explicitly encode local extrema values and all outperformed other encodings with equivalent visual variables. Additionally, with box plots, a possible strategy for completing this task would be to compare the length of the box and whiskers (one value) moreso than the difference between extrema (two values). Validating this strategy is an interesting potential direction for future work.

Averaging

For this task, participants were asked to compare means of months. In piloting, the highest *average* value was often confused with the highest *absolute* value, so these values were decorrelated in the stimuli. Stimuli were evenly sampled from Δs of 1,2,3,4, with validation stimuli at $\Delta = 20$.

Results: Encoding had a significant main effect ($F(7, 1984) = 22.6$, $p < .0001$). The three encodings which explicitly encoded discrete monthly averages (the composite graph, box plot, and color stock chart) and discretely blocked woven colorfields significantly outperformed the remaining encodings ($F(1, 1984) = 122$, $p < .0001$). Within clusters, there were several pairwise results of interest. In particular,

composite charts outperformed woven color fields ($F(1, 1984) = 4.24$, $p = .0395$), and regular colorfields outperformed line graphs ($F(1, 1984) = 11.4$, $p = .0008$).

These results partially support **H1**—colorfields, which support preattentive methods of summarization, outperformed line graphs, which do not. The data also partially support **H2**—composite graphs, which explicitly encode mean, outperformed woven colorfields, which do not. However, color stock charts, which also explicitly encode monthly averages, did not outperform woven colorfields, which are designed to support ensemble statistical processing (e.g. mean and variance) for visual aggregation. The data more fully support **H3**—all of the encodings which discretely aggregated the data per-month outperformed the other encodings.

Spread

For this task, participants were asked to compare the spread of each month. Since strict control over standard deviation requires complex optimization, I measured spread using the more practical related statistic of absolute deviation ($\frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$). Linear scaling about the monthly mean was used to tune the absolute deviation of individual months to fit our constraints. Even so, it is difficult to generate large differences in variation as each point must remain in the [0,100] interval. As "spread" is an ambiguous term, I decorrelated the month with the highest absolute deviation from the month with the largest range. To avoid floor effects for what was in piloting a difficult task, stimuli were evenly sampled from Δs of 2,3,4, and 10, with validation stimuli with $\Delta = 15$—the largest that could be reliably generated in sufficient numbers.

Results: Encoding had a significant main effect ($F(6, 1736) = 36.8$, $p < .0001$). Box plots outperformed color weaving ($F(1, 1736) = 13.7$, $p = .0002$), which in turn outperformed all the remaining encodings ($F(1, 1736) = 50.0$, $p < .0001$). Standard colorfields outperformed both boosted colorfields and modified stock charts ($F(1, 1736) = 17.9$, $p < .0001$). Noise had only a marginal effect on performance ($F(1, 1736) = 3.39$, $p = .0656$), and the number of distractors had no significant effect ($F(3, 1736) = 0.847$, $p = .4679$).

These results provide partial support for **H1**—woven colorfields performed better than nearly all other encodings, as weaving allows for quick visual summarization of the variance of a region despite not explicitly encoding this value. **H2** was fully supported—only box plots explicitly encoded a statistical variable that was highly correlated with absolute deviation (IQR) and best supported this task. There was

little support for **H3**—while the top two encodings both explicitly blocked data together into months, composite graphs were not statistically different from any of the other encodings despite being blocked with respect to a benchmark statistic (average).

Outliers

For this task, participants were asked which month contained the highest number of outliers. The task required both extracting summary statistics to characterize the distribution of values and numerosity estimation of points violating these statistics. Outliers were generated by amplifying days in the original random signal that varied largely from the series mean to between 2.25-2.75 standard deviations from the mean. To avoid visual "plateaus" where consecutive outliers appear as on data point, outliers were spaced at least 3 days apart and no month contained more than 8 outliers. Spread can confound outlier count, so the month with the highest absolute deviation was decorrelated from the month with highest number of outliers by reducing the absolute deviation of the high outlier month. To avoid confounds between the month with the greatest number of outliers and the month with the largest outlier, the largest value was decorrelated from the month with the most outliers. For this task, Δ means that if the winning month had x outliers, the other months had at most $x - \Delta$ outliers. Stimuli were evenly sampled from Δs of 1,2,3,4, with $\Delta = 5$ for validation.

Results: Encoding had a significant main effect ($F(5, 1488) = 28.3$, $p < .0001$). A Tukey HSD showed two clusters, with event striping outperforming all other displays ($F(1, 1488) = 127$, $p < .0001$). The only other significant difference among conditions was the color woven display, which under-performed all of the remaining conditions ($F(1, 1488) = 11.4$, $p = .0008$).

These results support **H2**—by increasing the saliency of outliers, event striping supported numerosity judgments of a statistically complex value better than all other encodings.

4.6 Discussion

The results of these experiments, summarized in Figure 4.4, confirm that different designs support different tasks. The three identified design variables provide a

mechanism for identifying elements of these designs that may be responsible for these differences.

- The choice of *visual variables* can allow the viewer to perform aggregation visually in cases where the quantity of interest is not explicitly encoded, or can facilitate discrimination between values which *have* been explicitly encoded.
- The choice of *mapping variables* can help the viewer by explicitly encoding the quantity of interest, but only if the relevant information is known.
- The choice of *computational variables* can align displayed information with the viewer's task if the task is known.

The results support the importance of these decisions: the predictions of how choices in these variables should influence the performance of the resulting designs are supported. For example, by matching display and task granularity, composite graphs, which display discrete monthly averages rather than as a continuous moving average, significantly outperformed modified stock charts for average comparison. They also suggest that substantial tradeoffs occur when designing for a specific task. For example, event striping underperformed standard colorfields for all summary tasks except for outlier detection, despite their visual similarity.

These results further indicate interactions between design variables. For example, explicitly encoding relevant statistics may not overcome natural deficits in point value extraction in color displays, as with color stock charts for extrema and range tasks. In contrast, the affordances of color weaving for visual aggregation outweigh these issues with color for average and spread. This suggests the potential for designs informed by perceptual mechanisms.

The interaction between the visual variables used to represent data and the statistics that are encoded by these variables helps designers better reason about the effect of visual clutter in visualization. Traditionally, designers reason about clutter as a function of visual variables. Considering computational variables expands our ability to understand trade-offs in visual clutter. If more information is available (e.g. computation is done at a high level of granularity), viewers are better able to accomplish specific tasks and process less information overall. However, reducing visual clutter through computational variables forces the analyst to consider the statistics actually used and how those statistics are applied (i.e. the

mapping variables used). This reduces the number of tasks that can be readily accomplished by the viewer.

The results with respect to visual variables are particularly poignant for this dissertation. These results confirm the hypotheses in Section 3, namely that color supports visual aggregation tasks combining data across all values more efficiently than position-based encodings such as line graphs. While performance differences between position and color for point comparisons reinforces prior assumptions about the utility of color in visualization, the superior performance of color for summary tasks suggests their utility in supporting visual aggregation. For example, comparing averages using a woven colorfield is roughly as effective as explicitly representing an average in a color stockchart, both of which significantly outperform the continuous positional averages in a modified stockchart. The results of these experiments indicate that low-level mechanisms for processing summary information may vary across visual features, and the effectiveness of a visual encoding may differ for point tasks, as in traditional graphical perception, and visual aggregation tasks. Color may be particularly well suited for visual aggregation as it allows the viewer to readily combine information across multiple datapoints.

Value for Design: Matching designs to tasks is important. Beyond providing empirical evidence of this importance to aggregation tasks in time series visualization, these findings provide actionable advice in how to consider such matching. As no design is likely to be effective for all tasks, designers must consider not only their understanding of the target tasks for a display, but also how specifically they want the display to support this task, at potential cost for other tasks.

By identifying three key design variables, these experiments provide specific questions for a designer to consider in matching visualizations to tasks. For aggregation tasks, the variables make explicit three key choices for composing a visualization design. This work provides not only a set of questions to consider in matching designs to tasks, but also predictions as to how the choices will impact performance for different tasks. The variables can guide a structured exploration of the design space, for example, to assess potential of different designs. Designers can use these trade-offs to create visualizations that better support specific analysis workflows by making empirically-grounded encoding decisions or building systems with multiple views using complementary encodings to support a broad range of tasks.

The possibility of effective visual aggregation provides new opportunities for designers to create visualizations that support aggregation tasks. The design variables provide connection between the emerging perceptual science and design goals, coupling task features to performance predictions. This work demonstrates the benefits and costs of different design approaches enabling designers to make informed choices about using each approach.

4.6.1 Limitations and Future Work

These experiments considers a small set of encodings and tasks for a specific but common data type. I believe these findings generalize to a wider range of situations, such as in genomic analysis or in designing encodings for two-dimensional data, but have not confirmed this empirically. More exhaustive testing of this theory is limited not only by the practical problem of running a vast number of experiments, but also in choosing tasks that can be assessed in a controlled experimental setting.

This work does not consider the various costs and tradeoffs in combining design elements. For example, a design encoding multiple statistics may support multiple tasks, or cause visual clutter that reduces its effectiveness at any one. Similarly, it does not consider the costs of misalignment between design and task. For example, does presenting data aggregated by month hurt performance at questions about weeks or data at weeks hinder tasks at months? In the future, I hope to better understand the tradeoffs of misalignment.

In practice, visualizations are often interactive, allowing the viewer to specify their task, rather than requiring the designer to make assumptions about what information a viewer is interested in. This work focuses on static visualizations, emphasizing the importance of aligning tasks and design. However, extending this work to consider interaction, including identifying new design variables, is important future work. The next chapter will show how interaction might use several of the encodings tested here to support complete analysis workflows.

5 THREE SYSTEMS FOR SCALABLE VISUALIZATION

The previous experiments show that viewers can effectively estimate aggregate values like mean and variance from color better than with position encodings. An underlying motivation for designing for visual aggregation is creating visualizations that support aggregate analyses at *scale*. The previous experiments do not necessarily demonstrate how color can support effective analysis as series grow longer or as more series are analyzed at once. While empirical methods could theoretically confirm the robustness of color for visual aggregation at scale, careful control over data and other potentially confounding factors becomes increasingly difficult as more data is introduced. Additionally, the goal of understanding visual aggregation for visualization is to design systems that support analysts in solving real problems.

Instead, I demonstrate the how color can be used to support visual aggregation at scale by developing systems that use these techniques to visualize data at scale. I will prove their utility through a series of case studies from domain experts that show how the systems support aggregate analysis at scale. The systems discussed here address three real-world analysis problems: comparing gene sequences, understanding language patterns in text documents, and evaluating machine learning results across molecular surfaces. These systems have dramatically increased the scale at which data can be analyzed in these applications. Through these examples, I highlight the generalizability of the work presented in Chapters 3 and 4 to domains beyond time series analysis. I also show the need for solutions that support visual aggregation tasks at scale.

5.1 Scalable Sequence Alignment Visualization in Genomics

The first domain application I will discuss is sequence comparison (also a model problem in Chapter 3). Sequence comparison is a fundamental task in the biological sciences. Scientists often need to compare genomic sequences, for example, to understand evolution, to infer common function or to identify differences. Because sequences are often too long for manual examination, scientists rely on alignment tools that automatically identify matching subsequences. Tools for visualizing these alignments are commonly used when performing sequence comparison. A

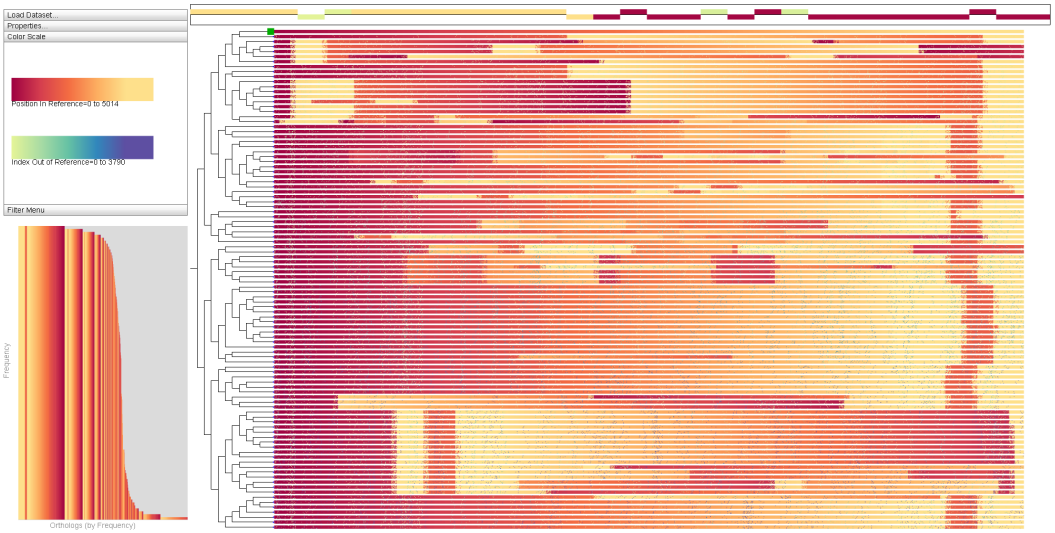


Figure 5.1: Sequence Surveyor visualizing 100 synthetic genomes generated by an evolution simulation. Each genome is mapped to a row and genes are ordered by position. Color encodes the position of the gene within the chosen reference sequence (top row, indicated by the green box). Genes are aggregated, with each block's texture reflecting the overall distribution of colors in that block. The dendrogram shows the phylogeny of the data set while the histogram shows the frequency distribution of orthology group sizes.

variety of approaches for displaying and exploring alignments exist, and have been incorporated into a wide variety of tools (see Procter et al. [2010] for a survey of popular tools). The amount of sequence information available is growing rapidly. Scientists are exploring larger numbers of genomes and longer genomes. However, most tools by design focus on providing in-depth exploration of a small set of sequences for predefined tasks. Focusing on point-level details obscures the task of tracing aggregate trends in large datasets (cf. Figure 5.6a). Looking at larger datasets at this fine level of detail is overwhelming, and does not scale to growing datasets.

Most comparative biological sequence visualizations are variants of four basic designs: juxtaposed value representations that leverage color or bar graphs to visualize raw sequence data, dot plots (scatter plots with sequence position on the axes), synteny views (which indicate matches relative to a reference), or parallel-coordinate views (which show alignment by drawing connections between sequences). These tools are conventionally designed explicitly for particular tasks and datasets. For example, the Broad Viral Viewer [Jen et al., 2009] provides for comparison of dozens of viral genomes, while Mauve [Darling et al., 2004] is useful for a half-dozen or so medium (bacteria-sized) genomes and Mizbee [Meyer

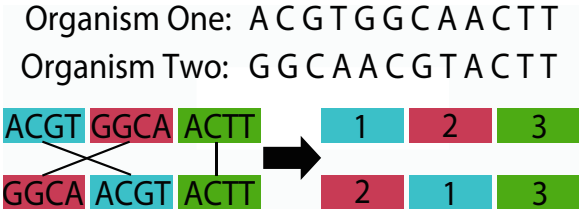


Figure 5.2: Genome alignments are computed from genome sequence data by identifying matching subsequences (left), known as *orthologs*. Ortholog groups are identified by integer tags (right). Sequence Surveyor uses orthology data to explore genome alignments. In real data, orthologs are far longer than four nucleotides.

et al., 2009] supports pairwise comparisons of larger genomes. Scalability limits may come from memory or performance issues as tools get bogged down with too much data. But often, scalability in biological visualization is hindered because the visual design breaks down: the displays simply become ineffective when there is too much data to display in detail.

Instead, leveraging the insights developed in Chapter 3, I developed the Sequence Surveyor system, shown in Figure 5.1, with collaborators in computer science and bioinformatics, to provide flexible overviews of large whole genome alignment datasets. This design provides a first proof-of-concept that embodies the advantages of color for scalability and visual aggregation hypothesized in Chapter Understanding Perception for Visual Aggregation, demonstrating how the proposed method of aggregation retains salient features as the dataset scales up. Sequence Surveyor allows scientists to examine patterns and trends in multiple genome alignment datasets of over 100 bacterial genomes (Figures 5.9 and 5.11), roughly ten times what conventional connection-based visualizations support ([Darling et al., 2004]). Because we cannot know *a priori* the kinds of questions the data will be used for, my approach provides flexible mappings that allow different kinds of patterns and trends to be made salient as the viewer explores the data. Mechanisms for filtering, zooming, and reordering the data help scientists cluster data according to different properties of the dataset to find large-scale features and connect these to smaller sets of details for further exploration, while color encodings support visual summarization and help mitigate visual clutter.

5.1.1 Biological Background

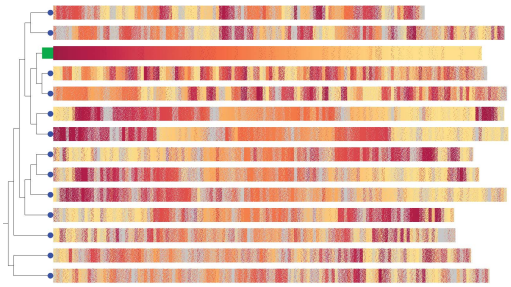
The primary task of alignment visualization involves viewing matching regions between a set of sequences. Alignment visualization is useful for many types of

sequence data, such as proteins and RNA. In this section, I focus on whole genome alignments (alignments that map genes across different genomes). Genomes are segmented into functional regions (i.e. genes), and each sequence is represented as an ordered list of genes (cf. Figure 5.2). Alignments identify groups of matching (evolutionarily-related) genes, known as *o*rtholog groups, present in one or more genomes in the dataset. These groups, as computed by the alignment, serve as identifiers for related classes of genes. This type of data is technically called gene-level alignment, but, for the purposes of this paper, the more general term *alignment data* will be used. While pairwise alignments (comparisons of two sequences) are the most common form of alignment, alignments between multiple sequences are becoming increasingly important as sequence information becomes more abundant and better understood.

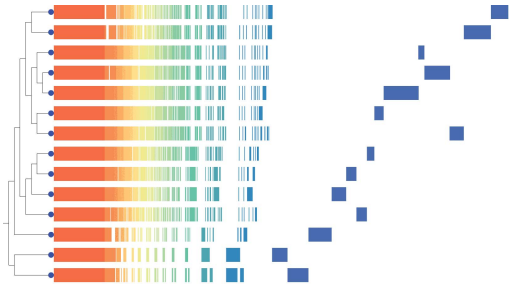
One important and complicating aspect of visualizing whole genome alignments is that there are potentially thousands of related elements which may occur in different orders and copy numbers in each genome. When trying to understand an alignment, a scientist often needs to consider other information such as the details of the sequences, annotation data, expression information, and other information generally associated with exploring a single sequence (see [Peeters et al., 2004] for a survey). One of the primary tasks that arises in whole genome alignment data is understanding patterns of *conservation*—the preservation of orthologous genes between species—in the dataset. Understanding patterns in conservation can allow scientists to make conjectures about evolution and common function of different species. Conservation can help answer questions about the presence of genes at different loci in the genome, origins of replication (i.e. where rearrangements of genes between different species begin), and proportions of the genome shared between different organisms. These types of general questions make whole genome alignment data important: by understanding conservation between genomes, we can begin to understand how different gene sequences function within an organism.

5.1.2 Solution Overview

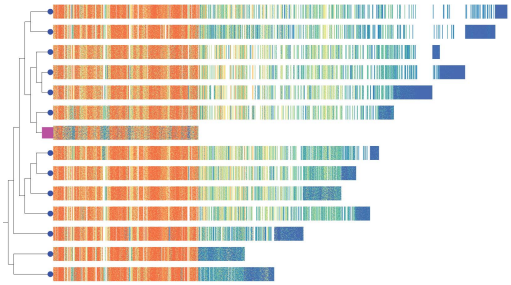
In Sequence Surveyor, multiple genome alignment data is visualized as horizontal tracks, with each row corresponding to a sequence and rows separated by whitespace to support visual search and summarization (see §3.2). Data within each track is visualized using color, according to the aggregation method outlined in



(a) Color by reference



(b) Membership frequency coloring and sorting by grouped frequency



(c) Color by membership frequency and order by position in reference

Figure 5.3: Sequence Surveyor provides flexile color and position mappings that address different questions about data. (a) Coloring by the position of genes in a reference genome (green rectangle) shows that genomes most similar to the reference, indicated by preserved color gradients, are not those most closely related (the adjacent genomes). (b) Frequency-based mappings can highlight patterns of presence and absence across species. Bands of genes create conservation "fingerprints" for each genome that align well for closely related genomes. (c) Membership frequency (most (red) to least frequent (blue)) combined with reference ordering (magenta box) highlight uncommon regions of the reference: green columns in the reference show that other species that share some relatively unique regions.

§3.3, to facilitate different visual aggregation tasks.

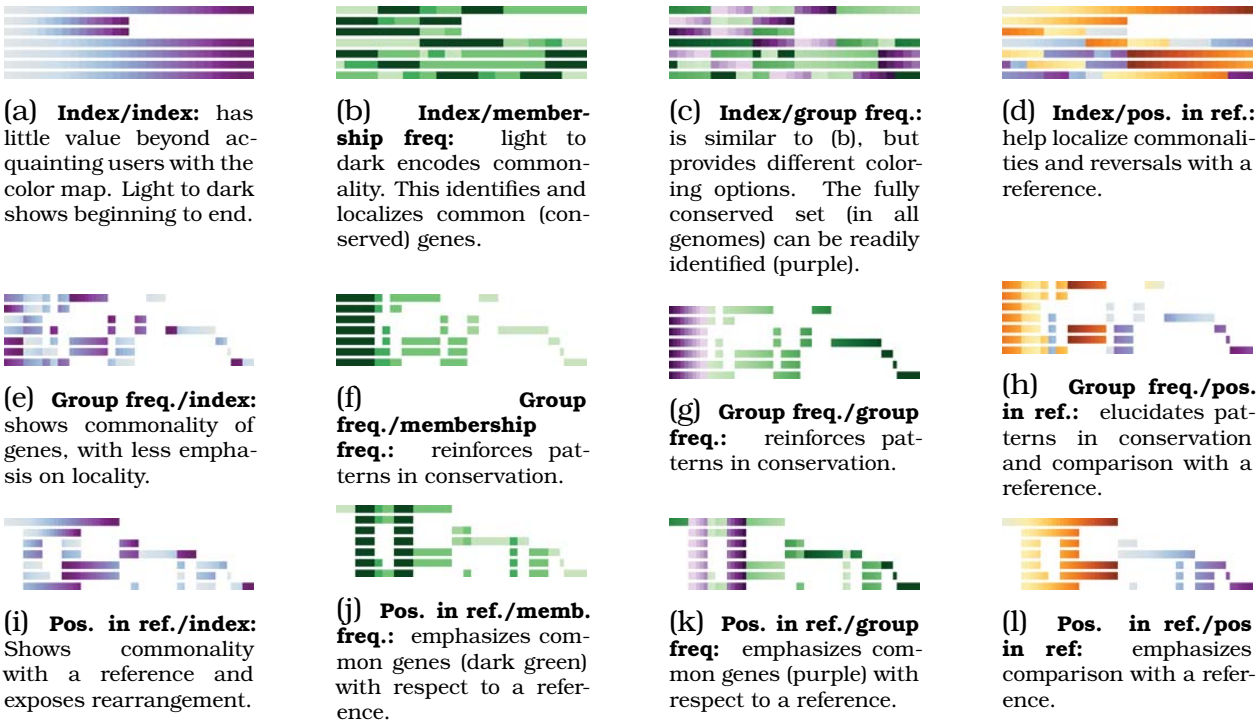
Gene properties are encoded within each track using both x-axis position and color to facilitate visual aggregation. A number of mappings are shown in Figure 5.4 and detailed in §5.1.4. Position encodings follow a left-right reading order, and can be mapped to derived axes generated by particular properties of the data to facilitate novel insights. Derived positions involve sorting orthologous genes by frequency (the number of genes matching it) or position in a selected reference genome. Color can also be mapped to either raw properties of the dataset or to other derived axes. Flexibility in these mappings allow to analysts to create meaningful patterns and clusters within the data. For example, positional ordering coupled with position in reference coloring identifies common genes and their rearrangements across the dataset, while ordering by frequency and coloring by position gives a sense of the conservation between sequences. Figure 5.3 illustrates some potential mapping combinations on real data.

Genes within each sequence are visualized as a series of screen-space blocks according to the methods presented in §3.3. Interactively switching between aggregate glyphs allows the viewer different kinds of aggregate insight into the dataset. Various other interaction techniques support point-level analysis. For example, hovering the pointer over a block highlights blocks that share common genes and explicitly enumerates the genes within the block. A histogram of gene frequencies and a phylogenetic tree provide linked views that highlight subsets of the data. Zooming and detail displays help connect large patterns to these specific details.

5.1.3 Design

With Sequence Surveyor, I wanted to create an alignment visualization tool able to scale to large numbers of genomes (dozens or more) and large genomes (thousands or more genes per sequence). At the same time, the system must handle the full complexity of these alignments, including rearrangements, reference dependent and independent tasks, and gene repetition. Furthermore, the study of such massive datasets is new: the questions to be considered are wide-ranging and this display may offer the opportunity to discover new questions.

Sequence Surveyor was designed to take advantage of perceptual processing to provide scalable overviews of data. While emphasizing visual aggregation in an overview system may come at the expense of providing the details usually



(a) **Index/index:** has little value beyond acquainting users with the color map. Light to dark shows beginning to end.

(b) **Index/membership freq:** light to dark encodes commonality. This identifies and localizes common (conserved) genes.

(c) **Index/group freq.:** is similar to (b), but provides different coloring options. The fully conserved set (in all genomes) can be readily identified (purple).

(d) **Index/pos. in ref.:** help localize commonalities and reversals with a reference.

(e) **Group freq./index:** shows commonality of genes, with less emphasis on locality.

(f) **Group freq./membership freq.:** reinforces patterns in conservation.

(g) **Group freq./group freq.:** reinforces patterns in conservation.

(h) **Group freq./pos. in ref.:** elucidates patterns in conservation and comparison with a reference.

(i) **Pos. in ref./index:** Shows commonality with a reference and exposes rearrangement.

(j) **Pos. in ref./memb. freq.:** emphasizes common genes (dark green) with respect to a reference.

(k) **Pos. in ref./group freq:** emphasizes common genes (purple) with respect to a reference.

(l) **Pos. in ref./pos in ref:** emphasizes comparison with a reference.

Figure 5.4: Sequence Surveyor views shown on a toy dataset, each combining a position mapping and a color mapping. Different mappings make different patterns emerge in the color field. Subfigure rows show different position mappings, columns show color mappings (see subfigure captions). The top genome is the reference for both coloring and position. Nucleotide-level start position mappings do not apply in this example.

considered in traditional visualization tasks. While some low-level detail can be obtained in Sequence Surveyor through interaction, supporting full multiscale analysis is outside of the scope of this project.

The design of Sequence Surveyor is derived from the perceptual analysis presented in §3.2. This analysis suggests a colorfield design, rather than the designs leveraging connection or position, which are more common in alignment visualizations. By encoding orthology using color instead of explicit connections, to some degree, I exchange accurate identification of individual connections for scalability. While color fields allow patterns and trends to "pop out" and visual structure to be estimated efficiently in large displays, this requires determining what patterns and structures are of interest. As this is not known *a priori*, I instead define flexible mappings (§5.1.4) that allow user control and exploration over how visual features map to data.

However, the set of tasks supported by Sequence Surveyor are a superset of

the general theoretical tasks introduced in §3.2. For example, in addition to finding high-level structures supported by visual summarization or searching for individual values, viewers may care about specific repeated subpatterns in the dataset, known as *motifs*. They may also care about detecting this pattern in reverse, known as an *inversion*, and detecting repeated patterns robust to some level of noise. These motifs (and the degree of noise in a motif between different organisms) can indicate important evolutionary relationships or sets of genes that collectively support an important biological function. Alternatively, a biologist might be interested in estimating the approximate similarity (or lack thereof) between different genomes or sets of genomes (e.g. 5.9). This similarity may be both in left-right order across color gradients and unordered across an entire genome.

I also provide different schemes for aggregation (§3.3), not only allowing the system to scale to data sizes much larger than the number of pixels, but also controlling visual clutter and if trends or outliers are more significant to a particular exploration. This design demonstrates the utility of this method of aggregation and provides examples of where different aggregate representations provide different insight into the data. Other aspects of the design that support aggregate analysis for genomic data include mechanisms for arranging the data for effective comparison (§5.1.5) and interaction techniques to aid exploration and connect to details (§5.1.6).

5.1.4 Mapping

Colorfields allow us to present a large field of information, yet have certain patterns and details readily emerge. However, the properties assigned to color and position within the colorfield will determine what kinds of information will form noticeable patterns. Unfortunately, the specific information that a scientist is looking for is unknown a priori. A scientist may have many different kinds of questions, and new questions will emerge as they begin to explore new datasets. Sequence Surveyor provides a flexible set of mappings from the data to the display, giving the analyst control over the information encoded by horizontal position and color (cf. Figure 5.3). While many of these encodings have appeared in previous tools, this approach provides a generalized view of alignment data through these interactive mappings (Figure 5.4).

Sequence Surveyor maps genes according to several natural and derived prop-

erties of the data. *Gene index* is a gene's position relative to the ordering of genes in the sequence, while its *start position* is its location in terms of the actual DNA (the lengths of different genes and gaps between them are included). The *position in reference* represents the gene index of a matching gene in a selected reference sequence. Frequency properties measure how many other genes match a given gene (the size of its ortholog group) and are important for understanding conservation across a dataset. *Membership frequency* counts how many different genomes contain at least one instance of an ortholog, while *gene frequency* counts how many times an ortholog occurs (this is typically greater than membership as a genes may be duplicated within genomes as paralogs). *Grouped frequency* further orders orthologs by the sets of genomes that contain them.

Any of these six properties may be mapped to color. Four may be mapped to position (of the frequency properties, only grouped frequency provides a total ordering required for a horizontal mapping). Different configurations of these properties make different kinds of patterns apparent in the display. Several of these mappings reflect the data mappings provided by common genomic visualization tools, while others present more unusual views of the data. See §5.1.7 for a discussion of how these mappings can be used in biological exploration. Genomes can be interactively reordered to facilitate comparison between different sets and reorganize the data as different patterns are revealed.

Sequence Surveyor provides a series of eleven different color schemes: ten Color Brewer [Brewer et al., 2003a] ramps and one flat gray to remove visual clutter from irrelevant data. For several mappings, two color schemes are chosen to highlight data belonging to different semantic sets. For example, the position in reference mapping uses one ramp for orthologs that match the reference, and a second ramp for those that do not (the solid gray ramp is particularly useful for this). In addition to providing aesthetic control, the color schemes provide the user with a certain degree of control over pop-out by allowing them the choice of color assignment for different attributes of interest.

Color mappings provide visual patterning over the data: blocks with similar properties map to similar colors. This creates color gradients in the display that encode large-scale trends. Breaks in the gradients can pop-out to highlighting variations in these global trends. It also supports the preattentive association of various data points by creating large fields of color at regions of high similarity. Sorting mappings take advantage of the visual system's predisposition to clustering. Sorting according to particular parameters clusters visually on these parameters,

imposing an orthology-based structure to the visualization: orthologous sets become spatially colocated. This allows the viewer to quickly identify regions of interest to scan for patterns and creates new high-level structures that can be readily summarized to support visual aggregation tasks.

However, non-traditional orderings of genes within colorfields can potentially hinder visual search tasks: the viewer may not know where particular data might be found. By default, Sequence Surveyor orders genes according to a conventional relative ordering. This ordering conforms to the common model of genomics data and can help anchor and contextualize exploration as the user interacts with the data [Liu and Stasko, 2010].

5.1.5 Data Display

The colorfield display presents an interactive overview of the entire dataset. However, additional coordinated visualizations provide supplemental insight into the data. Two primary coordinated visualizations are used: a phylogenetic tree to contextualize evolutionary patterns between organisms and a histogram that communicates the distribution of gene conservation throughout the dataset.

The phylogenetic tree shows the evolutionary relationship between genomes within the dataset—the evolutionary lineage of the different organisms present in the dataset is encoded using a tree. Using the phylogenetic data as to order the genomes within the dataset clusters genomes according to an approximation of their pair-wise similarity. This organization colocates genomes that are likely to share a large number of genes.

Overall gene-level information is summarized in the histogram. The height of histogram bar represents gene frequency and orthologous gene groups are sorted and aggregated according to the same frequency metric. The resulting shape conveys the overall frequency distributions of genes within the data set. A lasso-selection filter highlights interesting frequency clusters in the main display. Brushing in the histogram coordinates with the phylogenetic tree by highlighting branches up to the most recent common ancestor shared by the genomes conserving the brushed genes. This interaction also highlights these orthologs in the primary display.

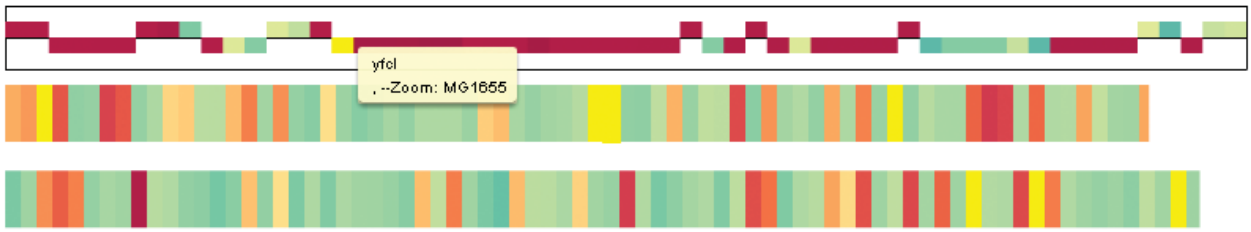


Figure 5.5: Overview+detail zooming manages the non-locality issues arising in multiple genome alignments. As the user mouses over blocks in the genome view, component genes of those blocks are visualized in the zoom window (top), positioned vertically according to the strand where they are found and horizontally according to the position mapping. Zoom can be locked onto a block for interactive functionality.

5.1.6 Interaction: Exploration and Zooming

Sequence Surveyor is designed to support visual aggregation tasks, allowing the viewer to identify high-level patterns in large scale data. However, the low-level details of the underlying data are still significant to exploring large datasets. Sequence Surveyor uses interaction techniques to reveal detailed information hidden by overview abstraction. Information about the genes, chromosomes and sequences represented in a block can be accessed in tooltip window. Brushing across blocks reveals genes conserved across different genomes—mousing over a particular set of genes highlights blocks containing orthologous genes. This also highlights the path between the target genome and its immediate sibling sequences in the phylogenetic tree, guiding organism-level comparison.

When used in moderation, connection is powerful for communicating point-level trends in data. Blocks containing genes of interest can be physically connected on demand to highlight specific conservation patterns. Similarly, viewers can filter for genes of a specific name or property. Filters reduce the opacity of blocks outside of the filter, preserving the overall context of the data while visually emphasizing genes of interest.

Because genes may occur at different loci within different organisms, conventional zooming techniques do not support low-level exploration—zooming to a single loci across all genomes is likely to be uninformative. Traditional zooming techniques, such as semantic and goal-directed zoom, can hide matching data as the viewer drills down. Sequence Surveyor instead uses an adapted overview+detail zooming technique (Figure 5.5). Mousing over an aggregate block sets it as the zoom detail block. The component genes of the block are broken down in the

zooming window at the top of the screen. Genes are visualized on either side of a reference line based on the strand of the DNA the gene is located on. Interacting with this zoomed block highlights occurrences of the gene in the overview, allowing the viewer to explore high-level patterns for specific genes.

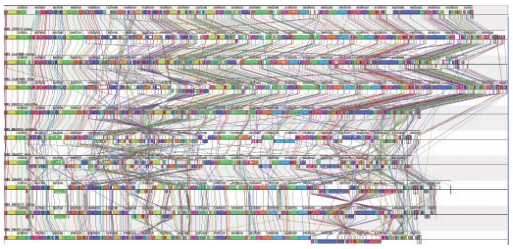
5.1.7 Applications

Parallels to Existing Tools

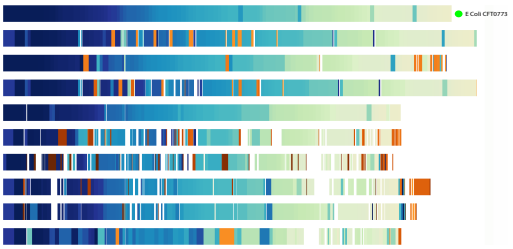
By allowing the user to customize visualization parameters on demand, Sequence Surveyor is able to present views familiar to users of common tools. This provides familiar paradigms for exploring the datasets at scale. Here, we explore the views provided by four popular alignment visualization systems—Mauve [Darling et al., 2004], Mizbee [Meyer et al., 2009], the Broad Institute Medea package [Jen et al., 2009], and the UCSC Genome Browser [Kent et al., 2002]—and show how the information from these views can be displayed at scale in Sequence Surveyor. These applications demonstrate how color can facilitate scalability where other encoding methods fall short.

Mauve: The Mauve viewer displays alignments using a parallel ribbon design: genomes map to rows and orthology is encoded by connection. Despite the scalability issues discussed in §3.2, Mauve is effective for observing matchings between genes to see the patterns of conservation and rearrangement. By mapping gene position to start position and encoding matching genes with similar colors (for example grouped frequency or position in reference), Sequence Surveyor can convey similar information at much larger scale. For instance, inversions creating crossing formations in Mauve are reflected in Sequence Surveyor as inverted color ramps. While, in practice, crossing patterns are often much more salient than color for small inversions, detail-on-demand links can be used to supplement the color-based encoding. Additionally, Sequence Surveyor's flexibility in coloring makes it easier to see observations of interest (see Figure 5.6).

Mizbee: Mizbee's genome view shows conservation between two genomes by examining the conservation between particular chromosomes in a source genome and orthologous genes in a destination genome. Color maps to the destination chromosome that the conserved region is found in and conservation is further indicated by orthology ribbonning. Per-chromosome conservation information can be seen in Sequence Surveyor by filtering by orthology to the chromosome of



(a) Mauve [Darling et al., 2004]. Reference genome *E. coli CFT073* forms the top row.



(b) Sequence Surveyor coloring by position in *E. coli CFT073* (green circle, top row) and ordered by start position.

Figure 5.6: Ten *E. coli* and *Shigella* genomes visualized by (a) Mauve and (b) Sequence Surveyor. The vertical genome order is the same in both cases. The conservation trends represented by orthology lines in Mauve become large color fields in Sequence Surveyor. Inversions appear as reversals in the color ramp. Regions not conserved appear as warm-colored blocks pre-attentively popping out of the visualization.

interest and using interaction to explore more detailed conservation relationships (Figure 5.7). Mapping color to position in the destination genome reinforces the synteny coloring employed by Mizbee. While Mizbee's mapping uses categorical coloring to support point-level pattern finding, the continuous coloring introduced by Sequence Surveyor may be better summarized in aggregate patterns than in Mizbee—ensemble statistics are more meaningful as color distances are semantically mapped.

Medea: The Broad Institute's Medea suite provides five different visualization perspectives for viewing sequence alignment data for closely-related viruses: the Circular Genome Viewer, Stack Map, ChromoMap, Dot Plot, and Viral Viewer. Because these viral genomes are small and tend to have only point mutations, the Broad tools focus on reference-based displays: there are no issues of non-locality as matching regions are co-located in the data set. Sequence Surveyor can support similar explorations to the Medea suite by encoding data using position in reference.

UCSC Genome Browser: While the focus of the UCSC genome browser has traditionally been on exploring individual genomes, there is also limited support for visualizing multiple sequences simultaneously. This approach selects a reference genome and places all other genomes in parallel tracks beneath the reference. A box in a track represents a subsequence that is conserved in the reference. Conserved regions are ordered according to their position in the reference genome.

Figure 5.7: Mizbee provides multiscale insight into gene conservation, but focuses heavily on chromosome-level analyses. In Sequence Surveyor, an analyst can filter on a chromosome to explore chromosome-level patterns: blocks that do not share genes with the target chromosome are reduced in opacity. Coloring according to a reference using event striping helps highlight conservation, for example, across 34 fungal genomes.

This conservation data can be explored in Sequence Surveyor by sorting the genes according to their position in the desired reference. Any elements conserved from the reference will line up beneath their corresponding positions in the reference genome (see Figure 5.8). Color provides an additional channel for additional analysis in context.

Unconventional Mappings

Sequence Surveyor supports exploration using less conventional mappings to provide insight into different properties of the data. Novel position mappings leverage summary processing to cluster genes more effectively than either color or connection. Most existing tools do not explore gene position orderings besides sorting by start position. While this mechanism is useful for viewing data when gaps are relevant, it increases the number of objects on the screen, thereby increasing cognitive load for search tasks. Alternatively, gene index sorting orders genes according to their local position in the genome, removing extraneous gaps in the data and dedicating more space to genes.

While many tools support coloring data according to a reference genome, small regions not conserved in the reference can easily be obscured. Mapping gene

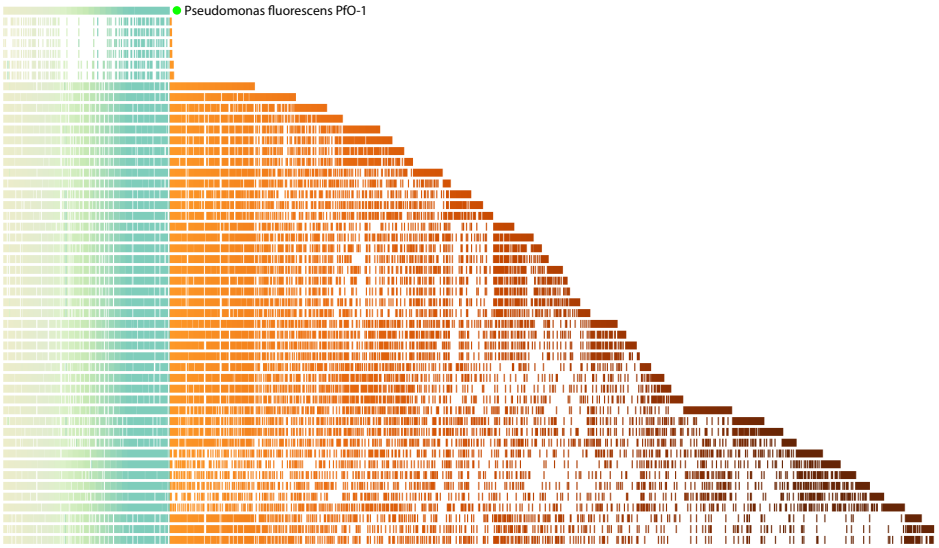


Figure 5.8: Genes from 50 bacterial genomes are sorted and colored according to their position in *Pseudomonas fluorescens PfO-1* to support analyses comparable to the UCSC Genome Browser (green circle). Genes not conserved in the reference are sorted according to their order in the remaining genomes (computed from the topmost genome downwards).

position to a reference segregates ortholog groups according to their conservation in the reference, preserving small unconserved regions. This mapping also reveals the degree of homology between the source and other genomes: the smaller the reference genome becomes, the fewer ortholog groups it shares with the remaining genomes in the set. Similarly, sorting by grouped frequency visually clusters data according to the sets of genomes each ortholog group is contained in. This provides insight into co-occurring genes. If paired with a start position or gene index coloring, these position mappings can display information about the organization of conserved regions in the data such as large-scale inversions and rearrangements.

Raw gene frequency is not commonly visualized in existing tools despite its intuitive meaning. However, coloring by gene frequency can reveal significant duplication patterns in the dataset, potentially signalling significant genes or bugs in the underlying data. This coloring also visualizes many-to-many correspondences between the instances of a group in different genomes.

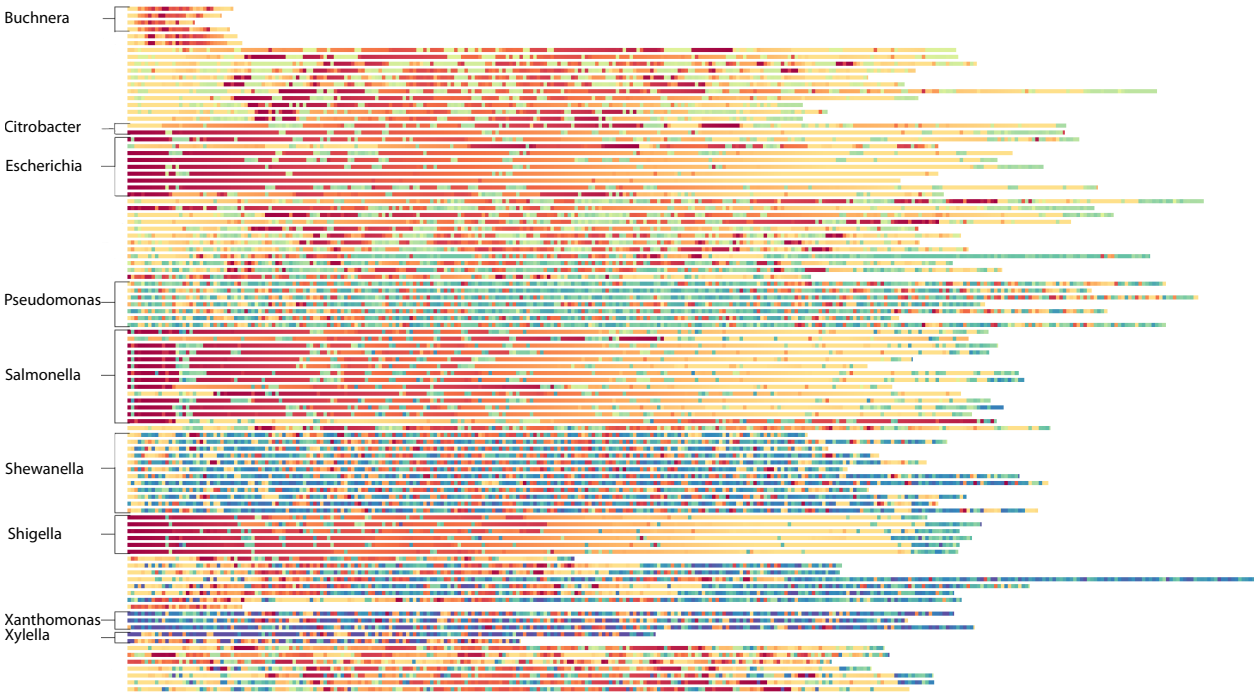
Use Cases

Four groups of domain scientists—evolutionary biologists, a systems biologist, a yeast biologist, and a bioinformatician—used Sequence Surveyor for sequence

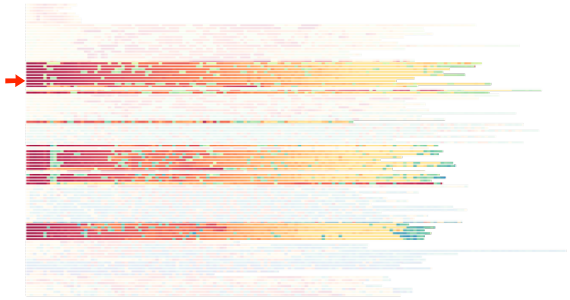
comparison. All four groups had large genome alignment data that they want to explore, but no analysis tools to support that exploration. They evaluated the tool using three bacterial datasets (100 genomes with up to 6,037 genes per sequence (cf. Figures 5.9, 5.11), a subset of the 100 bacteria dataset with 50 bacteria and up to 5,765 genes per sequence (cf. Figures 5.8), and 14 plant pathogens with up to 4,507 genes per sequence (cf. Figures 5.3, 5.10)), one Mauve alignment of ten *E. coli* genomes (cf. Figure 5.6), and one draft multi-chromosomal fungal dataset with up to 17,349 genes per genome (cf. Figure 5.7). My collaborator prepared the datasets from the domain scientists, including computing the alignments (the large alignments took 10 days of CPU time).

Users appreciated Sequence Surveyor as an examination tool useful for discovering and describing aggregate trends in data. They were immediately struck by the scale of the visualizations, not just in terms of size, but also in terms of diversity. Most were pleasantly surprised as they made observations comparing organisms thought to be unrelated. For example, filtering allowed them to quickly identify interesting genes and view the conservation of those genes even in unrelated sequences. The 100 bacteria dataset aligns genomes from a variety of organisms, like *Yersinia pestis* (Black Plague), *E. coli*, *Salmonella*, and *Xylella fastidiosa* (a plant-bourne pathogen). The organization of the data according to evolutionary families proved to play an important role in comparing this diverse dataset. Coloring by position in a reference organism from a given family highlighted high levels of conservation between related families, visualized as continuous gradients (Figure 5.9). Closely related families generally conserve the reference color ramp, whereas less related families introduce new, more divergent color patterns. Furthermore, it allowed the biologists to identify *Citrobacter* genomes by eye from their conservation patterns and place these genomes near the related *Escherichia* genomes to better facilitate comparison. More global conservation patterns can be seen using grouped frequency sorting (cf. Figure 5.11a).

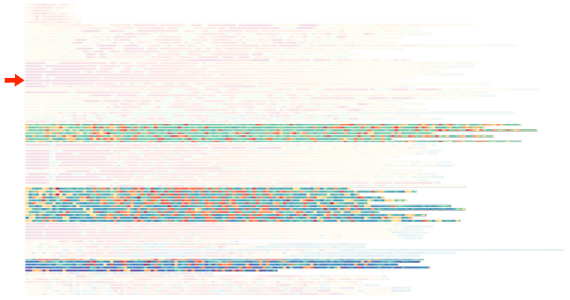
Sequence Surveyor allowed the scientists to quickly identify the set of genes that were conserved across the entire dataset, also known as the “ancestral core”, formed by the leftmost columns of genomes when mapping position to grouped frequency ordering. With respect to systems biology, the ancestral core is often composed primarily of essential metabolic genes. Being able to quickly identify these metabolic genes through this ancestral core can help highlight locality patterns between metabolic genes of interest from specific metabolic pathways. From an evolutionary standpoint, these core genes can reveal interesting functional



(a) Color gradients across genomes reveal evolutionary relationships.

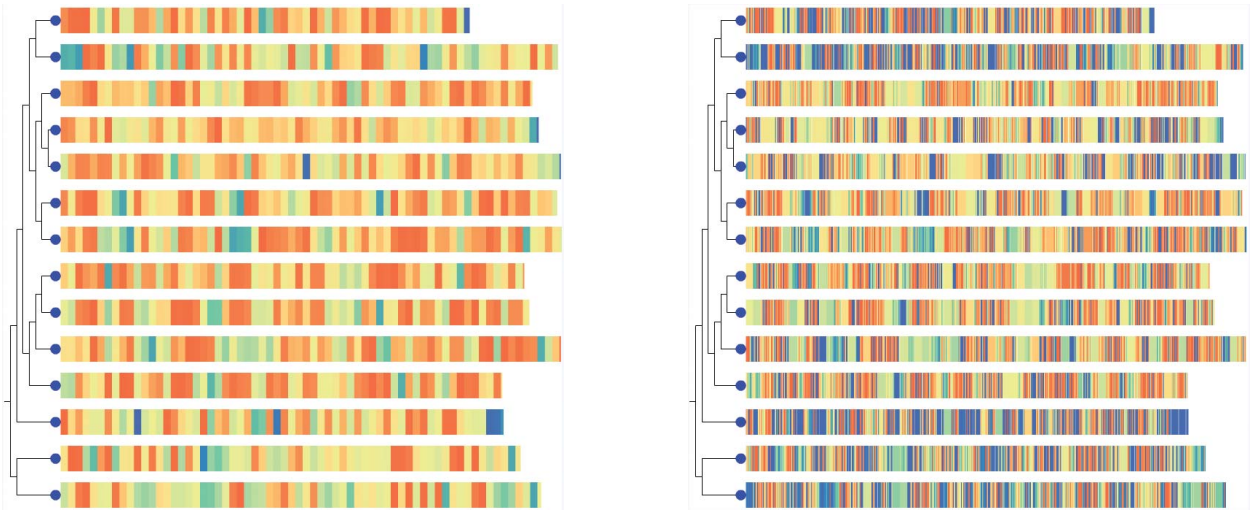


(b) Genomes related to *E. Coli* preserve a red-to-yellow gradient.



(c) Genomes largely unrelated to the reference are largely green and blue.

Figure 5.9: Genome order can help reveal patterns between families of genomes. (a) Sorting one hundred bacterial genomes by index and coloring by position in an *E. coli* organism highlights the high conservation between (b) *Escherichia*, *Shigella*, *Salmonella*, and *Buchnera* genomes through warm colored bands and lack of conservation between (c) *E. coli* and the *Pseudomonas* and *Shewanella* genomes.



(a) Robust averaging shows that genes are well conserved overall (warm-colored blocks).

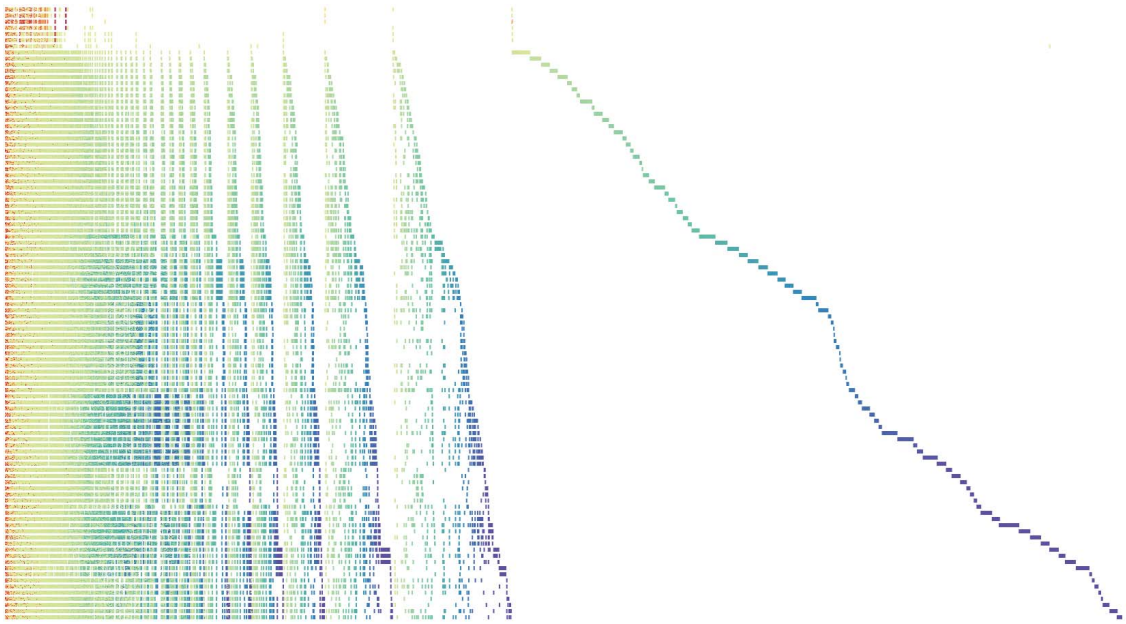
(b) Event striping highlights the outliers, exposing the distribution of more unique genes (blue).

Figure 5.10: Fourteen bacteria colored by membership frequency shows the conservation of genes and their spatial organization.

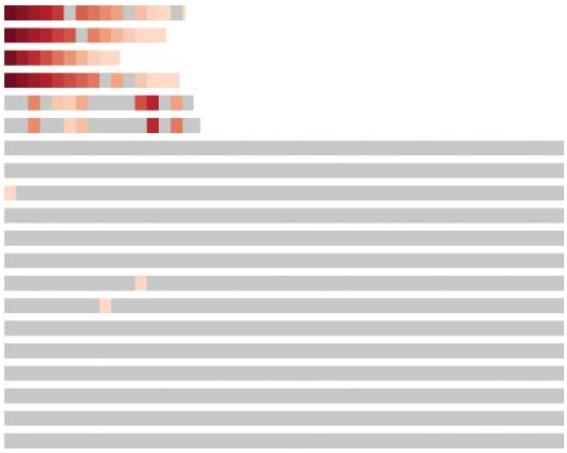
properties of different genomic regions. The *Buchnera* genomes are drawn from insect parasites whose genomes have been pared down to an essential set of genes necessary for survival. By adjusting the parameters in Sequence Surveyor, this observation becomes readily apparent as nearly all component genes of these genomes appear as part of this ancestral core. The biologist can even gain insight into the loci at which these genes are conserved within other families of bacteria. The ability to manipulate the representation of the comparison of *Buchnera* genomes and the rest of the dataset is communicated very visually in Sequence Surveyor (Figure 5.11).

My collaborators found Sequence Surveyor's ability to address different questions valuable. While exploring the data, position mappings like grouped frequency allowed them to quickly address questions that we had not previously considered, such as what set of genes is conserved only in a specific subset of the genomes. Also, they commented on how the tool's ability to blend location and conservation data in a flexible setting would allow them to quickly identify the location of interesting clusters of genes and how tuning aggregation settings could support the exploration of unique features in their data (Figure 5.10).

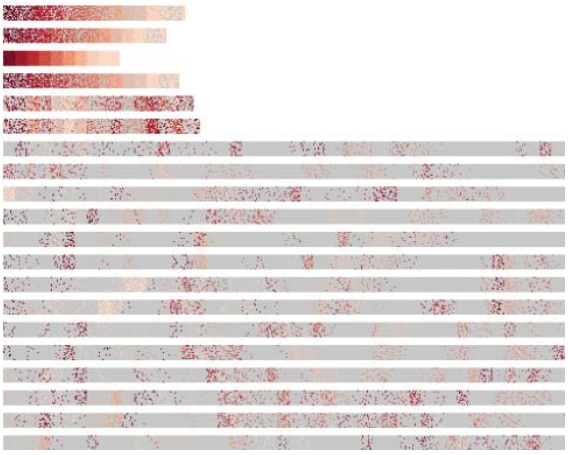
Visualizations of large-scale patterns in data are also valuable for discovering bugs in datasets and alignment algorithms. As an example of Sequence Surveyor's value as a debugging tool, with it my collaborators were able to identify a number of



(a) Grouped frequency position clusters the genes conserved in all organisms (the "ancestral core") in the leftmost columns.



(b) Averaging shows that few regions are dominated by the genes in the ancestral core



(c) Color weaving reveals the prolific distribution of the genes in the ancestral core.

Figure 5.11: A visualization of one hundred bacterial genomes helps to identify a candidate set of genes required for bacterial function. The top six genomes, *Buchnera* insect parasite genomes, are concentrated in this cluster, reinforced by position in reference coloring (red). The clustering of these genomes to the left of the display highlights genes necessary for bacterial function (the ancestral core), whereas the genes to the near right are likely to provide the organism with specialized function. The genes in the ancestral core, while significant, do not account for most of the variation in the data—sorting the *Buchnera* genes by their natural position, coloring them red and (c, d) using different aggregate representations reveals how these genes are distributed in the dataset.

problems with a draft alignment of 37 fungal genomes used during our testing and evaluation. First, the visualization quickly revealed that this dataset contained a putative ortholog group of over 60,000 genes. This group popped out easily due to an extreme skew in the histogram and again by using gene frequency coloring. Upon a more detailed exploration of the genes in this group, the extreme duplication revealed itself to be a bug in the orthology assignments and was removed from the alignment. A second major issue revealed by the tool was that a number of genomes did not contain many genes that had orthologs in other species. This discovery prompted a manual inspection of parts of the alignment, which ultimately led to the identification of some inconsistencies in the labeling of genes by the alignment code. Without Sequence Surveyor, it is likely that it would have taken a lot more time for these problems to be discovered.

5.2 Generalizing Sequence Analysis to Text Analytics

I hypothesize that the design of Sequence Surveyor can be applied any analysis problem comparing data with a total ordering and similarity mapping (or "orthology") between datapoints. Obvious extensions include visualizing amino acid and nucleotide-level alignments. However, an unconventional application of these techniques is to text analysis. Texts, in essence, form sequences of words. A scholar can analyze linguistic patterns and literary structures by visually aggregating information across different texts.

In this section, I introduce TextDNA, a system for scalable text analysis built with collaborators in English. TextDNA uses the same principle design components as Sequence Surveyor to support analysis a different domain. This work exemplifies the generalizability of color for visual aggregation in a domain that is likely more familiar. This section will focus primarily on three novel insights that my collaborators were able to discover using this system. The images in this section are from the original prototype system; however, a web-based version that includes hardware acceleration for one-dimensional aggregation has recently been deployed (`http://vep.cs.wisc.edu/TextDNA/app/templates/list.html`).

5.2.1 N-Gram Analysis

This section will explore two different types of text data: ranked n-grams and raw text. N-grams count how often strings of *n* words occur in a text. The Google N-Grams dataset (Michel et al. [2011]) provides these numbers (plus other metadata such as publication date) for Google Books. Most methods for exploring this kind of data—including the visualizations that accompany this dataset—visualize data for a handful of words over time, focusing on patterns for specific words rather than specific texts or time periods (e.g. Krstajic et al. [2011]). However, understanding n-gram data in such a large corpus (over 5 million texts) provides aggregate insight into the evolution of written language.

This section presents two findings drawn from a dataset focusing on word usage over time. The data contains the 1,000 most popular words (1-grams) per decade in the Google N-Grams dataset between 1660 and 2010. To parallel Sequence Surveyor, words are treated as genes, ranks as positions, and decades as genomes.

The first example demonstrates how scholars used TextDNA to identify a significant typography shift within the dataset. The second example shows how the ability to visually combine information between sequences can help understand linguistic shifts over time. The third shows how TextDNA can also be used to tie findings from visual aggregation tasks to specific, point-level examples.

Looking into the Long S

Figure 5.12 visualizes the 1,000 most popular words per decade between 1660 and 2010. Decades are ordered chronologically from top to bottom, with each decade represented as a single row. Words are represented as colored blocks within each row. Their ordering in each row is reflective of their popularity: the most popular word is on the left and the 1,000th most popular word is on the far right. Words are colored according to the decades that they co-occur in (i.e. their grouped frequency, see Section 4.3), with red words occurring in the 1,000 most popular in each decade of the dataset and blues occurring in the top 1,000 most popular in an increasingly small subset of the decades. Blocks of words encoded using event striping to highlight unusual words within the data. My collaborators were able to use this exploration to find an important class of words, referred to as 'Long S words' that were misprocessed in the original dataset. In this section, I will outline how they were able to identify and isolate erroneous results.

In this view, the leftmost side of the display is almost entirely pale red, indicating

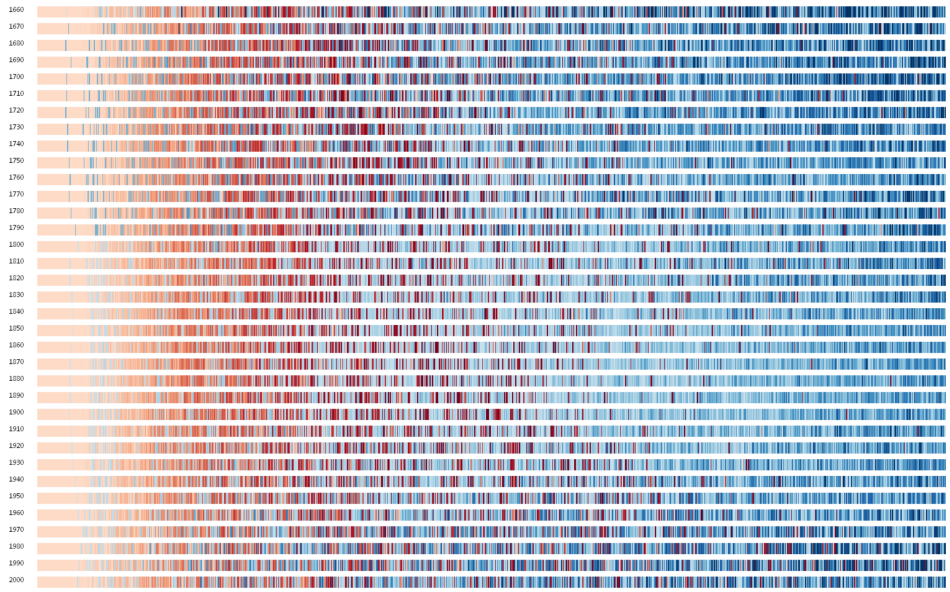


Figure 5.12: TextDNA visualizing the top 1,000 words per decade between 1660 and the modern decade. Aggregating the data using event striping reveals several uncommon words (blue) appear frequently (to the left of the display) in texts between 1660 and 1800.

that the top 100 or so words in each decade all are popular in every other decade. The relative uniformity among the first 100 words across all decades suggests that, in general, the top 100 words and their order are relatively constant across all 35 decades in the dataset. However, several outliers also pop out in this region as blue stripes at the left of the display. These blue blocks indicate words that are extremely popular, but are only popular in a specific subset of decades.

Zooming in on one of these blocks reveals that these words, like 'fo' and 'alfo,' are only popular in a small number of decades. Figure 5.13 filters words that occur roughly as frequently as these throughout the dataset. Blocks that contain words with this frequency stay opaque, those that do not are made partially transparent. This filtering highlights an interesting pattern: there is a large cluster of red blocks in the upper left of the main visualization that abruptly stops at 1800. These words exhibit the same patternas the outlier words in Figure 5.12. No similar patterns appear after 1800.

This swing suggests that there is a significant change in writing around this time. While less popular words only appearing in the top 1,000 words in 40% of the decades is not unusual—words "die" (fall out of popular usage) reasonably frequently—highly common words tend to be those central to written English, such as 'so', 'the', and 'and.' The cluster of words contains a large number of examples

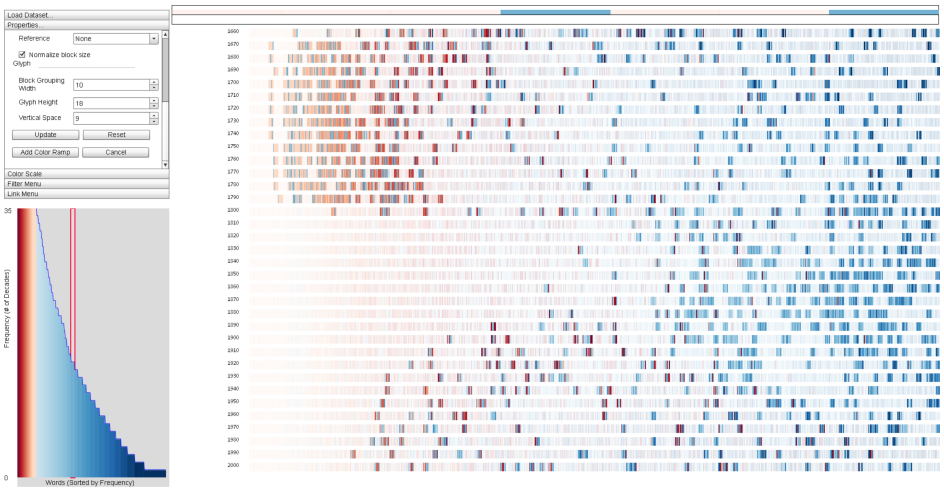


Figure 5.13: Filtering words that appear in the 1,000 most popular in 13 to 15 decades. Words with this frequency pattern are opaque while other blocks are made transparent. The opaque words form two clusters—one cluster of popular words (upper left) before 1800 and a second of less popular terms (on the right) after 1800. The crisp boundaries of the upper left cluster suggests that something interesting might have happened in 1800 that dramatically influenced word popularity.

of a typography convention: the 'Long S' convention. Until 1800, typography sometimes used an elongate 's' character in place of a traditional 's'. To the OCR used to process the Google N-Grams data, this 's' is interpreted as an 'f', creating words like 'fo' and 'faid' from 'so' and 'said.'

Clustering these words together can help disentangle Long S words from standard words that simply fell out of popularity. Ordering words by grouped frequency clusters words by the subsets of decades in which they are popular. Words popular in all decades are placed in the leftmost columns, and the words in the subsequent columns are found in an increasingly smaller subset. Regions of white within a decade indicate columns of words absent from that decade (e.g. the rightmost columns are words unique to a single decade).

Coloring by popularity ('gene index') maps the most popular words in a decade to a dark red and the least popular to a dark blue. Words that are extremely popular, but only in a subset of decades form red columns to the right of the display. This allows us to closely identify major timeframes in which the Long S typography convention was used (Fig. 5.14). While the leftmost red columns are expected (they represent terms that are both popular and common to all decades), there is also a column of words that are reasonably popular in all but one decade from 1660 to 1800 and all but one from 1670 to 1800. These columns also appear

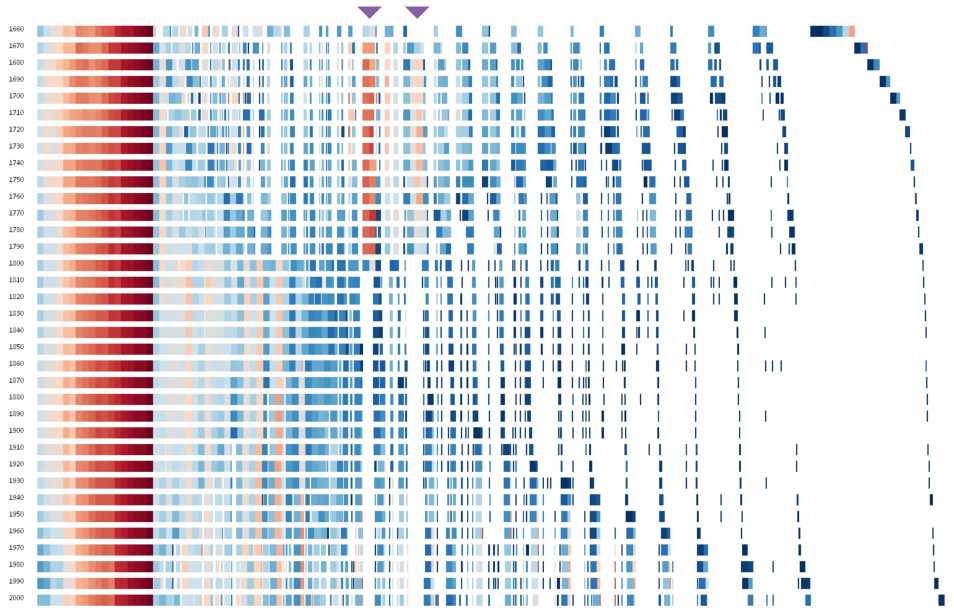


Figure 5.14: Clustering words according to the decades in which they are popular and coloring according to popularity (red words are more popular than blue) clusters together candidate Long S words. Two columns (indicated by the purple triangle) show that there are a large number of words that are popular between 1660 or 1670 and 1800 but do not occur in any subsequent decades. The analyst can then zoom in on these regions to better understand how prolific this typography convention is in the dataset.

to be predominantly composed of Long S terms. My collaborators have used this data to develop heuristics for correcting for the Long S in their datasets.

Historical Patterns in Modern Words

Popular words change over time as a function of culture, historical events, and a number of other factors. By comparing the popularity of words in past decades to that of more modern decades, scholars can assess, for example, how quickly written language is changing and what historical events significantly impact modern writing.

One way to explore these patterns in TextDNA is by setting a decade of interest as a reference. Figure 5.15 uses the most recent decade in the dataset (2000-2010) as a reference. Words in purple are among the 1,000 most popular in that decade, whereas orange words are not. Words are ordered according to their popularity within each decade (most popular on the left) and color weaving aggregation is used to emphasize aggregate patterns while still preserving regions with interesting variation.

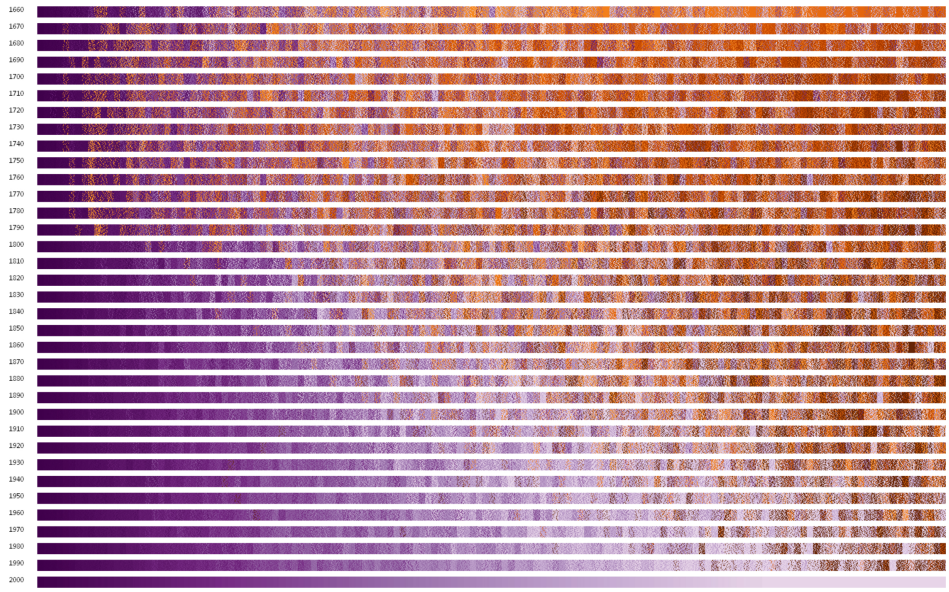


Figure 5.15: Each row represents one decade, with 1660 at the top and the 2000s at the bottom. Most popular words within each decade are on the left, least popular are on the right. Words in purple are popular in the decade of the 2000s, orange words are not. The shape of the orange and purple clusters that form across decades reveal at a high-level how written language has evolved over time.

Visually aggregating data across color reveals an upper triangular pattern across decades: an orange triangle in the upper left (less popular words) and a purple triangle in the lower right. By visually aggregating the data to form this pattern, viewers can estimate the approximate boundary between the orange and purple triangles. This boundary provides an approximation of how quickly words come into and fall out of popularity. As you look back in time, decades have increasingly fewer popular words in common with the most recent decade. Weaving reveals some variation (i.e. oranges in fields of purple and purple in fields of orange) in this pattern that provides interesting areas for further exploration.

An angled band of light purple cuts across the center of the more recent decades in the display. This suggests that several words that are less popular in the 2000s may be words whose popularity has been steadily decreasing since the mid-19th century. It also suggests that a significant portion of the terms that were popular between 1660 and 1790 are not popularly used today, inclusive of the Long S words from the previous section.

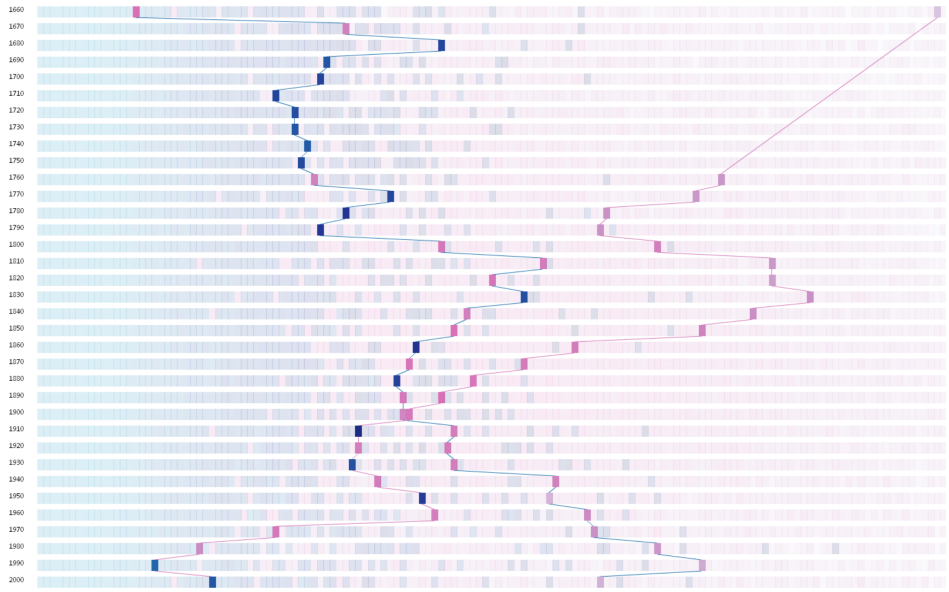


Figure 5.16: The relative popularity of words can provide interesting examples that support high-level insights. For example, the shift between 'woman' (pink line) and 'wife' (blue line) demonstrate how a historical event correlates with a shift in written language. 'Woman' increases dramatically in popularity after the Seneca Falls Convention (1840s), becoming more popular than 'wife' in the decade where women steadily earn the right to vote in the US (1910s).

Anchoring High-Level Patterns in Specific Examples

While understanding aggregate patterns in word usage overtime supports a number of analyses, scholars need to anchor these findings in specific examplars. For example, the previous section illustrates how language generally changes overtime. By anchoring this analysis in specific examples, scholars can start to understand how key historical events may drive this change.

In Figure 5.15, among the least popular terms in 1660 is a single dark purple word. This word represents a term that was extremely popular from 2000 to 2010 but barely in the top 1,000 most popular terms in 1660.

Zooming into this block shows that the purple term is the word 'women'. Further interaction reveals that the term was popular in 1660, but did not again enter the top 1,000 most popular until 1760. Interactively linking instances of this block (adding lines between words) connects instances of the word in the display.

Scholars can contextualize the increased usage of 'woman' overtime by comparing its popularity overtime to a similar term—'wife.' The word 'wife' is extremely popular in 1660 and is in among the 1,000 most popular words in every decade of

the dataset. However, its lighter purple color suggests that it is less popular than the word 'women' in modern writing. The analyst can again interactively connected instances of 'wife' using a line. The path of both connected terms reveals specific information about the popularity shift of 'wife' and 'women' in relation to other terms.

An analyst can isolate these words based on their relative frequencies—'wife' appears in all decades, but woman does not—by coloring using grouped frequency to distinguish between instances of 'woman' (pink) and 'wife' (blue). They can filter on these terms to make their pattern more salient (Figure 5.16). Filtering reveals a notable feature of this data: 'women' becomes more popular than 'wife' starting around 1910, contemporaneous with women earning the right to vote in America. Prior to this crossing, 'woman' rapidly increases in popularity starting in 1840 (the decade of the Seneca Falls Convention) and this increase continues to the modern day.

The example demonstrates how TextDNA can be used to anchor aggregate insights with point-level examples—popular written words have steadily shifted in popularity overtime (§5.2.1), and part of that shift may be attributable to historical events.

5.2.2 Viewing Story Structure

Raw text data also forms a sequence—texts present words in a fixed order. TextDNA can be used to analyze linguistic patterns within a text. In this example, I present one finding my collarborators uncovered in the novel *She: A History of Adventure*. The text was first filtered to remove stopwords, then grouped into its constituent chapters.

My collaborators hypothesized that the novel had substantial linguistic shifts as the novel progressed, but no method for exploring this hypothesis. By treating the order in which a word appears in a text as its 'index,' my collaborators were able to use TextDNA to better understand this hypothesis.

Figure 5.17 depicts this text in natural reading order with each sequence representing a chapter. Words are colored with respect to their order in the chapter containing the climax of the novel. While the coloring (and therefore the primary word usage) appears generally consistent thoughout the text, there are two substantial structures containing relatively unique words. The first, in the second chapter, is a large block of blue. This represents flashback events

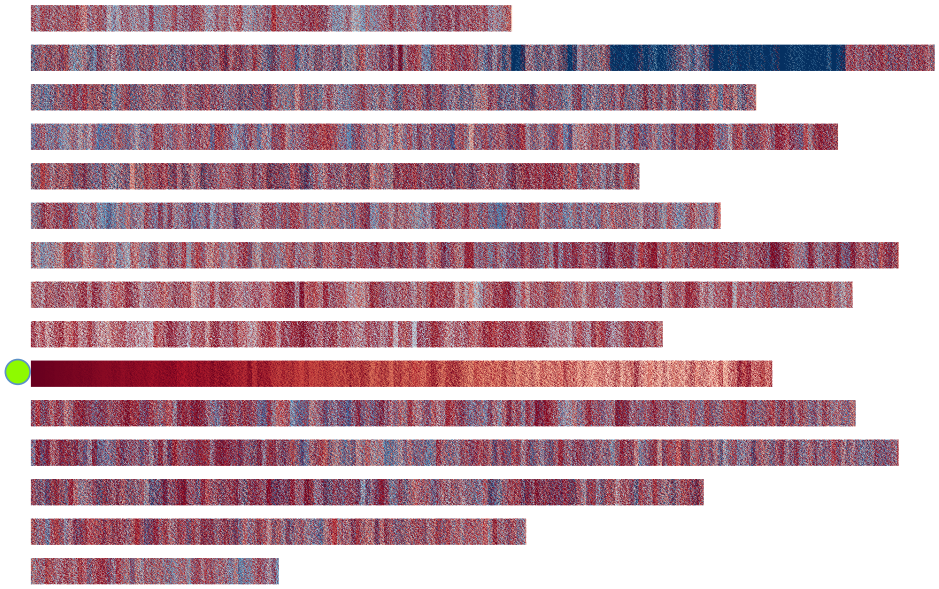


Figure 5.17: The raw text of *She: A History of Adventure* visualized using TextDNA. Chapters are represented as rows, with words ordered according to their natural reading order in the text and colored according to their position in the chapter containing the climax of the story (green). This coloring reveals two areas with unique wording: the blue area at the end of the second row and the yellow area at the end of the reference. These structures correspond to the text where the main plot is first established (blue) and resolved (yellow).

that establish the main plot of the text. The second, the yellow at the end of the reference sequence, identifies the passage containing the primary climax of the novel. These two chapter both have dramatically different linguistic structure than the remainder of the text, providing some evidence in support of my collaborator's hypothesis.

These two structure allow an analyst to "see" the novel's structure at a glance. Using TextDNA on raw text can highlight passages of potential interest, allowing the analyst to aggregate the words of a text into a single image.

5.2.3 Discussion

Sequence Surveyor and TextDNA combine a perceptually-motivated colorfield display, flexible mappings, selectable aggregation strategies, and interaction techniques to provide overview visualization of multiple whole-genome alignments and similar data types. The initial feedback from domain collaborators in both biology and the humanities suggests that they are excited to have a tool capable of allowing exploration at this scale.

This design does not yet address all aspects of analysis in these domains. For example, it provides no mechanism for displaying other data such as the certainty of a mapping or annotations of the genes in biology, or the source passage or part-of-speech labels in text analysis. Multiple selection, grouping, and conjunctive filtering are all mechanisms that could enhance the interaction techniques to widen the kinds of questions that can be explored easily.

Scaling to larger datasets poses new challenges. Handling longer sequences will require better zoom mechanisms—the current mechanism is bound by screenspace. Handling more sequences will require development of "vertical" aggregation strategies to group sequences as well as interaction techniques for looking at detailed comparisons across sets. Experience working with experts in both domains will suggest a wider variety of questions that may require new view organizations to present data. My collaborators are excited about the potential of overview tools for presenting their data when they publish their findings. When and how to leverage visual aggregation for expository applications where the target tasks are known may differ from the exploratory applications these tools were designed for. Exploring this distinction is important future work.

These systems do, however, present examples of how visually aggregating large collections of information matter for real-world analysis problems, and how designs based on color can facilitate these tasks at larger scales than previous solutions. The examples discussed here represent a sampling of the aggregate insights that my collaborators have derived using these systems.

5.3 Scaling Up Molecular Visualization

Sequence Surveyor and TextDNA demonstrate the utility of color for aggregate visual analysis for one type of data—one-dimensional sequences. However, data can take a number of different forms. In this section, I explore how the ideas presented so far generalize to a broader variety of data types. Specifically, I introduce a system for analyzing corpra of three-dimensional data from structural biology that leverages the ideas and techniques discussed in Chapters 3 and 4.

The core challenge of structural biology is to understand how the form of a molecule connects to its function. This is often accomplished by developing computational models that predict locations on the surfaces of molecules where, for example, one molecule will bind with another. These models are validated by comparing their results with experimentally-derived ground truth. Inspecting

these results on a single molecule is challenging as the data is bound to an irregularly-shaped 3D surface.

My collaborators primary strategies for exploring validation results are to either rely on statistical methods, which reduce performance to a single, decontextualized number, or to visualize the data for one molecule at a time, and compare performance across independent instances of the visualization program tiled across the monitor. Detailed examination of the results of an experiment involving dozens of molecules is prohibitive.

With collaborators in computer science and molecular biology, I developed a system for exploring the results of classification validation experiments in structural biology. The challenge is to provide an overview of the results of an entire validation experiment with many molecules, allowing the viewer to identify locations of interest, while retaining facilities for examining the specific details of interesting sites. This approach allows biologists to explore algorithm performance across a corpora of three-dimensional surfaces using a small-multiples view designed to allow a viewer to see aggregate properties of individual molecules as well as to identify details of interest that lead to these properties. While this discussion will focus on the overview components, the overview is connected to a detail view that provides specialized navigation controls over the 3D structures, allowing regions of interest to be examined rapidly. See [Sarikaya et al., 2014] for more information on the detail features of the system.

The overview approach outlined here is based on the idea that an overview can be designed specifically for understanding aggregate properties over multiple scales. Using 3D views of molecules for the overview is impractical, as they require more space, more time to navigate each surface, and do not afford quick summarization. Instead, I build on the work presented in Chapters 3 and 4 to design two-dimensional representations of three dimensional classifier data that allow the viewer to quickly assess results across an entire set of molecules. Visual aggregation tasks for this application occur at multiple scales—my collaborators need to understand performance over a single molecule, a set of molecules, or the entire corpus. As in Sequence Surveyor, this system provides interactive reordering and aggregate representation to facilitate different kinds of visual aggregation task.

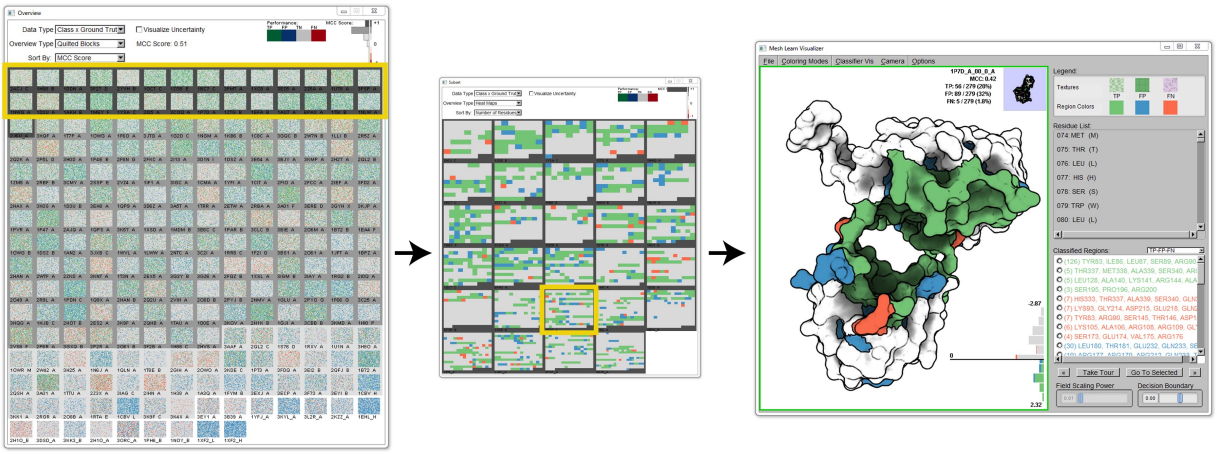


Figure 5.18: Visualization of a validation experiment for a DNA-binding surface classifier. The corpus overview (left) is configured to display each molecule as a quilted glyph and orders these glyphs by classifier performance to show how performance varies over the molecules. Selected molecules (left, yellow box) are visualized as heatmaps in a subset view (middle) and ordered by molecule size to help localize the positions of errors relative to correct answers. The detail view (right) shows a selected molecule to confirm that most errors (blue, red) are close to the correctly found binding site (green).

Biological Background

Bioinformatics classifier experiments are common: for example, a recent survey [Irsoy et al., 2012] notes several hundred papers per year, in just three bioinformatics journals, involve presenting classifier validation results. The survey notes that most of these papers report only simple statistics, at best providing statistical confidence tests.

Better tools for exploring the results of these experiments could improve predictive model development and application. For example, identifying specific molecules or classes of molecules where a classifier performs well may help in understanding the generality of the predictive model. Identifying false positives may help in refining an algorithm. Patterns of false negatives may suggest alternative mechanisms not represented or captured in the model training process.

The results of classifier validation experiments measure both predictive performance and confidence in a prediction. A classifier makes a prediction (positive or negative) marked by its correctness (true or false) for each location on a molecule.

This work demonstrates how the ideas surveyed in Chapter 3 can support scalable aggregate analysis beyond one-dimensional data and introduces new aggregate representations driven by the needs of this domain problem. While

this approach is demonstrated in a specific application for examining molecules, I believe that the contributions generalize to similar domains.

5.3.1 Task-Driven Overviews of Classifier Data

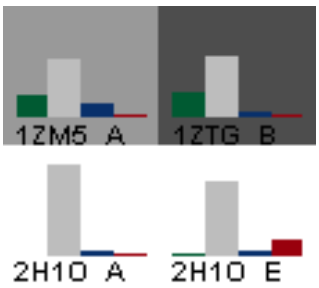
Experimental results for binary classifiers generate a large number of classification decisions, each of which has one of four outcomes (true positive (TP), false positive (FP), true negative (TN) and false negative (FN)), that form the binary confusion matrix [Stehman, 1997]. While the data is simple, it grows quickly: experiments generally are run over dozens of molecules, and there are tens to hundreds of decisions for each molecule.

My goal is to provide an overview of the collection of decisions and their corresponding experimental results. The overview should both show overall performance and help identify the specific molecules, and even parts of molecules, for which the classifier performs well or not. For instance, it should allow the viewer to assess whether performance is uniform across all molecules or variable, to identify groups of molecules that perform similarly, to identify performance outliers or anomalies, or to see aggregate patterns of performance between molecules. These assessments can occur at different scales within the data. For example, an anomaly might be a particular molecule whose performance skews results, or a family of molecules skewed by concentrated groups of false negatives.

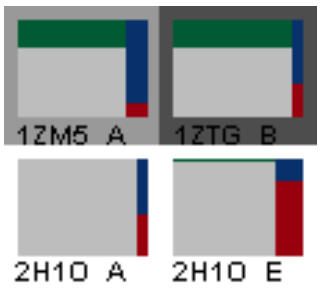
My approach uses two main ideas to support these requirements. First, it emphasizes flexibility, allowing the viewer to reconfigure the display to suit their task. It allows for interactive reordering to generate meaningful performance clusters and interactively specifying different sets of molecules for exploration. As in Sequence Surveyor, the system supports interactively switching between different glyph designs that support rapid visual aggregation and processing of different properties of the data. These design allow the viewer to see both the aggregate properties of the data and low-level details that form these aggregates.

5.3.2 Reorderable Small-Multiples Design

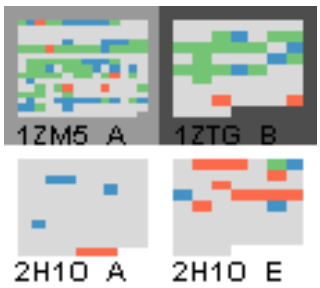
The overview visualizes classifier results from corpra of molecules using a small-multiples display, where each molecule is shown as a small glyph in a grid. Glyphs abstract data from the molecular surface into a two-dimensional representation. As three-dimensional representations complicate visual aggregation for reasons



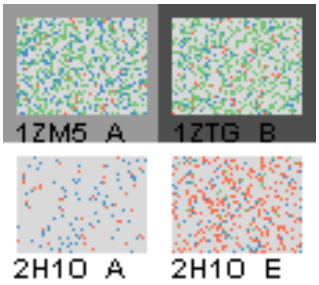
(a) Histograms support proportional comparison.



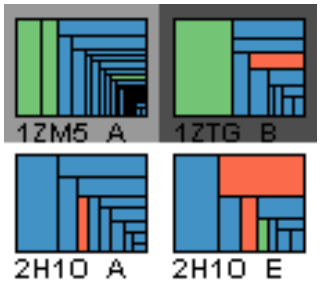
(b) Confusion treemaps show weakly represented classes.



(c) Heatmaps show proportions and structure.



(d) Quilted blocks show summary statistics preattentively.



(e) Cluster plots show spatial cluster sizes.

Figure 5.19: Five overview glyphs support different summaries of performance for classifier performance data.

discussed in the next section, data is instead summarized using a space-filling two-dimensional glyph.

Different designs for the glyphs are provided (described below), but they share features that allow for preattentive summarization. Each glyph relies heavily on color encodings in order to support summarization and pop-out, and the regular ordering of the glyphs helps support visual search. Each glyph has a gray border whose lightness gives an indication of the overall performance (MCC score, with darker borders representing a higher value).

The small multiples can be reordered to explore different types of questions. For instance, ordering by performance (e.g. accuracy or MCC) places molecules with similar performance together and allows for rapidly identifying strong and weak performers. Ordering by molecule name facilitates finding a specific item of interest. Ordering by metadata (properties of each molecule) emphasizes correlations between that property and performance. Coupling the different orderings with different glyph designs provides a wide range of configurations to support

various questions. For example, sorting by the size of the molecule and choosing an appropriate glyph type can not only show whether large molecules perform better or worse than others, but can also indicate whether the errors form large groups on the molecules.

The overview provides some basic interaction features that directly support common tasks. Selecting a glyph can open the molecule in the detail view for closer examination. Sets of glyphs can be selected and opened in a new overview window, allowing for more localized analysis of subsets of the dataset. The user can annotate the glyphs in order to track which molecules have already been examined or should be explored in greater detail.

5.3.3 Glyph Design

Glyphs abstract classifier performance data from the surface to a two-dimensional representation. Visualizing large corpra of data limit the screenspace available to visualize each molecule within the corpra—making complex three-dimensional shapes difficult to see. Additionally, because at least half of the molecule is occluded, some form of navigation or surface unfolding would be required to evaluate performance over the entire surface. The highly irregular shapes of molecules, with their significant pockets and protrusions, make meaningful flattening difficult.

Instead, I leverage nonspatial two-dimensional views that sacrifice information about spatial arrangement in order to remedy problems inherent in three-dimensional views. Further, these views can be designed to facilitate rapid visual comparisons both within an element and between multiple elements. This system leverages color as the dominant channel to encode classification decisions to support rapid visual processing at scale, mapping TP to green, FP to blue, FN to red, and TN to gray.

This color mapping leverages salience to support classifier analysis tasks by considering *a priori* characteristics of the data and task—TN are extremely common in the data and are mapped to gray to decrease their saliency, while FN represent highly undesirable classifications that generally require attention and are mapped to red to aid pop-out. TPs map to green based on conventions familiar to my collaborators. At an overview level, mapping data to a categorical, rather than continuous, coloring not only follows established visualization practice [Brewer et al., 2003b], but also may improve performance on some visual aggregation tasks—the visual system can more efficiently select datapoints that are more

discriminable [Duncan and Humphreys, 1989].

This system allows the user to switch between different glyph designs in order to configure the display to their task. Each design supports certain kinds of visual queries.

Histograms (Figure 5.19a) are a standard encoding and are useful for showing the precise performance distribution within a specific molecule. However, they become harder to interpret when a single class dominates, and do not necessarily afford efficient visual aggregation as well as color.

Confusion Matrix Treemaps (Figure 5.19b) sacrifice some of the inter-class fidelity of histograms, but better show weakly represented classes and make better use of space to afford preattentive size judgements between elements. A vertical divider delineates the proportion of correct classifications (left side), and incorrect classifications (right), providing a quick indication of the predictive accuracy.

Heatmaps (Figure 5.19c) encode the data from each decision using small patches visualized in sequence order, similar to a colorfield (§4.3). Because the size of the patches in a glyph is inversely proportional to the number of decisions in the corresponding molecule, this display gives a sense of the molecule's size. Averaging (§4) and proportion estimation [Correll et al., 2013] are supported by the color encoded design. As residue sequence order is related to spatial proximity, this view can also provide some insight into how the various points are grouped along the surface.

Quilted Blocks (Figure 5.19d) are essentially two-dimensional color woven blocks (§Designing Aggregate Visual Encodings)—the image is similar to the heatmap glyph permuted at the pixel level. This representation exchanges structural fidelity to make preattentive summary statistics, such as mean and variance, easier to access (Chapter Task-Driven Aggregation for Sequence Data) and to help highlight performance patterns at the molecular level. Further, the visual system's ability to readily average quilted glyphs may also help blocks of unusual performance pop-out.

Cluster Plots (Figure 5.19e) use a squarified treemap representation [Bruls et al., 2000] to indicate groups of similar classes that are spatially clustered on the surface. While the glyph does not convey the positions of the groups, it does convey their number and size.

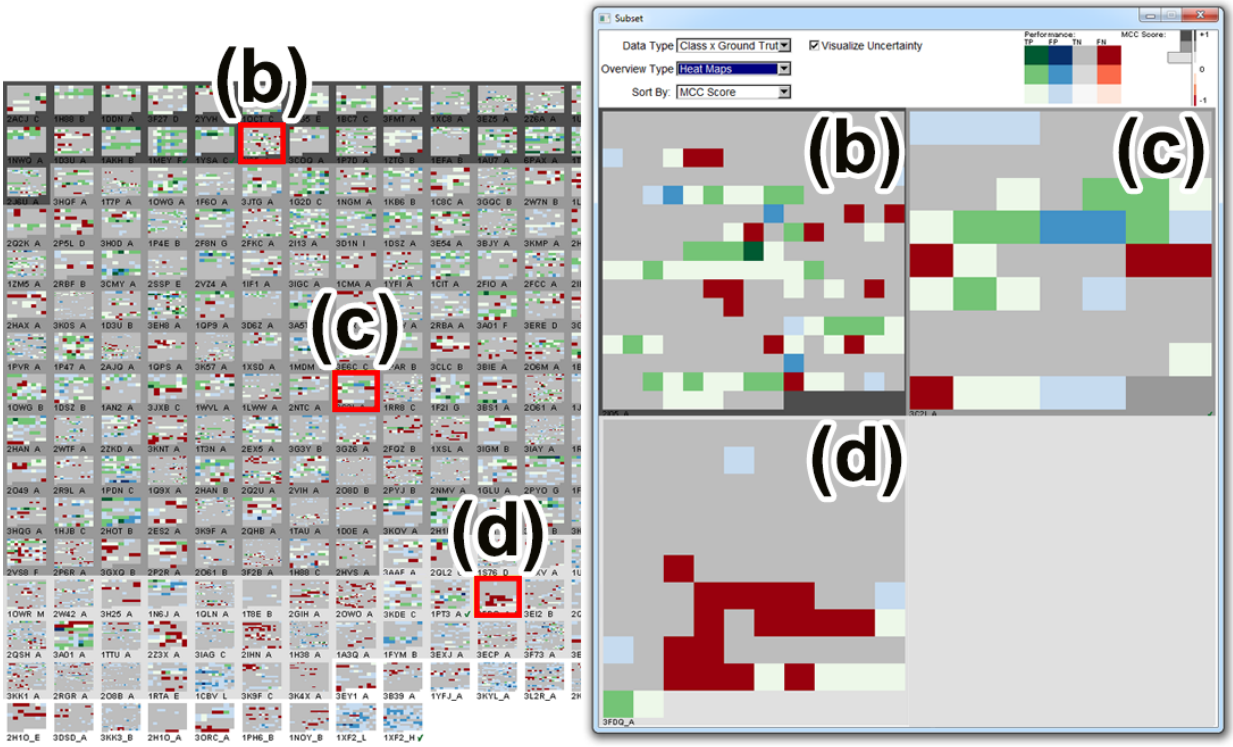
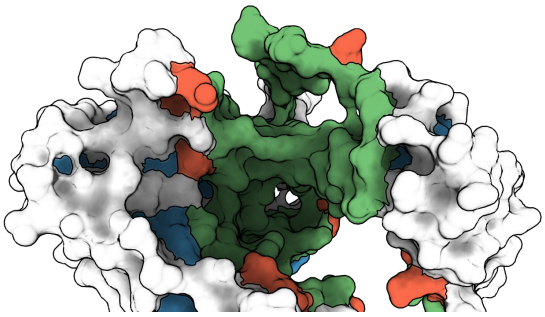


Figure 5.20: An overview of DNA-binding classifier performance for 216 molecules. The overview window (left) displays the corpus rendered as heatmaps (§5.3.3), giving an idea of aggregate performance across the corpus. Glyphs are sorted by statistical performance (MCC score), with top rows corresponding to high performing molecules (dark grey borders) and bottom corresponding to poorly performing molecules (white borders). At all levels of performance, the classifier generally fails with high confidence for false negatives (red) and low confidence false positives (pale blue) as shown in the subset image on the right. The heatmap allows high confidence false negatives to readily pop-out.

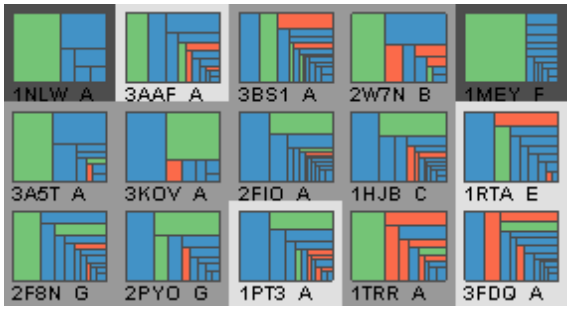
The overview can visualize either raw binary decisions (positive or negative) or supplement these decisions with the respective confidence of each decision. The viewer can optionally show the algorithm's confidence in each prediction in the heatmap and quilted displays. When visualizing confidence data, each of the four colors is replaced by a three-step sequential color ramp in the same hue drawn from Colorbrewer [Harrower and Brewer, 2003].

5.3.4 Case Studies

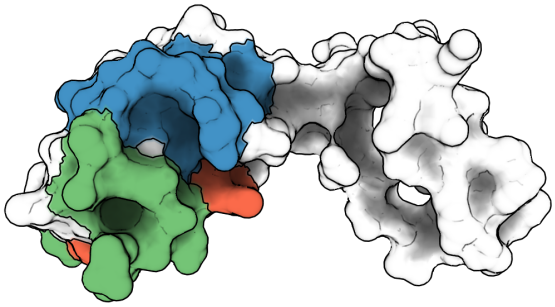
This section outlines two case studies in which this system supported classifier prediction analysis at scale. The datasets consist of a DNA-binding classifier with



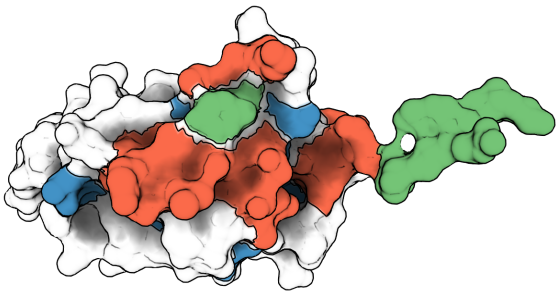
(a) PDB: 2I05_A is well-classified by DBSI. A large pocket (green) holds DNA surrounded by false negatives and false positives.



(b) A region cluster plot shows proteins with similarly sized false positive and true positive regions.



(c) PDB: 2W7N_B shows large region of false positives adjacent to the predicted binding site.



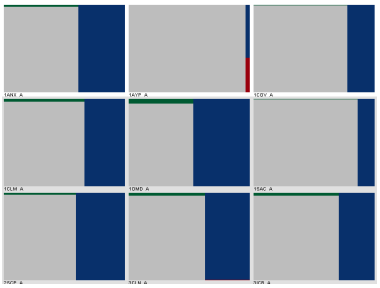
(d) PDB: 3FDQ_A; a large cluster of false negatives forms an irregular binding site shape, consistent with other proteins.

Figure 5.21: Analyzing the spatial clustering of a DNA-binding classifier provides insight into how biochemists could improve prediction performance.

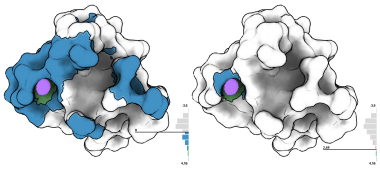
a test corpus of 219 proteins (Figure 5.20) and a calcium-binding classifier with a test corpus of nine proteins (Figure 5.22). Prior to our tool, assessment of results was done by looking at tables of statistics, and by loading surface colors into standard molecular graphics tools.

5.3.5 DNA-Binding Classifier

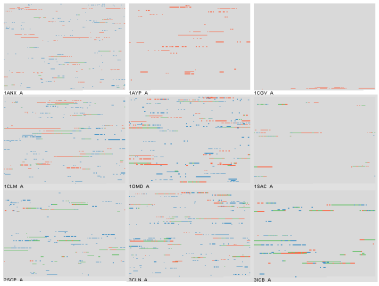
Figures 5.20 and 5.21 show a validation experiment for DNA-Binding Site Identifier (DBSI)—a model that predicts if DNA will bind to different residues on a protein's surface ([Zhu et al., 2013]). Ground truth labels indicate that DNA has been found to bind to the protein structure within five Angstroms of the residue. The model performs well according to traditional summary statistics, including F1 and MCC scores. However, my collaborators wanted to explore predictive performance at scale in order to further refine the classification algorithm.



(a) Confusion treemaps shows the calcium-binding classifier overestimates potential binding sites.



(b) Classification for PDB: 3ICB_A with a decision boundary of 0.0 (left) and 2.61 (right).



(c) A heatmap overview with a decision boundary of 2.61 shows fewer false positives, suggesting the possibility of multiple binding mechanisms.

Figure 5.22: Analysis of a surface descriptor-based, calcium-binding classifier. Modifying the decision boundary indicates that calcium may bind in multiple environments not adequately generalized by the classifier.

Figure 5.20 shows the DBSI test set (219 proteins with 41–932 residues each). Using an overview with quilted blocks ordered by statistical performance (MCC) shows three different types of errors made by the classifier. Molecules with good overall performance have large numbers of true positives with some FN and FP. Molecules with middling performance mostly generate large false positive regions. Poorly performing molecules generally have large numbers of false negatives.

Region cluster plots reveal a consistent pattern for molecules with several false positives. These molecules have large TP regions with many small incorrectly classified regions. Examining these clusters in detail (e.g. Figure 5.21a) shows that the small errors usually surround a correctly identified site. These "near-misses" are unlikely to be meaningful in practice—precise localization is difficult because proteins are dynamic—but suggests that considering spatial grouping may improve classifier performance.

Molecules with large numbers of false negatives also suggested areas of improvement for the classifier. False negatives, when viewed in detail often clustered together to form a long narrow shape (Figure 5.21d). This error points to a potentially interesting biological consideration—instead of the typical binding scenario where the protein envelopes the DNA, the binding sites instead seem to tuck themselves into the grooves of DNA.

These observations show how identifying aggregate molecular performance can help a biologist better understand data across a corpra of molecules. Each would

have been difficult, or impossible, to make with the traditional approach of tables of statistics and manual inspection.

5.3.6 Calcium-Binding Classifier

The second application uses the system to validate a calcium-binding classifier based on surface descriptors [Cipriano et al., 2012], but using a simpler machine learning approach than in the original paper. The validation experiment had 11 proteins. As decisions were made for each vertex in the molecular mesh, each molecule had between 11,000 and 63,000 datapoints.

This classifier performs poorly over the test corpus (MCC: 0.163, Fig. 5.22a). A large number of false positives (blue) are readily apparent in the overview—the classifier overestimates the number of binding sites. Adjusting the decision boundary of the classifier to be more conservative (Fig. 5.22b, right) better captures the true binding sites for some molecules. However, adjusting this boundary for the entire test corpus causes entire binding sites to be missed (red, false negatives, Fig. 5.22c).

Returning to the overview reveals a large number of small binding sites, suggesting that calcium may bind to multiple locations on the molecular surface. However, the variation in performance caused by manipulating the decision boundary implies that a simple model is insufficient to characterize all of the ways that calcium may bind. More complex or ensemble models are necessary to capture this variation.

5.3.7 Discussion

This section introduces an approach for exploring protein surface classifier validation results. The approach couples two-dimensional overviews of three-dimensional data with a detail view for examining this data in context. The overview helps not only to identify aggregate patterns of performance across the corpus, but also helps biologists search for molecules with interesting performance patterns.

The overviews emphasize perceptual support for visual aggregation tasks to both understand performance for a single molecule as well as across multiple molecules. As a result, the overview does not provide any summary glyphs that convey the relative spatial layout of disjoint classifications. For example, none of the current encodings can show that the false classifications occur close to true ones.

To date, the evaluation of this approach has been limited to a few anecdotes and use cases. While specific elements of the design could be evaluated in controlled studies, direct assessment of the overall approach is more challenging. Tests on controlled data sets can allow the confirmation that users can actually identify the kinds of performance patterns our system is designed to expose. However, a better validation of our approach will be its success at helping in the design of more effective classifiers.

Even in our initial use cases, the system has helped reveal insights into the physical groupings of the classifications on protein surfaces. Overviews allowed identifying trends and selecting examples to explore in detail, helping biochemists readily understand aggregate performances in their data and helping molecules with interesting overall performance readily pop-out.

5.4 Discussion

In this section, I outlined three systems that leverage color to support visual aggregation in real-world analyses. Each of these systems substantially increased the scale of data analysis beyond previous methods and demonstrated the utility of their component designs through a series of case studies. These case studies demonstrate how the systems facilitated new insight into complex data.

However, these systems only provide initial verification of the broader focus of this section: that color encodings in data visualization support visual aggregation and increase visualization scalability. The systems provide evidence in support of this claim for real-world examples, but these comparisons are neither empirical nor exhaustive. Most of the case studies discussed in this section were generated while actively working with users to help them explore their data. Further qualitative testing is needed to formally verify the utility of these methods at scale and better understand their overall limitations. They are instead intended to begin a new dialog about the utility of color for visualization beyond single-value tasks and how the concept of visual aggregation can inspire new approaches to visualization.

A better understanding of the wealth of designs that can support visual aggregation tasks and of the kinds of aggregation information users may be interested in would have immense utility for visualization designers as datasets continue to increase in scale. In work outside of this dissertation, I am working with collaborators to explore the space of visual aggregation tasks and understand how these tasks can lead to new questions for researchers in both visualization and

psychology.

Part of this understanding could better information how different designs could support a breadth of visualization tasks. Interactively specifying glyphs within each of the systems discussed here provides flexibility in design, but requires the user to make informed configuration choices. While efforts such as those presented in Chapter 4 provide some guidance for matching these glyphs to task, in my experience, the most common factor in using a glyph is the user's preferences based on visual appeal.

This first portion of this dissertation has established a theoretical basis grounded in perception for using color to support the visual aggregation of information at scale. This basis informed the design a several perceptual glyphs for flexibly aggregating one-dimensional information. I validated the assumptions made in the design of these glyphs through a series of experiments matching visualization designs to visual aggregation tasks. The results of these studies highlights differences between recommendations made in traditional graphical perception studies and the types of encodings that may support visual aggregation. To verify the scalability of these results, I applied these techniques to three real-world applications, resulting in systems that support analysis at scales significantly larger than previous approaches. These systems provide proofs-of-concept that allow designers to realize about designing for visual aggregation at scale, and I hypothesize many other potential designs exist.

Collectively, the theory, methods, experiments, and systems presented in this first part of the dissertation provide substantial evidence for the utility of color for supporting visual aggregation at scale.

Part II

Considering Color in Practice for Point Tasks

6 DESIGNING FOR COLOR IN SURFACE VISUALIZATION

While visual aggregation is critical for designing scalable visualization systems, effective visualizations support data understanding at both at the aggregate and point levels of detail. For example, in the molecular visualization system discussed in the previous chapter (Section 5.3), a biochemist can identify interesting aggregate performance patterns from the aggregate overview, but will often still want to explore performance over specific molecules in detail. In doing so, they can understand the structural contexts behind where an algorithm is underperforming or identify alternative properties that may explain performance.

Using color across both large and small scales can help an analyst remain oriented as they move through data at multiple scales—the information is represented consistently across different levels of detail. Designs using color may support visual aggregation effectively, but color is less effective at small scales, where tasks generally compare individual datapoints [Cleveland et al., 1985]. For example, identifying spots of performance across four categories is easy—the viewers can readily identify blues, greens, and reds. The challenge arises when encoding quantitative data, such as confidence in classifier predictions, where differences between steps in an encoding are more subtle (c.f. Fig. 6.1). There are many potential challenges to using color to encoding quantitative data. For example, the visual system can only detect a limited number of colors [Ware, 2000] and factors of a visualization design, such as background color [Mittelstädt et al., 2014] or mark sizes [Carter and Silverstein, 2010], influence viewers abilities to correctly identify color. This means performance may degrade as analysts make increasingly precise judgments about information encoded using color.

A large part of why color encodings fail is that how designers use color is directly informed by metrics from colorimetry, such as HSV and CIELAB (see Stone [2004] for a survey). HSV focuses on how colors will be displayed rather than perceived. Perceived lightness, which is often critical in color ramp design, and precise relationships between colors are not well predicted by this model. Color difference metrics, like CIELAB, are designed to understand the sensitivity of the human eye rather than how designers might create effective encodings. While these metrics capture perceived differences well, they measure perceived color in isolation under heavily controlled conditions [L'Eclairage, 1978]. They do not consider aspects of color presentation that might degrade perceptions, such as complex artificial surfaces, web-viewing, or variable mark sizes. In this part

of my dissertation, I show how visualization designers can create visualizations that better support point tasks using color by measuring and modeling color perceptions for visualization.

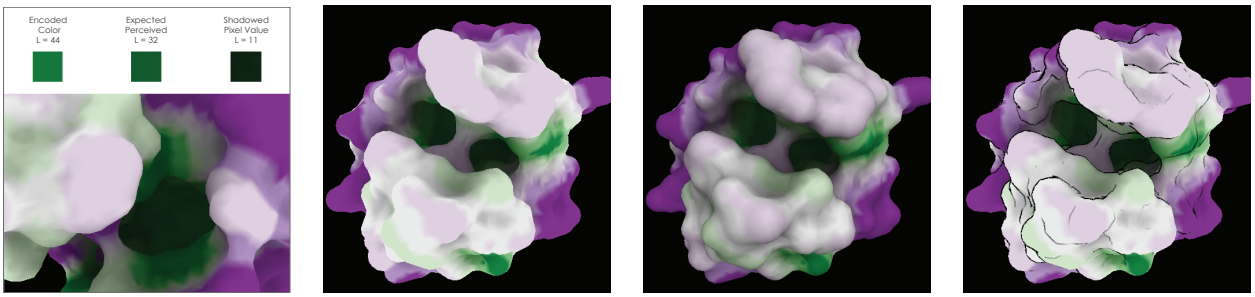
Designing visualizations that use color effectively can be accomplished in one of two ways. First, designers can tune other parameters of a visualization to support accurate color interpretations. Second, designers can construct color encodings that are robust to other constraints of a visualization. In this chapter, I will focus on the first method. I will show how designers can create visualizations that improve accurate color identification using color for molecular surfaces, where color encodings can be complicated by surface shading used to convey shape and structure. In the following two chapters, I will explore the second approach. I will explore how designers can create color encodings that are effective for a given visualization context by designing for an intended use context (web viewing, Chapter 7) and for specified constraints on mark size (Chapter 8). Please note that many of the examples in the remaining chapters use subtle color differences designed to be viewed in digital displays. The figures may not appear correctly in print.

6.1 Overview

Color is an intuitive and commonly used channel for visualizing data directly on three-dimensional surfaces. Color encodings can intuitively represent data within the context of a surface. Visualizing data in context is especially critical for surfaces such as molecules, where functional and structural features provide a meaningful scaffold for understanding charge, binding sites, protein-protein interfaces, and other data.

However, shading models used to render surfaces directly impact color encodings: shadows and shading manipulate color to convey depth, resulting in a conflict between representations of shape and data. Surface features, like pockets and loops, often hold interesting areas for exploration, but tend to be the most deeply shadowed. Misinterpreting color encodings in these regions adversely impacts a visualization's effectiveness, but removing surface shading impairs perceptions of surface depth and shape. By understanding how visualization design impacts how accurately viewers can read colors from shaded regions, designers can create surface visualizations that better support both shape and data comprehension.

Reading color-encoded scalar data from the surface of a molecule requires



(a) Actual and perceived color of the pocket

(b) Ambient Occlusion

(c) Ambient Occlusion + Directional Lighting

(d) Ambient Occlusion + Contours

Figure 6.1: Our findings, exemplified by hydrophobicity data in the shadowed regions above, show that visualization design significantly impacts viewers' abilities to read data encoded on a surface. (a, b) Ambient occlusion surfaces support viewers in reading shadowed data, which is improved by (c) directional shading. Conversely, (d) stylized shape cues may hinder this ability.

matching colors against a legend or key. Because the *image color* of the data on the surface depends on shadows and shading, the *apparent color* of the data may not be the same as the unshaded *key color* in the legend. In this work, I explore how visualization design affects viewers' abilities to match shadowed image colors to the corresponding unshadowed color in a key. In the real world, this task would be enabled by lightness constancy—the ability of the visual system to use various visual cues to disentangle color and shadow. Lightness constancy is well studied in perceptual science, and a number of theories and models exist explaining how the different visual cues contribute to this ability. However, these models focus on explaining constancy in real world or simple synthetic scenes (see [Kingdom, 2011] for examples). They provide little guidance in how the mechanisms for interpreting surface colors may be affected by the stylized or simplified techniques used to render interactive complex surfaces in visualization. Lightness constancy is also sensitive to a variety of visual factors: in studies, simply moving from the real world to a virtual image has significantly impaired constancy [Olkkonen et al., 2009]. Techniques commonly used in visualization, such as ambient occlusion lighting [Landis, 2002], may remove many visual cues that theoretical work indicates are used for lightness constancy, such as lighting direction [Ruppertsberg et al., 2008].

In this work, I derive inspiration from lightness constancy to understand how visualization design can support surface visualizations that effectively leverage color encodings. In a series of experiments, I measure color-matching performance for molecular surface visualizations rendered using ambient occlusion. I

confirm that viewers can read color encodings in shadow with some accuracy for simplified rendering methods often used in visualization, but that how the surface is visualized directly influences the strength of this ability. The visualization techniques used to render a surface can significantly improve or inhibit viewers' ability to correctly interpret shadowed colors. Our results point to a correlation between techniques that enhance depth perceptions and improved performance in interpreting shadowed colors. These results can guide designers in creating surface visualizations that more accurately depict shadowed data. They also illustrate trade-offs for designing surface visualizations using color. Given the complex and unfamiliar structures of molecular surfaces, I anticipate that these results could be applied to visualizing surfaces in other domains. A summary of results is presented in Figure 6.1.

6.1.1 Background

Visualization allows analysts to explore data in the context of a surface by mapping visual representations of data onto a rendering of the surface. The resulting image combines a number of different visual factors to support data analysis. The visual system has several different constancy mechanisms that account for variation in visual factors. Color constancy, for example, allows viewers to resolve colors under different lighting conditions. It has three principal elements [Foster, 2011, Newhall et al., 1958]: lightness constancy, hue constancy, and saturation constancy. All three components can be used to encode data along a surface [Ware, 2000]. Supporting their constancy allows visualization designers to use these channels effectively. In this section, I focus on lightness constancy as it allows the visual system to account for luminance variations underlying the shadows and shading that convey surface structure.

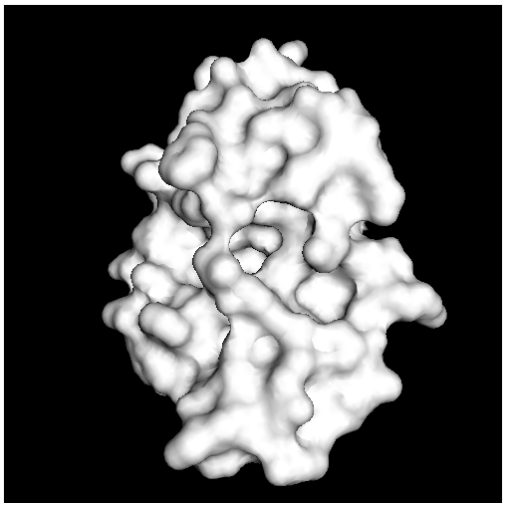
Perceptual psychology has established models to explain how lightness constancy functions account for changes in illumination in the real world [Brainard and Freeman, 1997, Bressan, 2006, Gilchrist et al., 1999, Kingdom and Moulden, 1992, Land et al., 1977, Rudd, 2010]. Existing theories hypothesize that properties such as contrast ratios between light and shadow [Cataliotti and Gilchrist, 1995, Rutherford and Brainard, 2002], shadow intensity [Newhall et al., 1958], lighting intensity [Grossberg and Hong, 2006], lighting direction [Ruppertsberg et al., 2008], object colors and reflectance [Cataliotti and Gilchrist, 1995, Granzier et al., 2009] and spatial cues [Allred and Brainard, 2009, de Almeida et al., 2010,

Hedrich et al., 2009, Olkkonen et al., 2008a] may all contribute to the brain's ability to disentangle an object's color from the lighting used to illuminate it. For example, the visual system may identify a luminance value in a scene, such as the lightest or average luminance value, as an "anchoring point" and adjust all residual colors accordingly [Kingdom, 2011]. The brain may also adapt to lightness differences in smaller spatial regions of a scene and adjust perceptions to maximize these local contrasts, in essence increasing the perceived dynamic range of the scene [Grossberg and Hong, 2006].

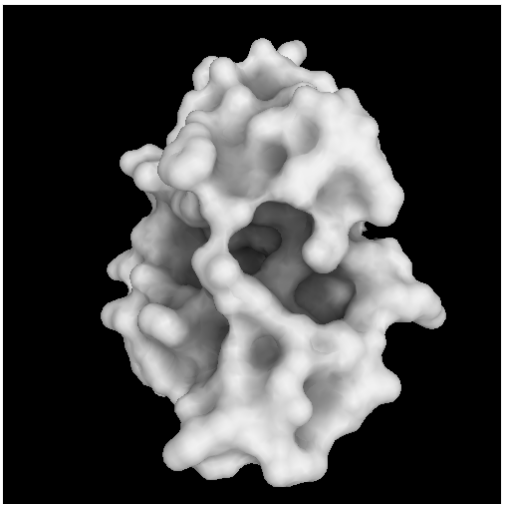
This prior work focuses on perceptual mechanisms, quantifying constancy as a function of low level visual features under highly controlled conditions for both artificial and naturally-occurring scenes. Studies of constancy in digital environments generally use simple stimuli, such as two-dimensional images (e.g. flat square planes or collections of randomly sized and colored rectangles that form "Mondrians") or checkerboards overlaid on simple three-dimensional shapes (e.g. cubes [Adelson, 1993, Agostini and Galmonte, 2002, Logvinenko, 1999] or creased rectangular planes [Adelson and Pentland, 1996]). It is unclear how these findings translate to visualization applications, where complex surface structures are often illuminated using approximated and stylized lighting models (e.g. ambient occlusion).

Surface visualizations commonly use ambient occlusion [Landis, 2002] to approximate global illumination Several properties of this illumination model may inhibit or even remove visual cues that are hypothesized to facilitate lightness constancy. For example, many theories suggest that lightness constancy relies largely on backcomputing color changes in a scene based on overall lighting and reflectance properties [Foster, 2011]. This idea of "estimating the illuminant" depends on the existence of measurable lighting contributions, including direction and relative intensity. However, ambient occlusion synthesizes equal light from all directions—the resulting illumination is directionless and of uniform intensity. This might inhibit lightness constancy and reduce viewers' abilities to interpret shadowed colors.

Most of what we know of constancy is based on identifying grey-scale colors under differening levels of illumination viewed under controlled conditions (see Kingdom [Kingdom, 2008] for a survey of experiments considering color). In surface visualization, color ramps are not grey-scale, but often communicate data values through variation in hue, lightness, and saturation, and viewed under a variety of conditions.



(a) Diffuse Directional Lighting



(b) Diffuse Directional Lighting plus Ambient Occlusion

Figure 6.2: Depth perception of a surface using (a) local illumination can be greatly enhanced by (b) adding ambient occlusion shading, which emphasizes the shape of structural features such as pockets.

The measures generated by perceptual models are not focused on providing feedback for designers; instead they quantifying mechanisms of the visual system, operating over specific visual features. These limitations make it difficult to apply these models to visualization: it is unclear how they inform whether different designs will sufficiently increase the effectiveness of a visualization. I consider the effects of lightness constancy as a measure of visualization effectiveness. I seek to understand how common visualization techniques influence how accurately viewers interpret surface colors in shadow.

6.2 Molecular Visualization

Molecules can be visualized in many different ways: as atomic representations [Corey and Pauling, 1953], stylized moieties [Pauling et al., 1951], or functional surfaces [Lee and Richards, 1971, Richards, 1977]. I focus on solvent-excluded surface models, which are commonly used in conjunction with color encodings to display molecular data in context (see DeLano [2002], Sarikaya et al. [2014] for examples). Data such as charge, binding affinity, and machine learning results are projected across these surfaces to increase the functional and spatial understanding of the dataset. Color is often used to visualize this data in popular systems like VMD [Humphrey et al., 1996], Pymol DeLano [2002], and BioBlender

[Andrei et al., 2012]. While the structural features of the solvent-excluded surface present visually interesting aspects of the surface for such investigations, surface shadows (which use grayscale color to convey depth) may be problematic when using color to encode data. Recent efforts have explored alternative techniques for visualizing heavily shadowed regions of surfaces, such as opacity reduction [Borland, 2011] and volume segmentation [Krone et al., 2011]; however, these methods focus only on deep pockets and reduce the visual quality of the overall surface to emphasize these pockets. Interactive techniques for exploring surfaces are also problematic for shadowed data—viewers may incorrectly interpret values obscured by shadow, making it difficult to accurately identify interesting regions to explore.

Although shape and shadow complicate color encodings, they are important for communicating spatial properties of the surface (Fig. 6.2). Recent research in volume rendering has explored how different shading models impact viewers' depth perceptions in visualization. Although they focus on volume visualization, the studies provide useful general insight into surface perception. For example, Lindemann and Ropinski [2011] evaluated seven lighting models to derive design suggestions for effective depth-based rendering. More recently, Grosset et al. [2013] demonstrated how subtle changes to a depth cue (depth of field) can significantly influence perceptions of a volume. Such research empirically evaluates common design decisions to confirm how different design choices impact perceptions of structural features in a 3D visualization. These experiments do not consider how these choices influence perceptions of color or other visual encodings.

Ambient occlusion is commonly used to convey depth in molecular surface visualizations in both research [Cipriano and Gleicher, 2007, Tarini et al., 2006] and production tools [Andrei et al., 2012, DeLano, 2002]. Ambient occlusion approximates shadows on a surface by assuming a constant light emitted from all directions, measuring the percentage of possible lighting directions visible from a given surface point, and attenuating the surface color at that point accordingly [Landis, 2002]. This provides a pre-computed approximation of shadow that conveys depth comparable to directional lighting models [Langer et al., 2000]. Yet, it often fails to convey subtle shape variations and is therefore often supplemented by other shape and depth cueing techniques such as diffuse illumination [Tarini et al., 2006], contours [Cipriano and Gleicher, 2007, Tarini et al., 2006], and haloing [Tarini et al., 2006] in molecular surface visualization.

I explore lightness constancy for molecular visualizations rendered using ambi-

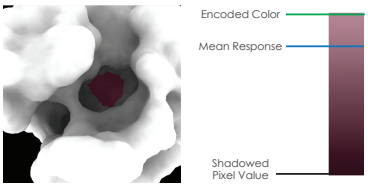
ent occlusion. I focus on how visualization design influences constancy to support accurate performance on a color matching task.While the ensuing studies measure viewer performance on solvent-excluded molecular surfaces, I anticipate that the findings of this study, summarized in Figure 6.3, are extensible to visualizing of more general classes of surfaces.

6.3 Motivation and Overview

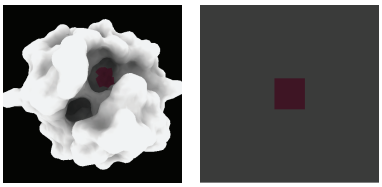
The way data are represented directly influences how accurately viewers interpret visualized data. For example, the rendering methods used to create a volume visualization impact perceptions of surface depth [Lindemann and Ropinski, 2011]. In visualization design, there is often a trade-off between how closely a visualization reflects the real world and how efficiently it can be rendered. We may choose to make this trade-off for many reasons, such as supporting interactivity, rendering on devices with different computational resources, or emphasizing certain properties of an object. By understanding how different visualization design choices influence how accurately visualized data is perceived, designers can begin to systematically reason about these trade-offs to design visualizations that support specific tasks.

Here, I explore how different design techniques for visualizing surface data influence how accurately viewers interpret shadowed data. I focus on visualizations rendered with ambient occlusion as it is commonly used to convey surface depth and shape without the computational overhead of more complex shadow rendering techniques. Ambient occlusion computes the shading values for a surface once, and those values remain constant regardless of the position of the surface. It exchanges many aspects of real world lighting captured by more complex global illumination models (e.g. interreflection in radiosity, lighting direction from cast shadows), which must be recomputed whenever the position of the light changes relative to the surface, for computational tractability. This trade-off can improve performance for interactive visualizations.

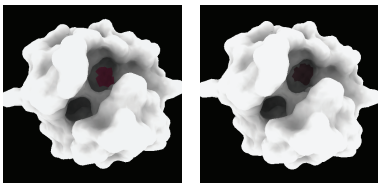
Ambient occlusion supports perceptions of surface depth using shading to simulate shadows. When a data value is encoded as color on an ambient occlusion surface, shading makes the pixel value of the image color on the surface darker than the original encoded color. For shadowed objects in the real world, lightness constancy enables viewers to disentangle colors from shadow. Many properties of more complex global illumination models are known to contribute to these



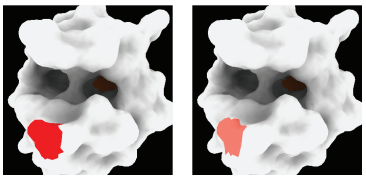
(S1) Shadowed colors on ambient occlusion surfaces generally appear lighter than the corresponding shadowed pixel value.



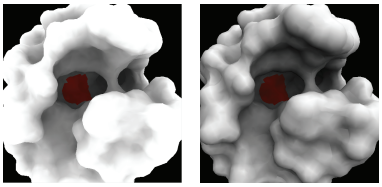
(S2) Simultaneous contrast between shadow and color does not fully explain viewers' abilities to read shadowed colors.



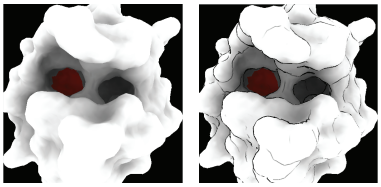
(S3) Shadow rendering is important to accurately interpreting color on a surface: incorrect shadows reduce constancy effects.



(S4) Viewers can distinguish between shadowed colors for common color ramps.



(S5) Directional lighting and stereo improve participants' abilities to interpret shadowed colors.



(S6) Contours reduce how accurately viewers can shadowed colors.

Figure 6.3: We explore how visualization design influences viewers' abilities to accurately read shadowed colors in surface visualization. We first verify that viewers can interpret shadowed colors on ambient occlusion surfaces and that surface shading and structure supports this ability. We then explore how different surface visualization techniques might improve or impair performance. These results can help inform the design of effective surface visualizations.

constancy effects, such as directional lighting, cast shadows, or interreflectance of light along the surface. Given the prevalence of ambient occlusion in surface visualization, I want to understand if this lighting can support accurate color interpretation in shadow and what aspects of visualization design influence these effects for surfaces rendered with ambient occlusion.

The studies presented here represent first steps in this exploration. The goals of this work are to (a) establish that viewers can accurately read shadowed surface colors from ambient occlusion surfaces, and (b) reason about the trade-offs in this ability for common surface visualization designs. These studies address these goals by answering six specific research questions (addressed in studies **S1** through **S6**). Like Anderson and Winawer's approach to analyzing the causes of constancy [Anderson and Winawer, 2008], I consider how different *design layers*—visualization design decisions that influence the presentation of a surface—may influence performance on a color matching task. I begin by verifying that ambient

occlusion surfaces can support lightness constancy effects (**S1**) and that these effects are a function of visualization design rather than contrast between data and shadow (**S2**). This verification suggests that viewers can interpret shadowed data with some accuracy and that the design of a visualized surface may influence this ability.

Once these properties are verified, I then explore how the shadow computation itself influences performance (**S3**). Specifically, I show that some simplifications of shadow models may hinder the interpretation of shadowed colors.

The ability to correctly infer the luminance of shadowed surface colors is imperfect: viewers still make more errors in interpreting unshadowed surface colors than shadowed colors. However, I find that performance in color matching tasks is still higher when luminance cues are included in the color mapping, as in most common practice color scales, than for hue and saturation alone (**S4**).

Surface visualizations often combine other visualization techniques with ambient occlusion to enhance perceptions of the depth and shape of a surface. In order to understand the trade-offs involved in these design decisions, I compare viewers' abilities to interpret shadowed colors when adding directional lighting, stereo cues, and suggestive contours. The results suggest that designs correlated with depth cueing (directional lighting and stereo) allow viewers to more accurately identify shadowed surface colors (**S5**), while stylized contours, which enhance shape percepts at the expense of shadow percepts [Kennedy and Bai, 2000], reduce performance (**S6**).

These studies, discussed in detail in the ensuing sections, collectively suggest that visualization design influences how well viewers' can read surface colors, and that there is a correlation between designs that support perceptions of surface depth and those that effectively communicate shadowed data. A summary of specific results is provided in Table 6.1. This work serves as first steps in mapping a design space for understanding lightness constancy in visualization.

6.4 General Methodology

I evaluated the relationship between color matching performance and surface visualization design through a series of color matching experiments. Each experiment required participants to match a data value encoded as color on a surface to its original color in a provided ramp. All experiments followed the same general procedure. Any variation from this methodology is discussed in

Figure 6.4: We mapped colored patches to three levels of shadow. Colored patches applied to molecular surfaces rendered using ambient occlusion gauged performance for molecular surfaces (top), whereas 2D squares (bottom) measured effects due to contrast with the surrounding shadow.

detail for each experiment. The manipulated design component was treated as a between-participants factor in all but one experiment (§6.7.2).

Participants were first screened for color vision deficiencies using digital renderings of Ishihara plates [Hardy et al., 1945]. Only participants who passed this screening were allowed to proceed, and a post-hoc questionnaire was used to further verify normal color vision. Participants were instructed that they would see a series of images with colored patches placed under different levels of shadow and would be asked to match the original color displayed in the image to a provided color ramp. A pair of example problems were provided to help illustrate the task. Participants were then shown 32 500×500 pixel stimulus images presented in a random order with the corresponding seven-step color ramps immediately to the right of the stimulus (Fig. 6.4). Participants recorded the color in each ramp they felt best corresponded to the shadowed patch by clicking on a color from the ramp and clicking a "Submit" button to move to the next stimulus. To mitigate adaptation effects, participants saw a gray screen for three seconds between respective stimuli (duration was selected through pretesting). Participants were given unlimited time for each response.

6.4.1 Stimulus Generation

Unless otherwise stated, stimuli consisted of static images of solvent-excluded surfaces rendered as a white surface on a black background. Surfaces were shaded using ambient occlusion plus a 10% constant ambient term. Shadows were generated by reducing surface luminance using ambient occlusion computed using the methods described in [Landis, 2002] by attenuating the luminance component of the surface assuming a white light and a gamma of 2.2 [Stokes et al., 1996a].

Surfaces were derived from four different proteins from the Protein Data Bank [Berman et al., 2000] (PDB IDs: 1BBH (bacterial), 1B7V (bacterial, Fig. 6.1, 6.4, 6.9, and 6.3), 1DB4 (human, Fig. 6.2), 3CLN (mammalian, Fig. 6.11) and generated via MSMS [Sanner et al., 1996] with each surface visualized entirely

within the image. Since the experiments were conducted in the browser, all images were prerendered with sRGB embedded color profiles.

A single colored patch was mapped to a unique position on each surface for each shadow level. Patches were of roughly equivalent size on each surface—some variation was caused by the curvature of the surfaces—and never directly bordered the black background. For each experiment, patches were generated for three levels of shadow: light (25% $\pm$ 2% shadow), medium (50% $\pm$ 2% shadow), and dark (75% $\pm$ 2% shadow), with all shadow levels measured after the 10% ambient lighting was applied. Patches were placed in shaded regions where no part of the region was lighter than the assigned shading level and at least 94% of the region was within the assigned shading level.

Each participant saw colors from two seven-step color ramps. To control the number of stimuli viewed by each participant, we selected three colors from each ramp as test colors to be displayed in shadow. Tested colors were selected such that each color was at least one just noticeable difference (JND) apart even in the darkest shadow condition. Throughout this work, we use the JND measure defined using crowdsourced metrics in [Albers Szafir et al., 2014] to help account for anticipated display variability.

Except for in the stereo pilot (§6.7.2), each participant saw 32 stimuli total: six stimuli at each of the three shadow levels (one for each combination of ramp and test color) and 14 stimuli with patches placed in an unshadowed position (one for each level of each ramp) for validation and to prevent biased responses from the reduced set of colors in the shadow conditions. Images were selected randomly from each of the four surface models and the stimuli were presented in a random order to minimize adaptation to a given color or shadow level. The use of validation stimuli with "obvious" correct answers (in this case, the exact pixel match to the surface patch) is commonly used to gauge honest responses in crowdsourced studies, where participants sometimes "click-through" the questions using random answers to complete the study as quickly as possible [Buhrmester et al., 2011]. Participants responding two or more ramp units away from the correct answer on multiple validation stimuli were excluded from our analyses.

6.4.2 Participant Selection

Participants were selected from two separate pools: in-person (20 participants total) and crowdsourced using Amazon's Mechanical Turk (322 participants total).

Mechanical Turk is known to be a generally reliable participant pool for graphical perception studies [Buhrmester et al., 2011, Heer and Bostock, 2010] and also allows us to measure performance for viewers under a spectrum of real-world viewing conditions. This approach may introduce variability in viewing conditions and devices, which prevents us from making precise claims about visual perception. We found comparable effects across both crowdsourced and in-person studies. We hypothesize that this variability may be beneficial for measuring factors significantly influencing performance under realistic conditions, but leave this verification to future work. To ensure the quality of our results, we followed known best practices for ensuring honest responses [Kittur et al., 2008], only recruited participants with at least a 95% overall "approval" rating, and used explicit validation questions. We also tracked both worker identification number and IP address across all experiments to ensure that each participant completed only one experiment.

We recruited 16 in-person participants (10 female, 6 male) between the ages of 21 and 31 ($\mu = 25.75$, $\sigma = 2.47$) to address **S1** and **S2** under controlled conditions. We then recreated these experiments using crowdsourced participants on Amazon's Mechanical Turk. We found consistent results between in-person and crowdsourced participants, confirming that crowdsourcing is a sufficiently reliable method for recruiting participants for our color matching task. We addressed **S1** through **S6** using a cummulative total of 322 crowdsourced participants (174 male, 147 female, 1 declined to report) between the ages of 18 and 65 ($\mu = 31.25$, $\sigma = 9.66$).

Certain visualization conditions are not amenable to crowdsourcing, such as stereo viewing (**S5**), which requires specialized displays. For our stereo experiment, we also required participants to have prior stereo experience due to the nuances of proper stereo viewing. We recruited 4 participants with prior experience with stereo displays for pilot study **S5**. We present preliminary findings from this study, but consider it a pilot as we were only able to recruit a limited number of participants, all with some familiarity with our task, due to our qualification restrictions.

We analyzed our data for each experiment except **S1** at the level of shadow $\times$ color ramp $\times$ primary independent variables using ANCOVAs (Analyses of Covariance) with question order as a covariate to account for interparticipant variation from repeated measures across conditions. In all cases, results from each participant pool were analyzed independently, and an equal number of participants

was considered for each condition within each experiment. If exclusions caused an imbalance between conditions within an experiment, participants were excluded at random until both conditions were balanced. Across all studies, only data from participants who reported normal or corrected-to-normal vision and no color vision deficiencies was considered. We observed no significant performance effects due to age.

6.5 Lightness Constancy for Surfaces

6.5.1 Do we see constancy effects for ambient occlusion surfaces? (S1)

Before exploring color matching performance as a function of design, my first experiment aims to verify that participants can match image colors to a key when image colors are darkened by ambient occlusion shadows. If these visualizations support lightness constancy, I would anticipate that viewers would match image colors closer to the original, unshadowed color than to the darkened pixel color in the image.

Methods

The procedure and stimuli for this experiment are outlined in §6.4. I carefully engineered two luminance-varying ramps such that, for each tested color, both the correct key color and the pixel value of the shadowed image color could be mapped to within one crowdsourced JND of a ramp color. These ramps allowed us to verify that the participants were able to employ lightness constancy in order to disambiguate between the pixel value of the shaded patch and its corresponding ramp value. Ramp luminance was varied in the CIELAB color space from $L^* = 9$ to $L^* = 87$, with each step separated by 13 units and $L^* = 35$, $L^* = 61$, and $L^* = 87$ used as test colors (Fig. 6.4). I centered the ramps around blue and red such that all colors remained within the monitor gamut and consecutive colors were sufficiently different. Each participant saw 16 stimuli from each ramp, resulting in 32 total responses per participant.

Participants were drawn from two pools: 8 in-person participants to measure constancy effects under controlled conditions and 17 crowdsourced participants from Mechanical Turk to measure effects under the diverse array of conditions ex-

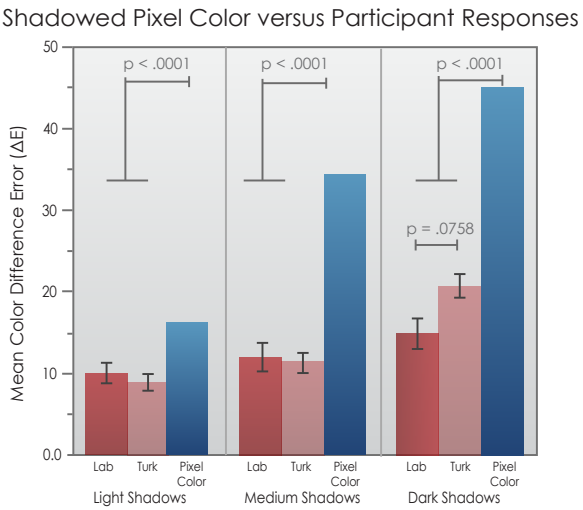


Figure 6.5: Mean difference between the correct patch color and participant responses in **S1**. Both in-lab and crowdsourced participants mapped shadowed colors significantly closer overall to the original key color than to the shadowed pixel value. All error bars encode standard error.

perienced in visualization applications. Two of the crowdsourced participants were excluded from the analysis for poor performance on validation stimuli, resulting in 15 participants total for analysis. Participants completed the in-person study using an Asus G51J Series Laptop with an NVidia GeForce GTX 260M graphics card in full screen using Google Chrome. Room lights were dimmed to control ambient illumination.

Results

Performance was measured as the difference between the correct key color and the color reported by participants. I use this metric rather than absolute correctness because constancy is an approximate phenomena—even in real scenes, constancy mechanisms cannot always exactly compute the correct color [Gilchrist and Annan, 2002]—"right" or "wrong" measures do not adequately capture lightness constancy performance. Figure 6.5 summarizes our results.

A repeated measures Multivariate Analysis of Variance (MANOVA) on each set of participants revealed evidence of significant constancy effects. There was a significant difference between colors reported by participants and the actual shadowed surface colors ($F_{in-person}(1, 45) = 19.7818$, $p_{in-person} < .0001$; $F_{Turk}(1, 93) = 29.9981$, $p_{Turk} < .0001$). Participants mapped shadowed patches to significantly lighter colors than the surface pixel color. This result was consistent across all shadow

levels ($F_{in-person}(2, 45) = 130.4005$, $p_{in-person} < .0001$; $F_{Turk}(2, 93) = 141.2638$, $p_{Turk} < .0001$). I did not find a significant difference between response errors at the three tested shadow levels in the in-person experiment ($F(2, 45) = 1.3754$, $p = .2566$) and between the light and medium shadows in the crowdsourced experiment ($F(1, 45) = 1.4694$, $p = .2264$). This lack of difference also indicates constancy effects—participants mapped these patches to roughly equivalent colors despite significant changes in shadow darkening. Overall, the performance of the crowdsourced participants is consistent with in-person participants, crowdsourced participants performed slightly worse on the darkest shadow conditions.

These results collectively suggest that participants are able to account for the effect of shadows on surface colors. Participants matched surface colors to colors significantly lighter than the pixel value of the shadowed color and darker shadows did not always influence the apparent color (Fig. 6.5). The consistency of these results across both in-person and crowdsourced conditions point to the robustness of this phenomena across viewing conditions and suggests its importance for visualization design. However, the results also suggest there may be room for improvement: there was still significant error in matching surface colors to the original colors, and this error might be improved by different visualization techniques.

6.5.2 Do viewers use structural information to interpret surface colors? (S2)

Results from **S1** demonstrate that participants are correctly identifying shadowed image colors as lighter than than the corresponding pixel color. However, these results do not confirm the cues the visual system uses to interpret these colors. It is possible that the darkness of the shadow surrounding a patch rather than structural information allows participants to interpret image colors. Simultaneous color contrast between a stimulus and the surrounding shadows accounts for some aspects of lightness constancy in the real world [Grossberg and Hong, 2006, Rutherford and Brainard, 2002], although it is insufficient to explain all constancy effects in three-dimensional surfaces [Allred and Brainard, 2009]. The visual system may normalize contrast for local windows around a patch at comparable depth plane [Adelson, 1999, Kingdom, 2011]. For visualizations, contrast between the patch color and the surrounding shadows may cause the local color patch to appear lighter than its actual pixel value—the dark shadow makes the patch

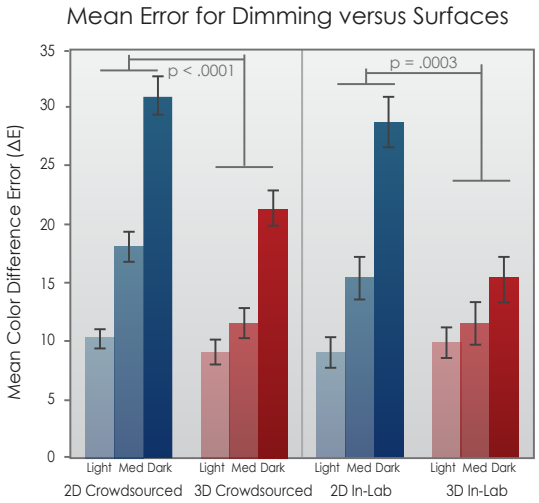


Figure 6.6: Viewers identified colors more accurately on surfaces than on dimmed two dimensional planes (**S2**), suggesting that surface structure plays a role in idenitfying shadowed colors.

appear lighter by contrast. In this experiment, I wanted to verify that contrast does not account for all of the effects reported in Section 6.5.1. If contrast sufficiently explains the **S1** results, how we visualize a surface will not significantly influence perceptions of shadowed colors.

To test if the observed constancy effects were due to simultaneous contrast, I compared color matching on visualized surfaces to two-dimensional "shadowed" patches. If the effects from the surface color matching task (§6.5.1) are due to contrast, I would expect no significant difference between color perception for 2D shadows and 3D surfaces. This would imply that performance depends on the darkness of the surrounding shadow rather than visualization design.

Methods

2D stimuli consisted of 100 pixel-wide colored square patches centered in a 500 pixel-wide white square. Patch size was comparable to the 3D surface patches. Both the patch and background square were dimmed to the tested shadow level to mimic the local lighting on the molecular surface—the white background was dimmed to the grey of the surface shadow to indicate the lighting shift, and the colored square was dimmed to the color of the shadowed patch (see Fig. 6.4, bottom). Participants were instructed that both patch and background were in shadow and that they were to identify the original, unshadowed color of the center patch. In order to simplify task instructions, 2D stimuli were not placed on a black

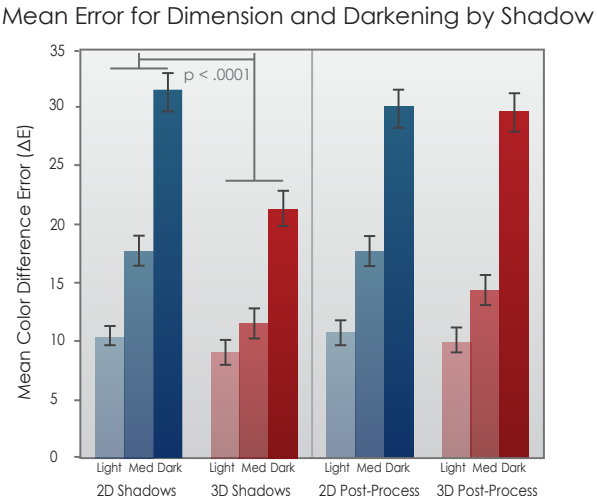


Figure 6.7: No significant improvements were seen between dimmed planes and surfaces darkened using non-gamma corrected image-processing methods. This suggests that constancy mechanisms leverage shadow information when processing surface colors (**S3**) and small changes to those shadows can damage their effects.

background. As colored patches were generally placed far from the background in the 3D condition, we do not anticipate any confounds from this decision: contrast effects in constancy are speculated to operate over local windows within a visual scene [Adelson, 1999].

Colors, shadow levels, general procedure, and stimulus distribution mirrored the 3D condition described in the previous section, including 14 unshadowed 2D validation stimuli (§6.5.1). Data was again collected from two participant pools: 8 in-person participants and 18 crowdsourced participants. Three crowdsourced participants were excluded for poor performance on the validation stimuli, resulting in 15 participants for analysis. Both sets of participants were run simultaneously with those discussed in §6.5.1, with dimension treated as a between participants factor.

Results

I ran a three-way ANCOVA (dimension, color ramp, and shadow level) on the difference between the original color and response color for the 2D data and the 3D data from **S1** for each participant pool. Participants matched colors on surfaces significantly more accurately than on equally darkened 2D patches (Fig. 6.6, $F_{in-person}(2, 262) = 13.3018$, $p_{in-person} = .0003$; $F_{Turk}(1, 532) = 23.9261$, $p_{Turk} <$

.0001). This accuracy varied significantly across shadow level ($F_{in-person}(1, 262) = 19.5687$, $p_{in-person} < .0001$; $F_{Turk}(2, 532) = 74.6342$, $p_{Turk} < .0001$), but not across color ramp ($F_{lab}(1, 262) = .2160$, $p_{lab} = .2160$; $F_{Turk}(1, 532) = 0.1031$, $p_{Turk} = .7483$).

These results suggest that the effects measured in **S1** (§6.5.1) are not entirely explained by simultaneous contrast: surface structure accounts for a significant proportion of the reported color matching performance.

6.5.3 Do correct shadows support constancy effects? (S3)

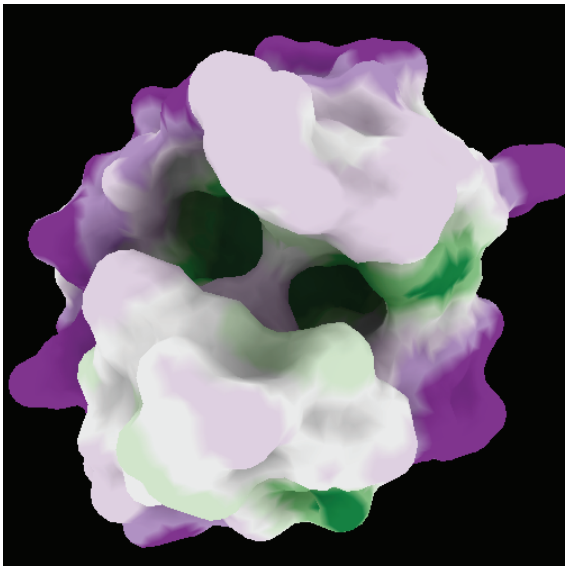
Correct ambient occlusion shadows apply shading by attenuating the amount of light emitted by the display. An approximate implementation might instead apply the darkening as an image post-process. The difference is subtle: because the image processing may occur in device-dependent RGB, the shadows would be differently affected by gamma correction (Fig. 6.8). While the magnitudes of these changes are small, they distort the gradients of the resulting shadows, which may contain important information for constancy [Gilchrist and Jacobsen, 1984].

While modern visualization systems generally apply shadows correctly, the effects of subtle differences in shadow application provide evidence of the connection between perception theory and visualization practice. Distorting these gradients may reduce performance on our color matching task. Given the subtle visual difference between the two conditions, a performance difference would imply that perceptions of shadows and surface structure influence the apparent color of surface data.

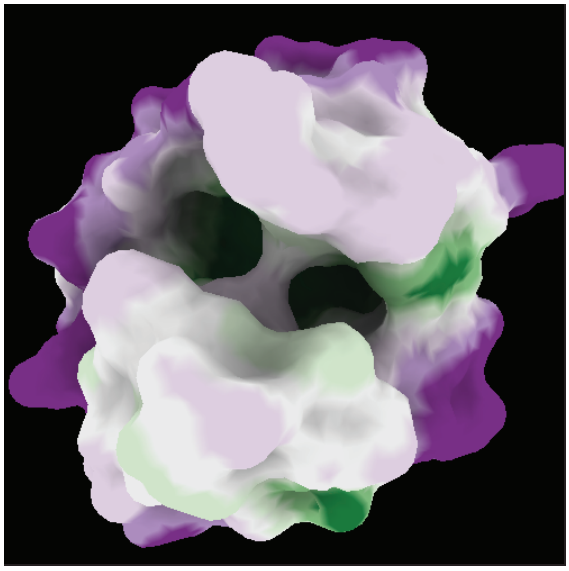
Methods

I replicated the previous experiments (§6.5.1 and §6.5.2) using stimuli that applied ambient occlusion attenuation to each channel of device-dependent RGB color. The luminance of all corresponding linearly and nonlinearly dimmed colors were within one L^* JND measured under crowdsourced conditions [Albers Szafir et al., 2014]. Dimension (2D versus 3D) was treated as a between-participants factor. The procedure otherwise mirrored those described in §6.4.

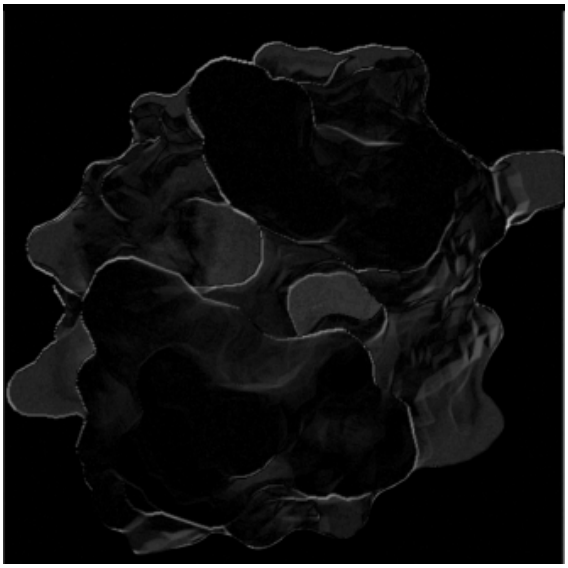
The study was run simultaneously with the crowdsourced studies discussed in Sections 6.5.1 and 6.5.2. I collected data from 34 participants on Mechanical Turk. Three participants were excluded from the 2D condition and one from 3D surfaces condition for performance on validation stimuli, resulting in 15 participants per condition for analysis.



(a) Hydrophobicity data with correct shadows



(b) Hydrophobicity data with incorrect shadows



(c) Color differences (computed as Euclidean difference in CIELAB) encoded as greyscale

Figure 6.8: Differences in CIELABΔE between correct and approximate shadows for the surface visualized in Figure 6.1b. Color difference is encoded using linear greyscale, with black representing areas of no difference. While shadow lightness was within one crowdsourced JND for all tested shadow levels, the incorrect shadow method changed the lightness and color gradients of shadowed colors.

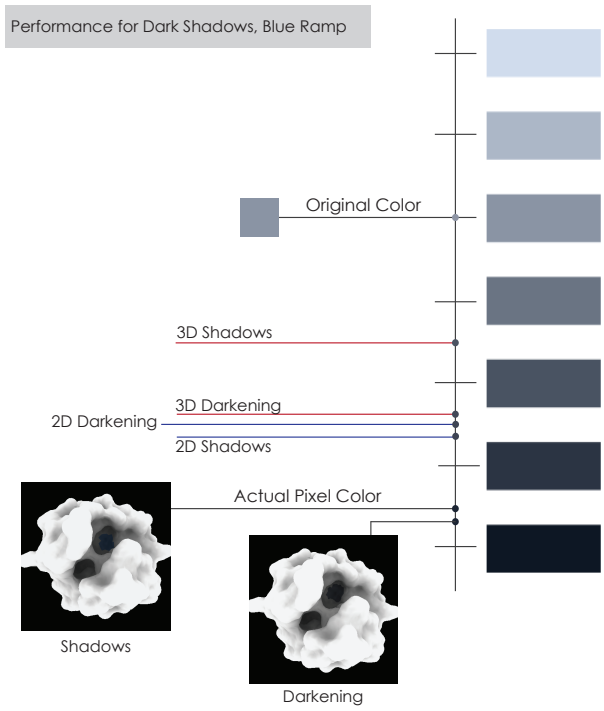


Figure 6.9: Molecular visualizations using standard ambient occlusion demonstrated the best color identification performance. The lack of a significant difference between 2D dimmed patches and 3D surfaces using image-processing darkening suggests that the visual system actively uses shadow information to extract shadowed surface colors.

Results

To address **S3**, I compared participant responses across all four crowdsourced conditions (dimmed 2D patches from Section 6.5.2, 3D surfaces from Section 6.5.1, image-processing darkened 2D patches, and image-processing darkened 3D surfaces). A four-way ANCOVA (dimension, shadow level, darkening type, and color ramp) was used to analyze participant responses. Participants identified colors significantly more accurately on 3D surfaces than 2D planes ($F(1, 1061) = 18.4228$, $p < .0001$). Shadow level significantly influenced performance ($F(2, 1061) = 160.0194$, $p < .0001$), but I found no significant effect of color ramp (($F(1, 1061) = 1.2035$, $p = .2729$)), and only a marginal main effect for darkening type ($F(1, 1061) = 3.8011$, $p = .0515$). I also found a significant interaction effect of dimension and shading ($F(2, 1061) = 5.5637$, $p = .0185$, Fig. 6.7). A Tukey's Test of Honest Significant Difference (HSD) found no significant difference between performance in both 2D conditions and the image-processing darkened surfaces, but revealed that standard shadowed surfaces outperformed all other conditions (at $\alpha = .05$).

These results suggest that precise shadow information facilitates data interpretation along a surface. While I found no performance differences for uniform, 2D planes, incorrect shading significantly decreases color matching performance on molecular surfaces despite the subtlety of the visual differences between the displayed images (Fig. 6.9). Incorrect shadows may communicate surface structure, but may not be enough to support constancy mechanisms in interpreting encoded data—I found no evidence that incorrect shadows provide any performance gains beyond what 2D shadows provide. These findings also suggest that visualization design decisions that manipulate surface shading may influence viewers' abilities to correctly interpret surface data in visualization.

6.6 Do constancy effects preserve luminance cues in common ramps? (S4)

In practice, well-designed color ramps integrate luminance variation with other color cues. Luminance is a strong cue for identifying colors in visualization; however, shadows compress luminance variation in surface visualization. Lightness constancy effects must sufficiently preserve luminance variations in shaded regions for luminance-varying ramps to retain their performance benefits over isoluminant ramps. While §6.5 provides empirical evidence that some of these cues can be preserved, it does so using carefully engineered ramps that strictly use luminance cues in order to gauge subtle effects. In this experiment, I compared three commonly used ramps that integrate luminance variation with their isoluminant equivalents to determine if luminance cues are beneficial for color ramps used to encode surface data.

6.6.1 Methods

Stimuli were constructed as discussed in §6.4, with any colors outside sRGB gamut [Stokes et al., 1996a] clamped via chroma reduction. Color ramps consisted of a purple-white-green (PWG) and a red-yellow-blue (RYB) diverging ramp from ColorBrewer [Brewer et al., 2003b] and a rainbow ramp (Ra) akin to that used in PyMol [DeLano, 2002]. As opposed to the red and blue luminance varying ramps from the previous experiments, these ramps represent common practice color choices for surface visualization—the ColorBrewer ramps represent good

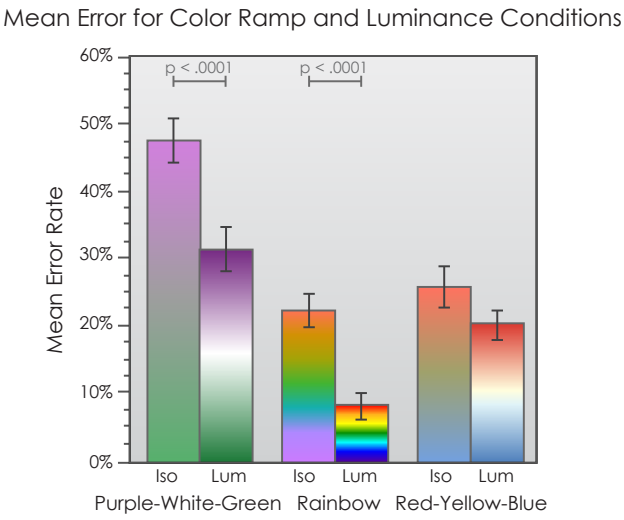


Figure 6.10: Luminance-varying ramps supported significantly better performance than their isoluminant equivalents, suggesting that lightness constancy helps viewers interpret data encoded with well-designed color ramps (**S4**).

practice for data encoding, while the rainbow ramp provides an example of extreme hue variation that is used in practice but suffers from several known limitations [Borkin et al., 2011, Borland and Taylor, 2007].

The isoluminant variations of these ramps were computed by setting the CIELAB L^* values of the ramps to $L^* = 65$, near the average luminance of all ramp colors. While the distance between colors is reduced in the isoluminant ramps, this reduction is only in lightness. As a result, I can gauge if lightness constancy effects are sufficient to preserve the luminance cues in the original ramps. To control for this compression, I verified that consecutive color steps in all ramps differed by at least three times our benchmark JND. A pilot identified three sample values from each ramp as potentially misidentified colors to be used as test colors: dark purple, light purple, and mid-green for PWG; orange, red, and mid-blue for RYB; and orange, cyan, and purple for Ra.

The experimental procedure was identical to that described in §6.4, with luminance treated as a between-participants factor. Each participant saw colors from one isoluminant ramp and a different luminance-varying ramp (PWG with isoluminant RYB, Ra with isoluminant PWG, and RYB with isoluminant Ra). Data was collected from 92 participants on Mechanical Turk. One participant was excluded from each of the PWG/isoluminant RYB and the RYB/isoluminant Ra condition for performance on the unshadowed validation stimuli, resulting in 30 participants per condition for analysis.

6.6.2 Results

As the difference between consecutive colors varied between ramps, I used the number of ramp units between the original and response color as our primary measure and absolute correctness as a secondary measure (Fig. 6.10). Since a direct mapping exists between the isoluminant and luminance varying ramps, this primary measure uniformly quantifies performance differences between ramps despite the fact that this difference is not necessarily uniform in color space.

I analyzed the primary measure using a three-way ANCOVA (luminance variance, shadow, and ramp). Overall, luminance-varying ramps significantly outperformed isoluminant ramps ($F(1, 1664) = 23.5519$, $p < .0001$). Performance varied significantly across shadow level ($F(2, 1664) = 53.8479$, $p < .0001$), and color ramp ($F(2, 1664) = 47.0705$, $p < .0001$). Both PWG and Ra ramps significantly outperformed their isoluminant equivalents ($F_{PWG}(1, 1664) = 17.3171$, $p_{PWG} < .0001$ and $F_{Rainbow}(1, 1664) = 9.1601$, $p_{Rainbow} = .0025$). While RYB outperformed isoluminant RYB on average, the difference was not significant ($F(1, 1664) = 1.4980$, $p = .2211$).

These results suggest that performance gains from luminance variation in well-designed ramps are preserved for ambient occlusion surfaces. This performance gain implies that lightness constancy matters in practice for surface visualization—luminance is a strong color cue; if designs better support lightness constancy, they will improve visualization effectiveness in practice.

6.7 Affects of Depth and Shape Cues

The results of the previous experiments indicate that lightness constancy may enhance the apparent color of surface data on molecular surfaces rendered with ambient occlusion. My first three studies (**S1**–**S3**, §6.5) suggest that the spatial cues created on a surface by ambient occlusion shading support participants in accurately matching surface colors to a key. Ambient occlusion provides both shape and depth cues. Both may enhance perceptions of surface structure, but, according to previous work, these factors may influence performance on our color matching task in different ways [Gilchrist and Annan, 2002].

Molecular surface visualizations often supplement ambient occlusion with other rendering techniques that provide additional structural cues. Adding depth cues to ambient occlusion surfaces can improve depth perceptions [Mather and Smith, 2004], but is unclear if these added cues will significantly increase color

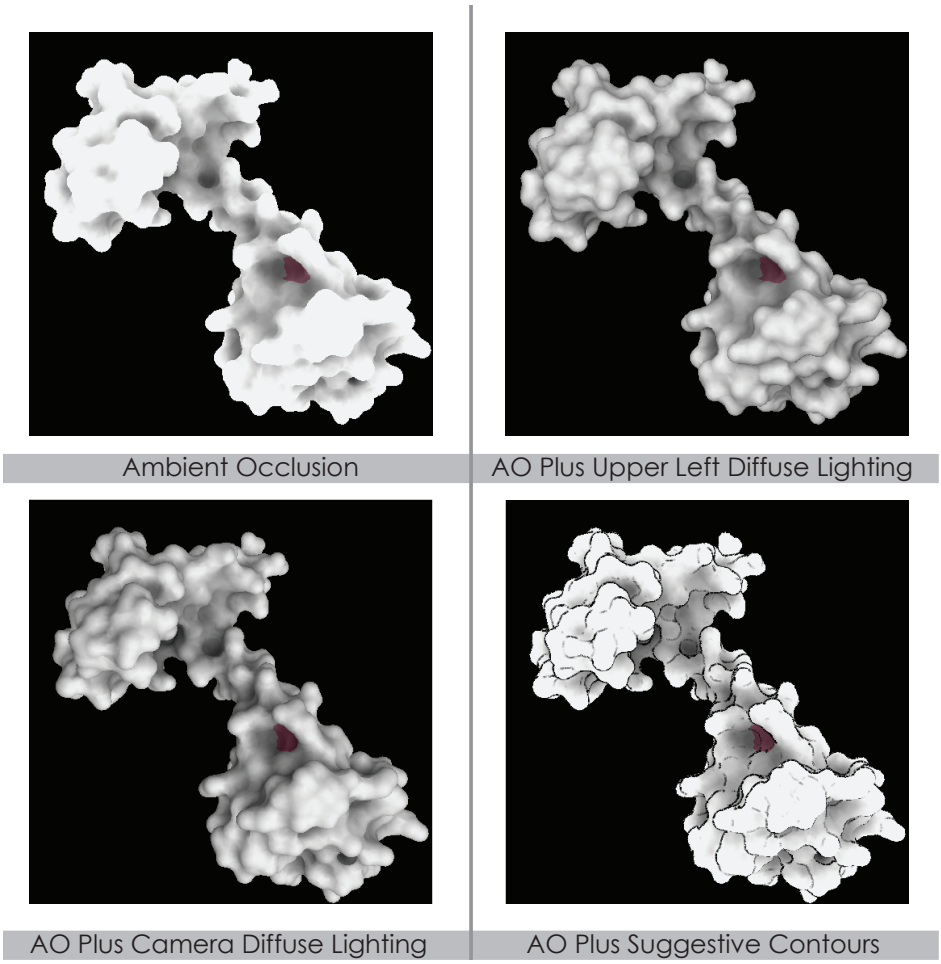


Figure 6.11: We compared different additions to ambient occlusion from the molecular visualization literature to explore how design influences viewers' abilities to interpret shadowed colors: diffuse local lighting (both sourced at the camera and in the upper left) and stereo viewing to enhance depth, and suggestive contours to enhance shape.

identification performance [Gilchrist and Jacobsen, 1984]. Strict shape cues can damage the abilities of viewers to infer shape from shading for a surface rendering [Kennedy and Bai, 2000], which results from **S3** suggest may, in turn, reduce performance. In this section, I compare three techniques commonly used in conjunction with ambient occlusion to enhance depth and shape cues in molecular visualization: directional lighting [Pettersen et al., 2004], stereo viewing [DeLano, 2002], and suggestive contours [Tarini et al., 2006] (Fig.6.11) to test how depth and shape cueing affect participants' abilities to interpret surface colors.

6.7.1 How do added depth cues from directional lighting affect constancy? (S5)

Local directional lighting is commonly used to supplement ambient occlusion in molecular surface visualization. This provides both an estimatable lighting direction and increased depth cueing, both of which may enhance constancy effects over ambient occlusion alone Hedrich et al. [2009], Ruppertsberg et al. [2008]. I anticipate that adding local directional lighting may improve color matching performance for surface visualization. This improvement might be dependent on the position of the light source, which influences how significantly the added lighting improves depth perceptions [Langer and Bulthoff, 2001]. I explored the relationship between local directional lighting and constancy effects using light sourced at two positions: from the upper left where it is strongly correlated with depth perceptions or from the camera where it provides substantially less depth cueing.

Methods

Stimulus images were generated as described in §6.4.1, with colors drawn from the seven-step red luminance-varying ramp (§6.5.1) and purple-white-green diverging ramp (§6.6). I generated two stimuli collections, each consisting of one set of visualizations using ambient occlusion alone and a corresponding set using ambient occlusion plus a directional light. The directional light was positioned at the camera in the first collection and to the upper left of the molecule in the second. Surface patches were slightly displaced from the previous experiments and between each collection to account for variation in shadow introduced by the directional shading, but patch placement was identical within each collection.

Supplementing ambient occlusion with directional lighting could cause surfaces to be significantly lighter than with ambient occlusion alone. This would potentially confound the experiment: superior performance of directional lighting may be caused by lighter shadows rather than structural cues introduced by directional lighting. To avoid this confound, I implemented directional light as diffuse shading and computed surface shading using the equation $A = 0.5 \times ao + 0.5 \times ao \times (\hat{l} \cdot \hat{n})$, where ao is the ambient occlusion value, $\hat{l}$ is the unit vector from the center of the molecule towards the light source and $\hat{n}$ is the unit normal. This model bounds the surface shading such that the directional plus ambient occlusion surface shading is never lighter than the raw ambient occlusion values.

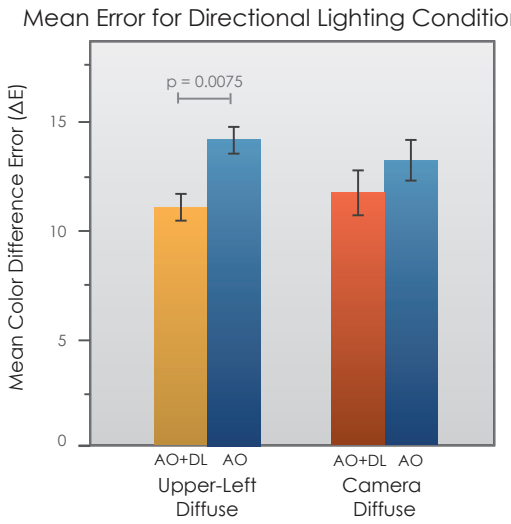


Figure 6.12: Adding directional lighting to ambient occlusion significantly improved viewers' ability to identify colors in shadow; however, this improvement appears to be correlated with the amount of depth cueing (**S5**) provided by the lighting direction.

The experimental procedure was otherwise identical to that described in §6.4. Each participant saw 32 stimuli from exactly one lighting condition. Data was collected from 108 participants on Mechanical Turk. Two participants were excluded from the upper left lighting condition and three from each condition in the camera-sourced lighting collection for performance on the validation stimuli, resulting in 25 participants per condition.

Results

I ran a three-way ANCOVA (shading model, shadow, and ramp) on the differences between the original color and participant responses for each stimulus collection (Fig. 6.12). I found significant main effects of shadow ($F_{upperleft}(1,801) = 76.1552$, $p_{upperleft} < .0001$; $F_{camera}(1,676) = 59.7345$, $p_{camera} < .0001$) and ramp ($F_{upperleft}(1,801) = 10.4646$, $p_{upperleft} = .0013$; $F_{camera}(1,676) = 10.4646$, $p = .0013$). Surface visualizations with directional lighting supported significantly better color judgments than ambient occlusion surfaces alone when the light was positioned to the upper left of the molecule ($F_{upperleft}(1,801) = 7.1918$, $p_{upperleft} = .0075$). Adding camera-sourced directional lighting also improved perceptions on average, but the difference was not significant ($F(1,676) = 1.4369$, $p = .2311$).

These results indicate that enhanced depth cues may be more important to interpreting color-coded data along a surface than lighting direction: lighting

sourced to the upper left of a surface provides both better depth cueing [Langer and Bulthoff, 2001] and greater color matching performance than lighting sourced at the camera. Artificially bounding shading in the directional lighting conditions makes shadows generally darker than in the baseline ambient occlusion condition. Therefore, I anticipate that effects seen in this experiment will likely increase in practice without this bound.

6.7.2 How do added depth cues from stereo viewing affect constancy? (S5)

The previous experiment provides evidence that enhancing depth cues may increase how accurately participants interpret color-coded information on a surface. Stereo viewing also increases depth cueing and is supported by many commercial molecular visualization packages DeLano [2002]. Experiments in psychology have found that the binocular stereo cues may improve constancy effects [Buckley et al., 1994, Yang and Shevell, 2002]; however, these studies are based on simple stimuli and do not use commercial stereo devices. I anticipate that stereo depth cues might improve participants' abilities to correctly identify color on surfaces, but it is unclear if other tradeoffs made by stereo viewing, such as reduced color fidelity, will outweigh these effects.

Methods: I tested stereo viewing using a within-subjects pilot study on a passive stereo display (Zalmon Trimon ZM-M220W). Two stimulus sets were generated: one consisting of row interlaced stereo visualizations and another with the corresponding monocular images. All stimuli assumed a uniform interpupillary distance. Participants were initially screened for stereo blindness and then shown a sample stereo molecule as asked to adjust their position until the object appeared as a continuous, three dimensional shape. The procedure was otherwise identical to that described in §6.5.1. Participants wore polarized stereo glasses through the entirety of the study in both the stereo and monocular conditions.

I compared stereo and monocular viewing in an in-person pilot across four participants. Because stereo viewing relies on proper display technologies and is highly sensitive to a number of parameters, I required participants to have prior experience with stereo viewing. This constraint limited the number of participants we were able to recruit. To help account for the limited number of participants in this study, I doubled the number of shadowed stimuli seen by each participant (each participant saw each shadow condition twice per tested color)

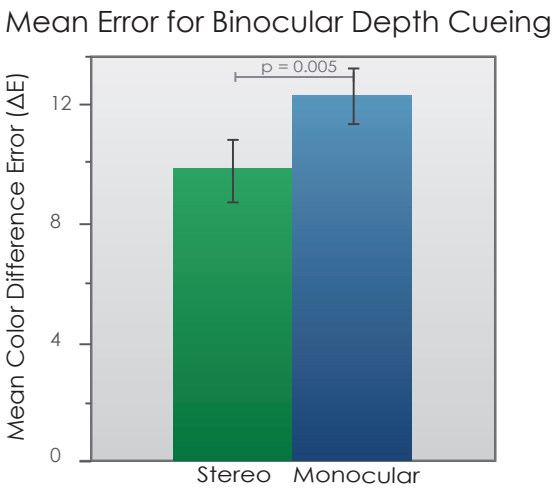


Figure 6.13: Surface color perceptions improved when molecular surfaces were supplemented with binocular depth cues (**S5**).

and treated stereo and monocular viewing as within-participants factors. Stereo and monocular viewing were blocked, with participants waiting at least 24 hours between each block to mitigate learning effects. While the size of this pilot limits the statistical power of our results, I did find significant results that I believe offer initial insight into the influence of stereo vision on constancy effects.

Results

I ran a three-way repeated measures ANCOVA (stereo, shadow, and ramp) on the differences between the original color and response color to compare stereo and monocular viewing, with block order included as a random covariate. I found significant main effects of shadow ($F(2, 278) = 14.4426$, $p < .0001$), viewing type ($F(1, 278) = 8.0351$, $p = .0050$), and ramp ($F(1, 278) = 4.8838$, $p = .00279$), but no significant interaction between block order and viewing condition ($F(1, 278) = 0.9097$, $p = .3412$). **Surfaces viewed in stereo supported more accurate color identification** than in the monocular condition (Fig. 7.7a). These findings support the observations about **S5** in §6.7.1: the binocular depth cues provided by stereo viewing may improve overall color identification in surface visualization. Further study is needed to verify the magnitude of this effect.

6.7.3 How do added shape cues affect constancy? (S6)

Sections 6.7.1 and 6.7.2 together indicate a correlation between depth cues and performance on our color matching task. However, techniques like directional lighting also improve perceptions of surface shape. Li and Pizlo [2011] shading and other depth cues are of secondary importance for comprehending shape compared to cues from edges and contours. The two sets of cues may be processed differently by the visual system. To reason about how shape perceptions might influence performance, I also measured performance for ambient occlusion surfaces with suggestive contours. Suggestive contours [DeCarlo et al., 2003] are used to in surface visualization to enhance representations of surface shape. Contours use lines instead of shading to emphasize high-level depth discontinuities along the surface, creating an image resembling a hand-drawn sketch. In previous studies [Kennedy and Bai, 2000], adding contours to a shaded surface inhibited shadow perceptions. Given the importance of shadow perception for constancy effects, as suggested by **S3** (§6.5.3), contours may consequentally inhibit participants' abilities to accurately map surface colors to their corresponding original color.

Methods

Two sets of stimuli were again generated: one consisting of visualizations using ambient occlusion and a corresponding set using ambient occlusion plus suggestive contours. Contours were generated using the implementation provided by DeCarlo et al. in the TriMesh package [DeCarlo et al., 2003] and layered on top of the original ambient occlusion surface. The procedure was otherwise identical to the previous experiments (§6.4).

Data was collected from 55 participants from Mechanical Turk. One participant was excluded for poor performance on the validation questions, resulting in 27 participants per condition.

Results

I ran a three-way ANCOVA (contours, shadow, and ramp) on the difference between the key and response colors. I found a significant main effect of shadow ($F(1,278) = 57.0108$, $p < .0001$). Adding contours marginally decreased performance over ambient occlusion alone ($F(1,278) = 2.7329$, $p = .0986$, Fig. 6.14). The marginal decrease in performance points to a potential trade-off between shape

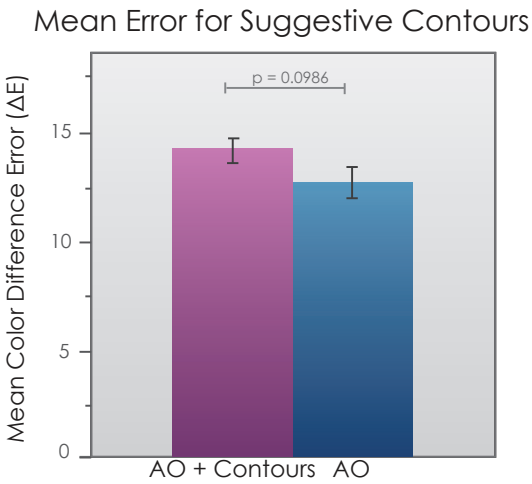


Figure 6.14: Enhancing shape using contours resulted in marginally decreased performance over ambient occlusion alone (**S6**).

and color identification performance for non-photorealistic rendering techniques in surface visualization. Combined with previous results [Kennedy and Bai, 2000], this provides further evidence of that accurate shadow perceptions improve the interpretation of color-coded data.

6.8 Discussion and Design Implications

In surface visualization, viewers explore data in the context of surface structure. Surface structure is commonly conveyed through shadows and shading, which may obscure information encoded on the surface. Supporting lightness constancy in visualization can improve how well a visualization supports accurately reading encoded data in shadow. The results, summarized in Table 6.1, demonstrate that despite several approximations made in surface visualizations rendered using ambient occlusion, viewers are able to interpret color-coded data along these surfaces. Performance for this task is directly influenced by visualization design. **S1** through **S3** isolate constancy effects and suggest that the visual system is leveraging information about synthetic shadows to disentangle encoded data from surface features, and **S4** suggests that these effects are sufficient to preserve performance gains from luminance variation for common color mappings. **S5** and **S6** inform how visualization design can influence color identification performance.

Visualization techniques generally represent trade-offs: they often improve performance for certain types of task at the expense of others. Surface visu-

alizations have traditionally been concerned with supporting depth percepts to convey structure and color percepts to convey additional data about that structure. These findings suggest that improving percepts of depth and of color may go hand-in-hand: techniques that enhance the apparent depth of a visualized surface may also improve how effectively the visualization communicates encoded data. Designers can leverage this correlation to develop visualizations that effectively convey surface data in context. Supplementing ambient occlusion with other visualization techniques that enhance depth perceptions, such as directional lighting or stereo viewing, may improve how effectively viewers can interpret information encoded using color on a surface.

This coordination between depth and color may be part of a delicate balance. The visual system appears to be sensitive to the methods used to communicate surface structure. **S3** and **S6** collectively suggest that some design choices can hinder perceptions of surface colors. The visual system likely processes shadows generated by ambient occlusion as more than simply surface shading. Small variations that damage the physical basis of these shadows can significantly diminish viewers' abilities to correctly interpret color encodings. Further, that contours marginally degraded constancy effects suggests that simply enhancing the perception of surface shape is not enough to improve this ability. Such design decisions may represent a tradeoff between perceptions of encoded data and of surface structure and could be used to inform task-driven design.

I anticipate that these findings will generalize to other types of surfaces beyond solvent-excluded molecular surfaces. While molecular surfaces represent a realistic use case where correctly inferring data in shadowed regions is often important, these surfaces are unfamiliar complex visual structures to our non-expert participants. The tested surfaces in the context of these studies therefore simply represent smooth, amorphous structures. As constancy effects are influenced by object familiarity [Olkkonen et al., 2008b], I would anticipate that for naïve observers, these results provide a baseline measure for color identification performance in ambient occlusion surface visualization more generally. Although these structures do not represent all possible surface structures (they are continuous and have no sharp corners), I believe that these results generalize to surfaces beyond solvent-excluded molecular surfaces but recognize that verifying this is important future work.

Study	Conclusion
S1	Viewers can read shadowed colors on ambient occlusion surfaces
S2	Structure helps viewers interpret shadowed colors
S3	Precise shadow information supports accurate color interpretation
S4	Luminance cues improve performance for common ramps
S5	Visualization techniques that improve depth perception enable viewers to more accurately identify shadowed colors
S6	Visualization techniques that improve shape perception may not improve performance

Table 6.1: Summary of Results

6.9 Limitations and Future Work

This work represents initial steps in understanding how visualization design can support viewers in accurately interpreting color encodings for effective surface visualization. I focus on measuring performance across to a small set of common design decisions for molecular surface visualization. Exploring other aspects of design could provide a deeper understanding of how to better support color encodings and other percepts in surface visualization, such as exploring effects of interaction, ramp design, or other shadow approximations like depth darkening. Comparison to more rigorous global illumination models could help illuminate how the approximations made by ambient occlusion influence surface perception. Generalizing these explorations across additional surfaces (e.g. space-filling models) or to surfaces in other domains (e.g. aerodynamics) would create a more general understanding of how we can consider perceptual phenomena to inform effective surface visualization design.

The task used in these studies was somewhat artificial by necessity. While identifying color values on a surface is a standard visualization task, each image in our experiments had only one colored patch. This removes potential complications due to contrast between patches or judgments from comparing multiple surface patches, both of which are possible in standard scenarios but would interfere with our ability to measure color identification performance as a function of visualization design. Future explorations might consider more complex tasks.

6.10 Conclusion

In this chapter, I show how visualizations can be designed to support better performance on point tasks using color for a specific use case: surface visualizations. Color is commonly used to visualize data on surfaces. However, visualization techniques that communicate surface shape often do so using surface shading. This shading can confound data encoded using color, as colors are darkened by shadows. Lightness constancy provides a perceptual mechanism for bridging this complication, allowing viewers to interpret shadowed colors in the real world. Its effectiveness in complex synthetic environments such as surface visualizations is not well understood. I confirmed the existence of lightness constancy for molecular surfaces rendered using ambient occlusion and present an initial exploration of how visualization design can impact the effectiveness of color encodings on these surfaces. These studies offer initial insight into how a consideration of constancy mechanisms can help guide effective visualization design.

This study only looks at how visualization design can support color for one use case: surface visualization. Surface visualization is an interesting and useful case—biologists frequently explore data on the molecular surface to explore data across complex surface features—but is very niche. I hypothesize there are many other ways that visualizations could be designed to support color encodings. For example, certain background colors might work well with different color ramps due to lightness contrast effects. The models presented in Chapter 8 focus on guiding ramps as a function of size, but could equally be used to select an effective minimum mark size for a desired encoding. Exploring other ways that visualizations can be designed to support point tasks with color is important future work.

7 ADAPTING COLOR DIFFERENCE FOR DESIGN

The previous chapter demonstrates how visualization design can improve how effectively color communicates data. In most scenarios, adapting the design of a visualization to improve how accurately viewers interpret color is not feasible—color is generally just one of the features used to encode data values. Instead, I argue that effective color encodings can be designed to account for the anticipated design of a visualization. That is, designers can choose colors based on the expected parameters of visualization viewing. This can be done by understanding how discriminability—the ability to tell colors apart—changes in practice [Stone, 2012].

In this chapter, I introduce a method for creating data-driven metrics of color difference perception. This method allows designers to model color difference for target viewing populations and adapt existing metrics to better account for the expected perceptions of target viewers. Understanding perceptions for target populations is important as variations in viewing conditions can substantially degrade viewers' abilities to distinguish between colors [Oicherman et al., 2008, Rizzo et al., 2002]. I apply this method to model color difference perception for crowdsourced workers. The resulting model can be used to design effective color encodings for the web.

7.1 Overview

Color difference models are often used in design applications to predict how noticeably two colors will differ. These models serve several purposes, such as determining sets of colors that are subtly different or in creating an encoding that interpolated perceptual differences between two colors (Fig. 7.1); however, they model perception under laboratory conditions, with correctly calibrated displays and constrained viewing environments. Given the rapid proliferation of visual content on the web and the increasing mobility of digital devices, visual media is becoming increasingly diverse, making factors that influence color difference perception, such as lighting conditions and display properties, highly variable in everyday viewing. Existing color difference models, while powerful descriptors of human vision, do not consider this variability, limiting their utility in design.

CIELAB is commonly used in design scenarios as it offers a color difference formulation based on Euclidean distance (ΔE^*_{ab}). In visualization, it is commonly used in systems [Cao et al., 2010], design techniques [Wang et al., 2008], and

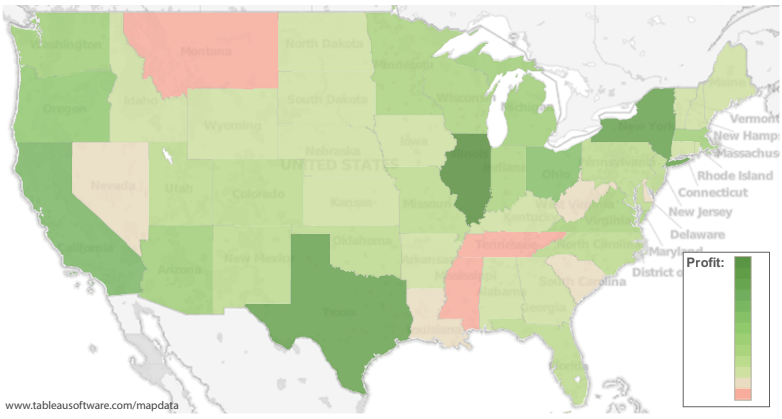


Figure 7.1: Visualizations often use color sets with large numbers of subtly distinct colors for encoding quantitative data. For example, this map encodes profit margins for different states using 13 color steps, but color differences are chosen based on expert intuitions rather than through validated metrics[1]. Metrics better tuned to visualization viewing can provide empirical guidance for designing such encodings.

evaluations [Livingston et al., 2011]. This metric is not as accurate as other appearance models, such as CIECAM02 [Moroney et al., 2002], but its simplicity makes it practical for design. In this chapter, I present an approach to adapt CIELAB to model color difference perception for real-world viewing populations that preserves its simplicity. As the range of viewing factors in these populations is too complex to model each independently, I instead capture these factors in aggregate by sampling difference perception across target viewers and use these samples to derive scaling factors for CIELAB.

The resulting model parameterizes CIELAB with respect to a given population and desired level of noticeable difference. Providing a parametric model tuned empirically to the target population exchanges the ability to make exacting claims about perceptual mechanism to instead create an *engineering* model that captures color difference in practice. This engineering model has several desirable properties for visualization designers. It is *parametric* as it can be tuned to reflect a desired range of viewers and conditions. It is *data-driven* as it derives these parameters from observations under the target viewing conditions, yet *practical* as this data can be collected quickly using a simple task and requires only small modifications to common design metrics. Additionally, the model explicitly considers the *probabilistic* nature of the data, providing designers simple controls for

defining how "noticeable" their desired color difference will be. In this chapter, I use this approach to model and validate discriminability on the web, using crowdsourced participants from Mechanical Turk. The resulting model provides an empirical metric for just-noticeable color difference for web content that aligns with common designer intuitions. To validate this approach, the crowdsourced model is compared to existing theoretical and laboratory benchmarks for color difference.

7.2 Background

CIELAB was designed such that one unit of Euclidean distance equals one just-noticeable color difference (JND). However, prior work suggests that this may be an overly optimistic estimate. Several experiments have quantified just-noticeable color differences for CIELAB, such as the empirical benchmark from Mahy et al. we use in this study ($\Delta E^*_{ab} = 2.3$) [Mahy et al., 1994b]. These studies also demonstrate that CIELAB is not fully perceptually uniform even under ideal conditions [Fairchild, 2005]. Revised models of color difference have been developed for CIELAB, such as ΔE^*_{94} and CIEDE2000 [Luo et al., 2001], that account for these nonuniformities using hue and chroma (see [Robertson, 2007] for a survey). However, these models tend be more mathematically complicated and less intuitive than the Euclidean ΔE^*_{ab} metric, improving accuracy at the expense of simplicity. Because of this trade-off, designers commonly use ΔE^*_{ab} in practice [Fairchild, 2005].

Existing CIELAB distance metrics quantify difference under laboratory conditions: lighting, display parameters (e.g. gamma and peak outputs), viewer position, and surround are all controlled. However, these factors substantially impact color difference perception [Oicherman et al., 2008, Sarkar et al., 2010, Stokes et al., 1992]. Some efforts have attempted to account for variation caused by individual viewing factors, such as ambient illumination [Devlin et al., 2006] or display media [Fairchild and Berns, 1993, Stone, 2001], but do not consider interactions between factors. More general models exist for specific contexts like airplane cockpits [Silverstein and Merrifield, 1982] or medical applications [Pizer and Chan, 1980], but it is unclear how well these models generalize beyond their target applications. My goal is to provide a color difference model that can be readily tuned to different environments and offers designers control over discriminability within those environments. I do this using a data-driven model sampled

under the target conditions (e.g. mobile devices or clinical settings). With this approach, designers do not need to consider each factor independently, but rather can account for expected variation factors in a more manageable way.

7.3 A Parametric Color Difference Model

My color modeling methodology builds on the CIELAB color difference model. CIELAB provides an effective approximation of color perception to create a space that is relatively perceptually uniform, yet sufficiently practical to use. The color space was designed such that the following assumptions hold [Fairchild, 2005]:

A1: The axes are perceptually orthogonal, so they may be treated independently.

A2: Euclidean distance (ΔE^*_{ab}) is an effective metric for perceived color difference.

A3: The axes are perceptually uniform: differences at the higher end of the scale and lower end of the scale are the same.

A4: The axes are scaled such that one unit along any axis corresponds to one just-noticeable difference.

Prior work shows that these assumptions do not always hold; however, addressing these points of failure vastly complicates measuring color differences. As a result, they are still frequently assumed in design as they exchange a small amount of perceptual accuracy for a degree of practicality desirable for many design applications. This trade-off is often worthwhile for all but **A4**. Color difference metrics are intended to tell when colors are discernable. Two colors separated by $\Delta E = 1$ will almost always appear the same (only 7% of participants in this study could detect a difference). In visualization, **A4** is sometimes addressed using non-empirical intuitions, but these intuitions are based on experience and what "looks correct" to a well-trained eye on a single display. They do not necessarily generalize well and are not grounded in user perceptions. More frequently, CIELAB is used as intended (a JND mapped to $\Delta E = 1$) irrespective of this known imperfection.

The model presented in this chapter aims to empirically adapt CIELAB such that **A4** holds for the designer's desired definition of "just noticable." Accepting the first three assumptions allows the model to do so using a simple extension to the CIELAB model. I model discriminability along each axis independently on the basis of **A1**, which has the added benefit of empirically correcting any imbalance

(a) Color Matching Stimulus (b) Forced Choice Stimulus

Figure 7.2: Free-response color matching tasks provide insight into discriminability, but are of limited utility for probabilistic modeling. We use a forced choice microtask to measure discriminability as a function of color difference.

between lightness and chroma. **A1** and **A3** collectively allow the model to use a single scaling factor for each axis, denoted as $ND_L(p)$, $ND_a(p)$, and $ND_b(p)$, such that a difference of one unit along the scaled axis is noticeable for p% of the target viewing population. I derive these scaling factors using simple tasks to quickly measure discriminability across color differences for the target population and model this discriminability linearly. These scaling factors normalize each color axis of CIELAB according to the designer's desired discriminability threshold using only one multiplication for each axis.

The resulting adapted color model meets the goals discussed in the introduction to this chapter: it is parametric, as the scaling factors can adapt to the viewing conditions; it is data-driven, as these parameters can be determined empirically from observations of the color difference; it is practical, as data collection can be done quickly and easily and model computations require only three multiplications beyond the standard CIELAB computation; and it is probabilistic, as designers can dynamically define their desired noticeable differences and adjust the model accordingly. I confirm this approach by modeling color difference perception for the web. The subsequent sections discuss three studies that construct and validate this example model. The first describes a color matching pilot that provides insight into the above modeling assumptions. The second illustrates the data collection and model construction methods. The third validates this model on 161 web viewers.

7.4 Insight from a Color Matching Task

Color difference is commonly measured using a free-response color matching task, where participants manually adjust a stimulus color to match a given reference color [Crawford, 1965, Sarkar et al., 2010]. The adjusted response colors provide a distribution of colors that appear to match the stimulus color. This procedure, akin to Maxwell's color matching experiments [Fairchild, 2005], has been used to measure JND for cross-media applications and provides substantial insight into color difference across color space.

I conducted a crowdsourced color matching experiment on Mechanical Turk to verify the modeling assumptions discussed in the previous section. Participants saw a 2° colored reference square centered on a 500 pixel-wide white background with a slider that adjusted the color of a second stimulus square (Fig. 7.2a). To help conceptualize these measures, 2° of visual angle is roughly the width of a thumb held at an arms length or approximately 42 pixels wide for a participant sitting 24 inches from the display. Participants were instructed to drag the slider until the adjusted color matched the reference color as closely as possible. Tested colors were sampled from the Swedish Natural Color System primaries [Hård and Sivik, 2007] and varied at equal intervals along each axis of CIELAB within the gamut defined by $\gamma = 2.2$ and a D65 whitepoint [Stokes et al., 1996b], resulting in 24 distinct colors per axis. Our modeling procedure uses a constant gamma and whitepoint as, in practice, designers cannot feasibly adjust content to such display-dependent conditions. By holding these factors constant and measuring difference perception across multiple displays, I can parameterize color difference for design using only information immediately available to designers. To simplify the color matching task for Turk workers who may be unfamiliar with CIELAB, participants were shown only one slider, corresponding to one axis of CIELAB (L^*, a^*, or b^*), as in [Alfvin and Fairchild, 1997, Sarkar et al., 2010]. The slider displayed the full color range within the gamut along the tested axis through the reference color. Axis was a between-participants factor.

I recruited 48 participants (16 per axis, 33 female, 15 male) from age 18 to 61 ($\mu = 34.32$, $\sigma = 13.22$) with normal or corrected-to-normal vision and no color vision deficiencies. Participants were screened for color-vision deficiencies using five digital renderings of Ishihara plates [Legrand et al., 1945] and asked self-report their approximate distance from the monitor, which was used in conjunction with DPI to compute the size of the 2° stimulus square. They completed a simplified

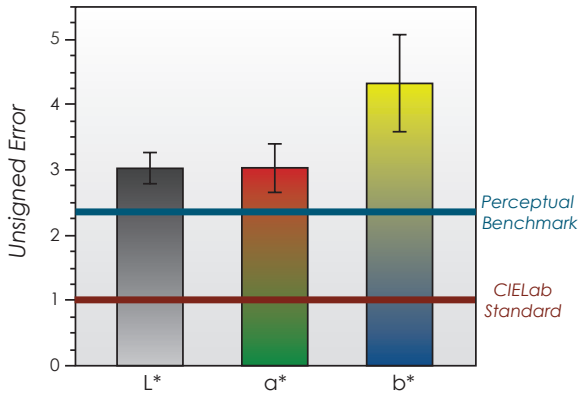


Figure 7.3: Mean error for the color matching task. Web viewing discriminability thresholds exceed existing benchmarks and may vary between axes.

tutorial to test task understanding and then asked to complete the color matching task for each of the 24 reference colors described above in a random order. Participants were given unlimited time for each response.

I analyzed the difference between response and reference colors (*error*) using a two-way ANCOVA (reference color and tested axis) with display distance and question order as covariates to account for interparticipant variation. The results confirm that color difference perception is reduced on the web: per-axis mean errors ($\mu_L = 3.21, \mu_a = 3.03, \mu_b = 4.33$, Fig. 7.3) were significantly larger than both the theoretical JND ($\Delta E^*_{ab} = 1.0$) and our empirical benchmark ($\Delta E^*_{ab} = 2.3$), suggesting that existing metrics underestimate color difference for the web. Error varied significantly between axes ($F(2, 997) = 11.2693$, $p < .0001$), but not within axes ($F_L(1, 871) = 1.6072$, $p_L = .2052$; $F_a(1, 862) = 1.8942$, $p_a = .1691$; $F_b(1, 755) = 0.1875$, $p_b = .6651$). These findings support **A3**, but suggest that each axis should be modeled independently.

However, these results do not provide probabilistic insight into color discriminability—the adjusted colors identify a window of indiscriminable colors around a given reference, but do not capture how likely it is that these colors will appear distinct from the reference for different viewers. Without controlled insight into the likelihood that colors will appear different, designers cannot tune the model to their desired application settings. In the next section, I propose a task model which addresses this limitation in order to quantify the likelihood distribution of color differences for a target population and allows quick and efficient data collection.

7.5 Constructing the Engineering Model

While the slider task provides insight into color difference perception, it suffers from a number of limitations. One limitation unique to crowdsourcing is that the sliders essentially provide continuous responses. Participants seek to complete a large number of tasks as quickly as possible to maximize their overall reward. For continuous response tasks, participants can optimize their time by providing answers that are "close enough" rather than taking the time to respond as accurately as possible.

Effective crowdsourced studies use a *microtask* model, providing simple tasks that require roughly as much time to answer accurately as to answer "close enough" [Buhrmester et al., 2011]. I designed a data collection microtask to measure how frequently colors appear to be different at specific color differences (*discriminability rate*) and use this measure to parameterize my color difference model. The task is a binary forced choice comparison of two colored squares based on the method of constant stimuli [Boring, 1917] (Fig. 7.2b). The squares differed in color by a controlled amount along one axis. Participants were asked whether the squares appeared to be the same color or different colors. Discriminability is then quantified as a probabilistic function of color difference for our sample population by measuring how frequently the squares appeared to be different colors at different levels of color difference. This method can be used to obtain a large amount of discriminability data efficiently: median response time for was 5.8 seconds per color pair.

7.5.1 Sampling Method

I can use the above microtask model to estimate per-axis scaling parameters ($ND_L(p)$, $ND_a(p)$, and $ND_b(p)$) representing the color differences along each axis perceived by p% of the target population. I computed these parameters for the web viewing model through a crowdsourced experiment involving 75 participants (37 female, 38 male) age 19 to 56 ($\mu = 31.05$, $\sigma = 9.75$) with normal or corrected-to-normal vision and no color vision deficiencies. Participants were asked to directly compare two 2° squares placed at opposite ends of a 8° white plane (Fig. 7.2b). One square was colored using a reference color randomly selected from a set of 316 colors from $L^* = 10$ to $L^* = 90$ evenly sampled from the CIELAB color space and within the color gamut defined by a standard PC gamma and whitepoint ($\gamma = 2.2$ and D65 whitepoint)[Stokes et al., 1996b]. The color of the second square

differed from the reference color by a controlled amount along exactly one color axis (between 2.5 and 8.5 ΔE^*_{ab} sampled at 0.5 ΔE^*_{ab} increments).

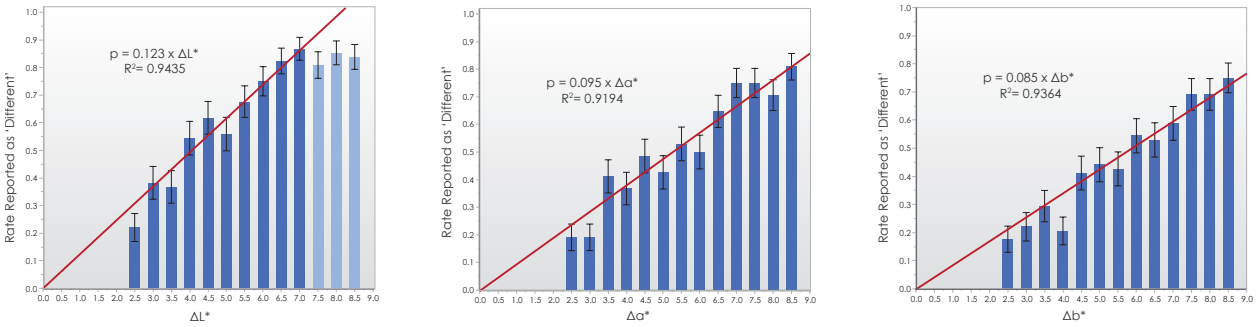
Forced choice tasks are vulnerable to gamed responses: participants could provide random answers to complete the study quickly. To help mitigate such gaming and also to unbias the stimulus set, I included 20 stimuli with identically colored squares and two with obviously different colors. Three participants answered less than 65% of the same-color questions or one extreme difference correctly and were excluded from our analyses.

Participants were first screened for color vision deficiencies using digital renderings of Ishihara plates [Legrand et al., 1945] and self-reported their distance from the display. They then completed three tutorial questions to ensure their understanding of our definition of "same" and "different" colors—two colors that varied in hue, two that varied in luminance, and two identical colors—and could not proceed until each was answered correctly. Participants were then shown a sequence of 61 stimuli (39 modeling stimuli and 22 validation stimuli) in a random order, with each reference color appearing twice and each color axis $\times$ color difference once. A two-second grey screen separated subsequent stimuli to minimize adaptation effects. Participants had unlimited time to respond.

I analyzed discriminability rates using a three-way ANCOVA (reference color, tested axis, and magnitude of difference) with display distance and question order as covariates to account for interparticipant variation. The magnitude of color difference significantly affected discriminability ($F_L(1, 2846) = 169.0197$, $p_L < .0001$; $F_a(1, 2846) = 163.0631$, $p_a < .0001$; $F_b(1, 2846) = 148.5278$, $p_b < .0001$). Discriminability also varied significantly between tested axes ($F(2, 2846) = 3.1380$, $p = .0244$). Reference color L^* significantly influenced discriminability ($F(1, 2846) = 17.3941$, $p < .0001$), but the effect was small—the lightest colors were 0.3% more discriminable than the darkest. We found no significant effects of reference a^* or b^*.

7.5.2 Parameterizing CIELAB

To derive the parameters of our engineering model, I create linear models of this sampled discriminability data. These models express the sampled discriminability rates as a linear function of color difference for each axis of CIELAB on the basis of **A1** and **A3** (Fig. 7.4). While identical colors should always have a discriminability rate of zero, sampling error can introduce noise that skews these linear models.



(a) L* (b) a* (c) b*

Figure 7.4: An illustration of our modeling approach. A linear model (red) is fitted to the rate of 'different' responses across measured differences and forced through zero to account for sampling. Only color differences where discriminability changes with distance are modeled (dark blue).

To correct for such discrepancies, I construct the models with intercepts forced through zero and further account for sampling errors by treating interparticipant variability as a random factor. Likewise, these models are only fit to data below the upper bound of discriminability (e.g where tested differences are not immediately perceivable; dark blue in Fig. 7.4a), a point referred to as the *k*nee [Carter and Silverstein, 2010].

The resulting models have the form $p = V_x d$ where x is the color axis, p is the desired discriminability rate, V_x is the slope of the model, and d is the color difference in ΔE^*_{ab}.I derive the parameters $ND_x(p)$ of the engineering model using the function

$$ND_x(p) = p/V_x \tag{7.1}$$

Assuming **A3** holds, a designer can divide color difference along each axis by $ND_x(p)$ to renormalize CIELAB such that p% of people modeled under our target conditions will detect color differences at $\Delta E_p = 1$. Control over this probability helps designers decide the granularity with which colors are sampled. For example, a model at $p = 50$% allows designers to create encodings with gradual yet detectable changes in color value. Alternatively, larger p values privilidge discriminability by ensuring color differences are more readily detected. This reduces the continuity between encoded values and also the number of possible color steps in an encoding. p can also help control for outliers in viewing conditions by considering a larger or smaller proportion of all responses. For example, $p = 50$% will be near the

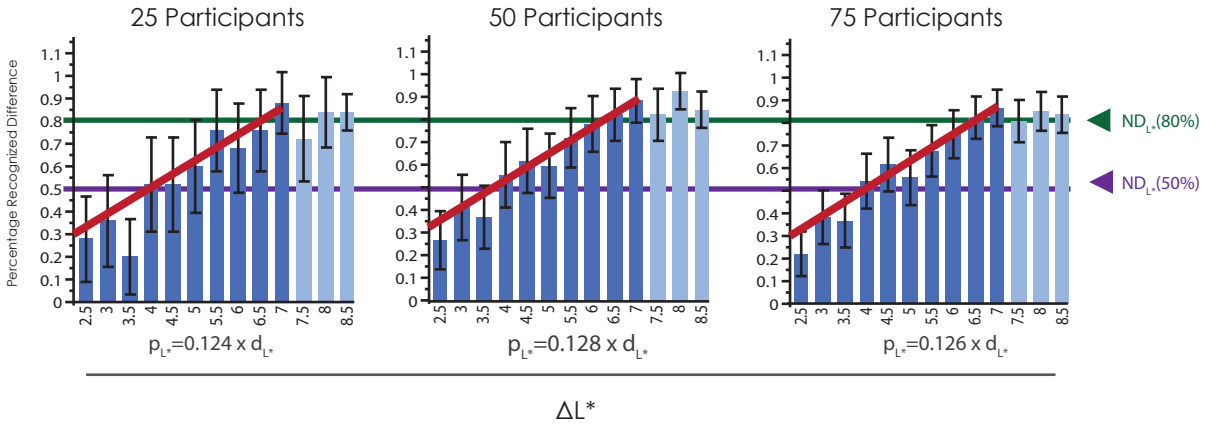


Figure 7.5: Models can be generated using a relatively few samples. As the number of samples increases, the confidence in model increases, but the parameters estimated by the model remain roughly constant.

anticipated median performance of the target population, whereas $p = 100\%$ will be more robust to worst-case performance.

Given two colors (L_1^*, a_1^*, b_1^*) and (L_2^*, a_2^*, b_2^*), ΔE_{ab}^* can be adapted to the viewing population as:

$$\Delta E_p = \sqrt{(\frac{L_1^* - L_2^*}{ND_L(p)})^2 + (\frac{a_1^* - a_2^*}{ND_a(p)})^2 + (\frac{b_1^* - b_2^*}{ND_b(p)})^2} \tag{7.2}$$

For the crowdsourced data, a traditional $p = 50\%$ JND maps to $ND_L(50) = 4.06\Delta E_{ab}^*$, $ND_a(50) = 5.26\Delta E_{ab}^*$ or $ND_b(50) = 5.88\ \Delta E_{ab}^*$ on the web (Fig. 7.4), which roughly aligns with designer intuitions often employed by a coauthor on this work who has extensive experience in encoding design. These color differences are notably larger than the CIE standard (1.0) and laboratory benchmark (2.3). Also unlike these benchmarks, these parameters vary for each axis.

I constructed models for 25, 50, and 75 participants (Fig. 7.5). These models yielded nearly identical parameter values, although the larger sample sizes provided greater statistical confidence in the parameters. This points to the practicality and reproducibility of our model: data from relatively few participants sufficiently characterize the model, and this characterization remained consistent across groups.

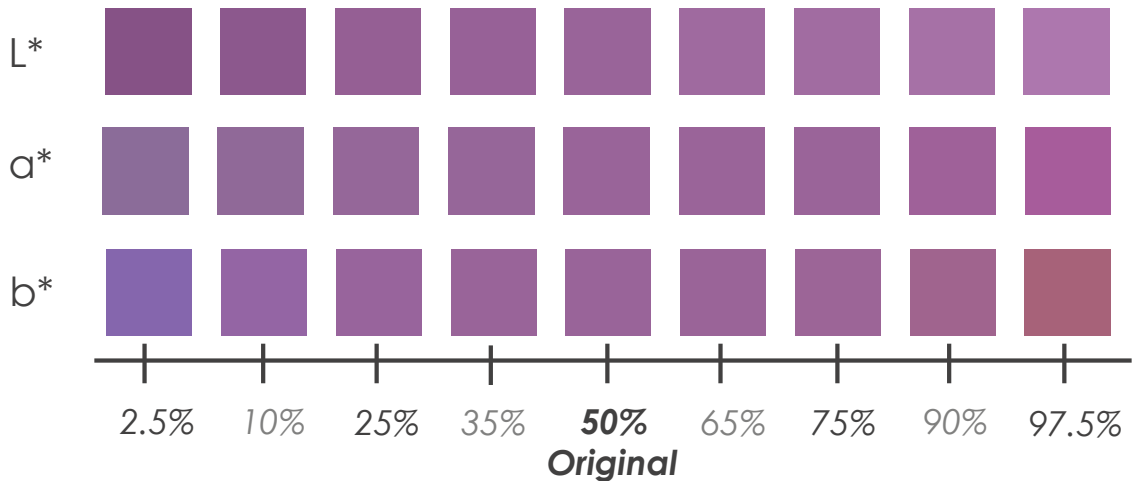
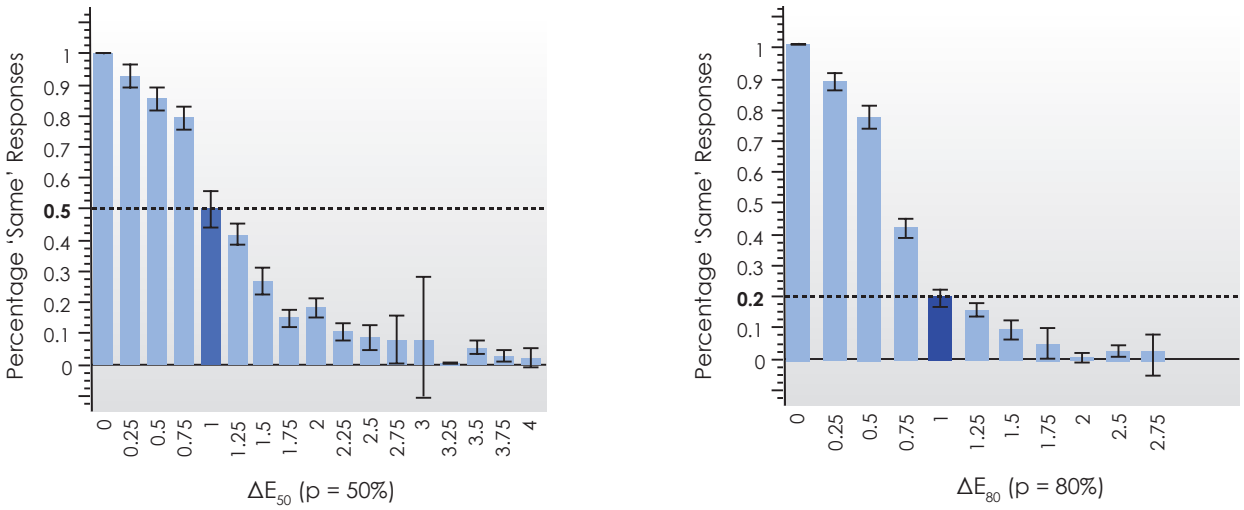


Figure 7.6: I use the error distributions from the color matching experiment to generate per-axis color differences for validating our model. The sample steps, visualized here per axis, match the reported color difference at nine sampled percentiles (x-axis) in Section 7.4.

7.6 Validating the Adapted Model

I wanted to confirm that the engineering model derived in the previous section generalizes from the smaller tuning population to the larger target population and that it makes effective predictions when the color changes are not axis aligned. This model predicts that if two colors are $\Delta E_p = 1.0$ different, then the viewing population will perceive them as different p% of the time. Colors with smaller ΔE_p will be perceived as different less frequently, and larger ΔE_p will be seen as different more frequently. I validated these predictions empirically using a second, larger group from the target population and a broader range of colors and color differences, including cross-axis differences. I collected data from 182 crowdsourced participants (106 female, 76 male) ages 18 to 64 ($\mu = 30.60$, $\sigma = 9.78$) with normal vision and no known CVD to evaluate the model. 21 participants were excluded for poor performance on the validation questions.

The validation procedure modified the parameter sampling task to include a broader range of colors and color differences. Reference colors were sampled uniformly from CIELAB, but more densely than in the parameter sampling task. Color differences were drawn from the error distributions of the color matching experiment to create nine color difference levels per axis (Fig. 7.6). For each stimuli, one difference level was applied to each axis of the reference color to emphasize variation across multiple axes. Difference level combinations were



(a) Validation results for $\Delta E_{50\%}$

(b) Validation results for $\Delta E_{50\%}$

Figure 7.7: Plotting the percentage of perceived matches against ΔE_p tuned to (a) $p = 50\%$ and (b) $p = 80\%$ differentiability for the crowdsourced model shows that the model effectively predicts noticeable difference.

drawn randomly for each participant and counterbalanced between participants. The procedure otherwise matched the parameter sampling task.

The findings validated model predictions (Fig. 7.7). Tuning the model to $p = 50\%$ discriminability predicts that participants can distinguish colors with a difference of $\Delta E_{50} = 1 \pm 0.125$ as different roughly 50% of the time. The validation data confirmed this prediction: colors at this difference were differentiable in 49.81% of trials. These predictions are considerably better than the CIELAB specification ($\Delta E_{ab} = 1.0$) or my laboratory benchmark Mahy et al. [1994b] ($\Delta E_{ab} = 2.3$), which were perceived as different for 7% and 13% of samples respectively. Further, colors less than $\Delta E_{50} = 1$ apart were consistently less discriminable, and more distant colors were perceived as more discriminable.

Model predictions were robust across discriminability levels. For example, tuning the model to 80% discriminability yields the parameters $ND_L(80\%) = 6.5$, $ND_a(80\%) = 8.42$, and $ND_b(80\%) = 9.41$). Applying this to our validation data, our population identified colors at $\Delta E_{80} = 1.0$ to be different in 80.62% of trials, confirming the model predictions. Across models at all discriminability levels, predictions were accurate to within 7% on average, and within 3.5% for models with $p \geqslant 50\%$.

7.7 Readily Recognizable Color Differences

At a certain point, color differences become readily perceptible like, for example, the difference between red and blue. Large-scale color differences are not well-modeled by CIELAB [Mahy et al., 1994b], but understanding the threshold where colors readily appear different is useful for applications in visualization. For example, a designer may want to select a set of categorical colors for hierarchical data such that related datapoints are readily discriminable, but close enough in color to be associated (e.g. one branch maps to different greens, while a second maps to different blues [Tennekes and de Jonge, 2014]).

In the proposed model, the minimum threshold at which two colors are guaranteed to be discriminable is equivalent to the parameter for $p = 100\%$. Prior work [Carter and Silverstein, 2010] has referred to this value as the "knee"—the point at which the asymptote (100% visibility) intersects with the linear model of color perception for smaller differences. However, the knee may be fuzzily defined in practice based on variations in display and limitations with CIELAB for measuring large color differences. This fuzziness may offset the knee value from that predicted by the linear model. In this experiment, I measure how perceptions change as color difference approaches the knee value.

To pinpoint the knee for web-based visualizations, I repeated the proposed data collection procedure using larger color differences. Color differences were sampled from 0 to 4 JNDs ($p = 50\%$) at 0.5 JND steps. This sampling ensures the $p = 100\%$ JND level fell roughly in the center of the samples. 27 source colors were tested, with color generated using the same procedure as in Section 7.5. The stimulus and procedure otherwise matched those in Section 7.4, with each participant seeing 106 total color pairs. I collected data 97 participants (49 female, 47 male; mean age 31, σ age 8.96). One was excluded for poor performance on the equal color stimuli.

The results are summarized in Figure 7.8. Given expected variation in reported responses, I consider color to be readily discriminable once increasing color difference no longer significantly increase the measured discriminability rate (e.g. larger color differences provide roughly the same responses). These results present an interesting pattern: for all of the reported data, the knee predicted by the crowdsourcing model ($\Delta L^* = 8.13$, $\Delta a^* = 10.53$, $\Delta b^* = 11.76$) falls short of that predicted by these results. Measuring the knee according to the above metric, these results suggest that a color difference is likely to be perceived with 100%

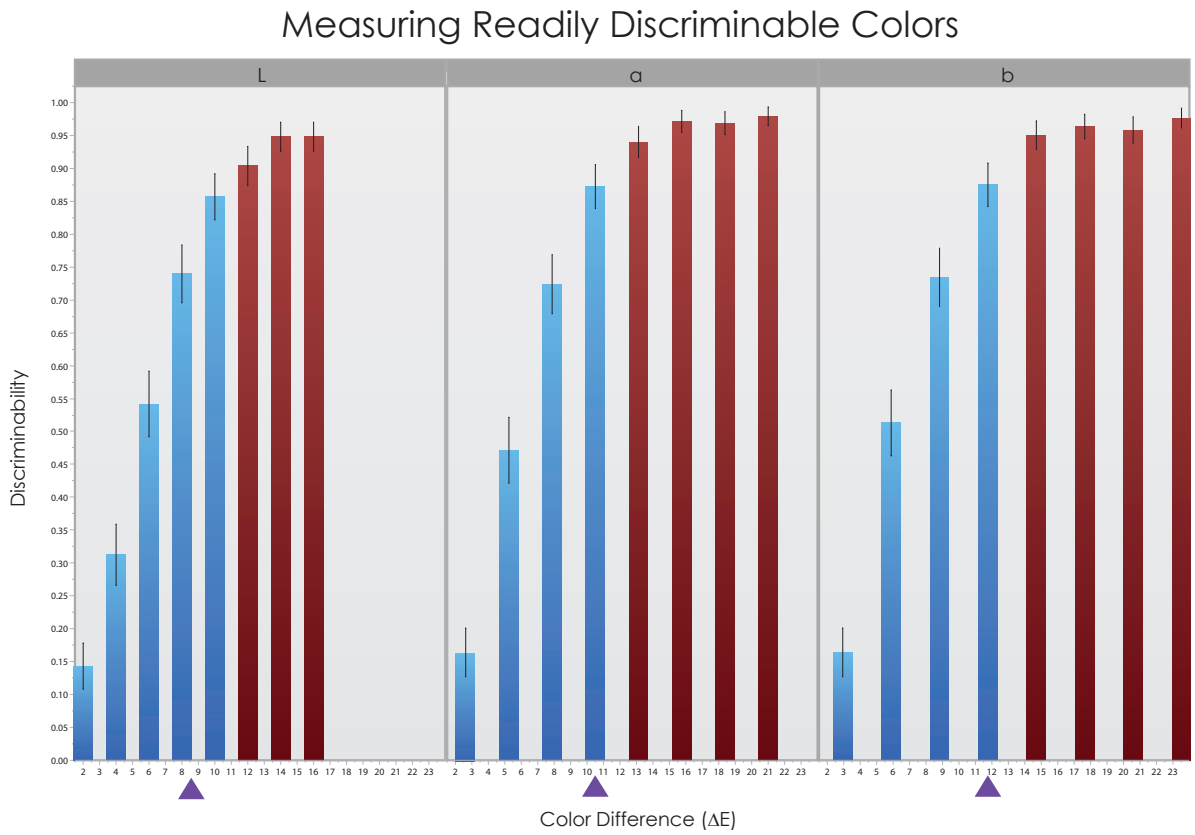


Figure 7.8: Discriminability rates for the large color differences. The knee predicted by the linear model (smallest color difference guaranteed to be discriminable, purple triangles) falls just short of where increased color differences no longer significantly increase discriminability (red bars), suggesting that perceived color difference may gradually level off.

certainty between 2 and 2.5 50% JNDs (or $\Delta L^* = 10.2$, $\Delta a^* = 13.2$, $\Delta b^* = 14.7$).

One possible explanation for this is due to sampling. Sampling introduces uncertainty in to the measures. Since this knee data is sampled over a different population that the previous metrics, expected variation might lead to different models. Each of these models would each generate a slightly different knee value. However, the shape of this data suggests that as color difference becomes readily apparent, discriminability behaves almost asymptotically. It steadily levels off, curving as color differences near 100% discriminability. This curving would extend the knee value to better align with these results and would also fit the shape of the data (Fig. 7.8).

From a perceptual standpoint, asymptotic behavior near the knee makes sense—CIELAB is notoriously poor with large color differences [Mahy et al., 1994a]. As colors become readily discriminable, some degredation in the CIELAB model might

result in a curving structure as perceptual linearity falls off. Verifying the precise structure of the knee is important future work.

7.8 Discussion: Limitations and Applications

The core feature of this modeling approach is that it is empirical: designers can tune the model parameters by sampling the target viewing population. A myriad of factors can influence color difference perception, ranging from displays to viewing environments to the viewers themselves. Rather than trying to analyze each potential factor, I instead capture their effects in aggregate empirically. This allows models to readily adapt to specific settings. For example, a designer can model senior citizens using tablets in dimly lit cafes or students viewing projectors in classrooms by simply sampling these populations to construct specific model instances. The fact that the model captures anisotropy in color difference perception helps the models provide good predictive performance and is also beneficial for visualization applications that leverage multiple aspects of color to encode information, such as luminance and hue.

The data-driven nature of this model is both a strength and limitation. Sampling quickly captures the specific conditions of a population; however, it offers no insight into how well an adapted model transfers between target populations. The efficiency of the microtask modeling approach helps alleviate this concern: large amounts of data can be collected quickly from a lay population. An additional limitation of this sampling method is that it does not characterize specific local viewing factors, such as gamma, ambient lighting, peak color output, and whitepoint. While this is a significant limitation from a colorimetry standpoint, it is a strength from a design standpoint as designers using color difference for cross-media applications do not necessarily have access to these variables when creating visual content; assuming constant parameters mirrors what designers do in practice.

Aspects of stimulus presentation, such as stimulus size and background color, may effect the results. The next chapter explores these issues, and, in practice, I still expect this model to achieve reliable results that significantly improve the discriminability of color encodings constructed using this metric. Also, I have only assessed a small number of applications to date. However, the fact that this approach works well on the challenging case of the web and also constructs plausible models for other factors, as explored in the next chapter, suggests that

it will be effective in other scenarios. I hope to explore new scenarios, such as mobile devices, in future work.

While the focus of this discussion is on visualization, color difference models are useful in a wide range of design applications including marketing, graphic design, digital art, image compression Eckert and Bradley [1998], segmentation Bhoyar and Kadke [2010], and watermarking Podilchuk and Zeng [1998]. As displays become increasingly mobile, designers must consider a broader range of conditions and devices when designing for such applications. Metrics for color difference in design need to consider how to generalize laboratory models to fit these real-world design requirements. The parametric color difference model presented here attempts to capture real-world perceptions for specific populations using a relatively small amount of data. This model also helps normalize color difference *between* color axes in practice, for example, to balance lightness and chroma for color map construction in visualization.

7.9 Conclusion

In this chapter, I present an engineering model of color difference for applications in design. This model attempts to account for variation in viewing condition by reparameterizing CIELAB using data sampled directly from a target viewing population. This approach allows us to account for the broad variety of factors encountered in modern design scenarios by creating a model that is parametric, data-driven, probabilistic, and practical.

The size of the marks that encode data also influences how different encoding colors appear. Most color metrics are defined for targets that are 2° (roughly thumbwidth at arms length or 42 pixels 24 inches away) or 10° (roughly the size of a fist at arms length or 203 pixels 24 inches away) of visual angle wide [Berns, 2000]. However, visualization designers have targets of many sizes to consider. A significant body of work has shown that viewers' abilities to distinguish between colors varies with size [Carter and Silverstein, 2012, Fairchild, 2013]. In this chapter, I show how designers can take advantage of anticipated bounds on the size of marks in a visualization to create discriminible color encodings.

When creating a visualization, a designer often sets an explicit range on the size of a mark, such as the minimum and maximum size of a scatterplot point or the width of a bar in a bar chart. If designers understand how perceived color differences change for marks of different sizes, they can use bounds on mark size to design color encodings that are guaranteed (with some probability) to be discriminable. However, there are as of yet no practical models to help designers control for effects of mark size on encodings. In this chapter, I use the techniques introduced in the previous chapter to model color discriminability as a function of mark size. I then show how designing for lower bounds on size can guarantee discriminability using bar charts with bars of a fixed width and variable height. I conclude by discussing how designers can use these metrics in practice to create color encodings that are robust to size variation for a given visualization design.

8.1 A Model of Color Difference Perception for Mark Size

Visualization uses a variety of visual features to represent data, including size. The ability to distinguish between encoded colors for marks of different sizes is important for point tasks in data visualization [Stone, 2012]. However, this ability degrades as marks grow smaller. In Figure 8.1, the bar colors, which indicate categories of products, are easy to distinguish. However, for smaller marks such as those in the scatterplots, the same colors become less visibly distinct. Colors in well-designed palettes may be distinguishable at large scales, but break down as marks get smaller. For example, in Figure 8.2, it is difficult to distinguish the

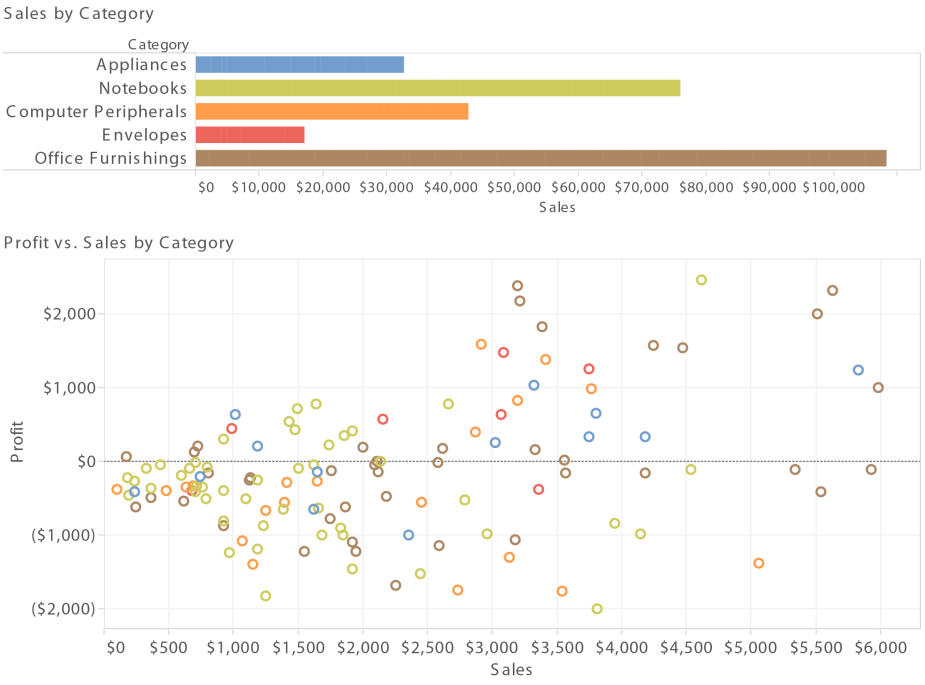


Figure 8.1: Colors that are comfortably distinct on bars are more difficult to distinguish on the small scatterplot marks.

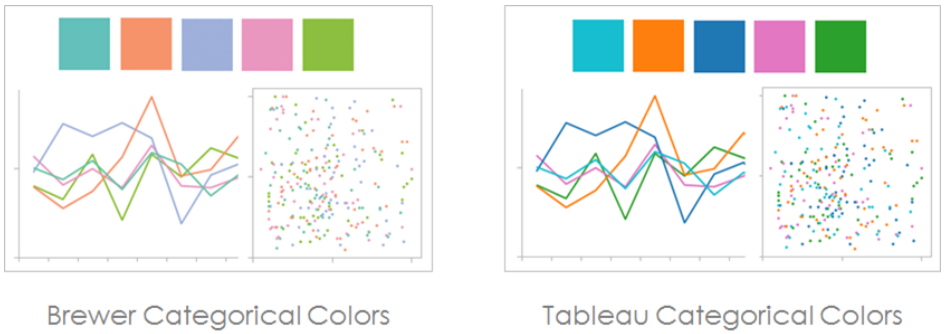


Figure 8.2: Colors that follow good design practice for large marks may not remain effective for small marks. For example, it is difficult to distinguish the pink and orange marks from this ColorBrewer ramp in a scatterplot with small marks (left). Handcrafted solutions have been used in previous systems (right) but require substantial expertise to design correctly.[1]

first and third and second and fourth Brewer colors when mapped to scatterplots.

Some systems, such as Tableau [2], use colors carefully crafted to be robust across sizes. Designing these encodings requires a great deal of expertise and a large degree of testing to determine their robustness. Further, color encodings that are robust to mark size may not be visually appealing for all mark sizes. While

[2] http://www.tableausoftware.com/

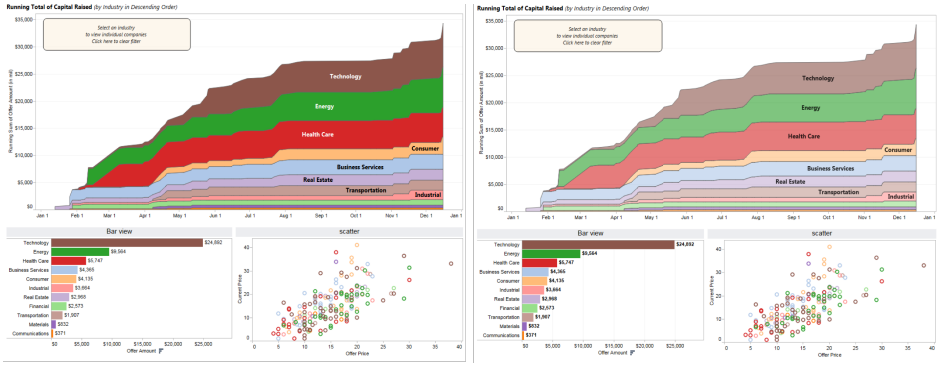


Figure 8.3: Colors that are robust for small marks may not be visually appealing for larger marks. For example, the colors in a scatterplot may be too saturate for an area graph (left). Designers can use heuristics to manually adjust colors for different kinds of marks; however, this process can lead to undesirable inconsistencies between different visualizations in a display (right).

systems like Tableau provide heuristics to handle these limitations in practice, these heuristics often introduce undesirable inconsistencies when using multiple visualizations simultaneously (Fig. 8.3). Instead, my goal in this chapter is to provide quantitative metrics for designers that allow them to create encodings that are probabilistically discriminable for different mark sizes. I will do this by modeling how discriminability changes as marks decrease in size.

In this section, I construct a quantitative model of how color discriminability changes as a function of size. This model emphasizes discriminability for small perceptual differences, such as just-noticeable differences (JNDs). I built this model using the method presented in the previous chapter to measure perceptions for real users in real viewing conditions. This helps create a model that is probabilistically robust to viewing variation. While the resulting model cannot necessarily model the visual system's specific sensitivities to color, it provides practical guidance for visualization design.

In this section, the "size" of a mark refers to its width. In practice, the size of marks can vary along multiple dimensions, such as width, height, or arc length. I aim to model color difference as a function of size in a way that can be applied generally. Rather than focus on a specific axis of size variance, I model discriminability for marks of uniform height and width. The experiments discussed here show that discriminability is better characterized as a function of width rather than the related measure of area. As a result, the model can be applied to visualization designs by considering the smallest dimension of a mark (e.g. length or width). Constraining color perceptions to the smallest dimension

ensures that colors will appear at least as distinct as predicted—increasing size along any dimension should increase the apparent difference between colors.

The model is derived from target sizes ranging from 6 degrees (approximately 120 pixels wide) to $\frac{1}{3}$ degree of visual angle (approximately 7 pixels wide). The resulting color difference function, $ND(p, s)$, provides a weighted Euclidean distance in CIELAB space, parameterized by two factors: a threshold p, defined as the percentage of observers who see two colors separated by that value as different, and a target size s, specified in degrees of visual angle. For example, a theoretical CIELAB JND is parameterized as $p = 50\%$ and $s = 2^\circ$ in this formulation.

Under this model, discriminability decreases as the target size shrinks and the difference in discriminability along each of the three axis changes unevenly. The decrease is dramatic. While these results also find that a traditional 2 degree JND on the web is near $\Delta E = 6$, for 0.33 degrees, the same discriminability is at a distance of roughly $\Delta E = 11$, with an even stronger variation in weightings along the three axes.

8.1.1 Procedure

To rescale CIELAB as a function of size, I used the same data collection proceedures as in Section 7.5: participants were asked to report whether two colors a fixed distance apart appeared the same or different. I calculated scaling factors for the L^*, a^* and b^* axes using the frequencies of these responses across different sizes and color differences.

8.1.2 Design

The models was again constructed using Mechanical Turk to measure color discriminability for web viewing. Participants were shown a series of pairs of colored squares and asked to identify whether the pairs were of the same color or different colors by pressing one of two keys (“f" key if the colors appear the same, and the “j" key if the colors appear different).

As in the previous model, for each pair of colors, one square was a standard sample, and the second differed by a small step along one of the three CIELAB axes. The position of the differing square was randomized for each stimulus. A set of 52 sample colors were selected by sampling uniformly along the L^*, a^*, and b^* axes and removing colors falling outside the gamut for the largest difference step (Fig. 8.4).

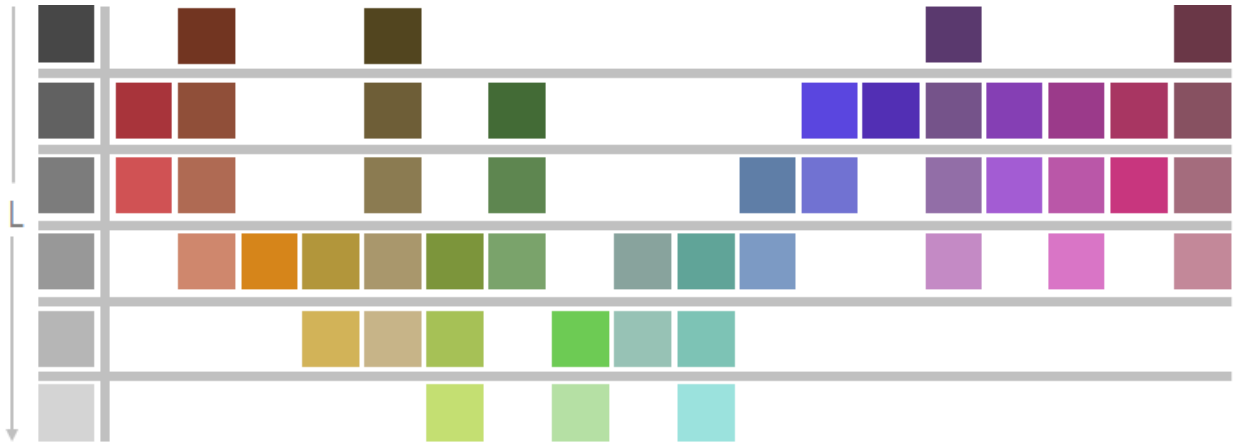


Figure 8.4: The 52 sample as distributed in CIELAB space.

To generate the differed colors, I defined a difference step for each size and sampled ±5 steps per axis. This creates 33 color differences per size, including 3 where the color difference was zero. I included zero difference cases both for completeness and to aid in validation. Each participant saw all 33 color differences a total of 3 times, but with different colors mapped to each color difference step. I distributed the source colors across participants such that we had an equal number of samples for each color × color difference.

Because discriminability is reduced as marks grow smaller, the tested steps were scaled for each size to avoid floor or ceiling effects as sizes changed. For sizes less than 2 degrees, the color difference steps were generated by normalizing Carter and Silverstein's model Carter and Silverstein [2010]—which models changes in color difference as a function of response time—such that a color difference step for the 2-degree square equaled 1ΔE. While the Carter and Silverstein models were based on a different performance metric, they provide a baseline for ensuring that tested color differences were sampled appropriately as the size of the mark decreased. Step sizes were linearly interpolated for sizes not sampled by Carter and Silverstein. For sizes 2 degrees and larger, a uniform step of 1.25ΔE was used.

I collected data using a total of four experiments, each evaluating three size sets: 0.33, 0.67, and 1 degree; 0.5, 1.25, and 2 degree; 2, 4, and 6 degrees, and 0.4, 0.8, and 1.625 degrees. I replicated the 2 degree value because our initial color difference step for 2 degrees of 1ΔE was found to be too small to collect useful data. The results from the larger step were used in the model. In all cases, the stimuli were presented a fixed distance apart (4 degrees) measured edge to edge, assuming a standard viewing distance of 24 inches.

For each experiment, participants first were prompted for their demographic information. Then they were then given a brief tutorial explaining the task at hand. Each participant saw 104 trials (99 experimental observations and 5 validation trials with a very different colors:, $\geqslant 20\Delta E$ difference). There was a 500ms white screen between trials to alleviate adaptation effects.

8.1.3 Statistical Analysis

Overall, I modeled responses from 624 participants (245 female, 339 male, 40 declined to state) between 16 and 66 years of age ($\mu = 33.71$, $\sigma = 11.60$) with self-reported normal or corrected-to-normal vision. Each participant saw each of the 52 stimulus colors twice, and each combination of color difference (difference amount $\times$ direction $\times$ axis) once for all three sizes. Color $\times$ size $\times$ color difference was counterbalanced between participants. This sampling density will predict discriminability rates for each tested color difference to at worst $\pm 7.5\%$ with 90% confidence.

To verify the validity of these results, I ran a 9-level ANCOVA on the responses across all four experiments in the study. Gender was treated as a covariate and interparticipant was modeled as a random factor. Mark size was treated as a between-subjects factor. I found significant effects of age ($F(1, 607) = 8.1342$, $p = .0045$) and question order ($F(1, 50826) = 16.7810$, $p < .0001$); however, I found no systematic variation for either factor. I also saw significant effects of the fixed color's L^* ($F(1, 50791) = 1448.323$, $p < .0001$) and b^* ($F(1, 50764) = 29.9342$, $p < .0001$) values, but not on the fixed color's a^* value ($F(1, 50764) = 0.1621$, $p.6873$); however, only L^* appeared to have a systematic influence on responses—discriminability was slightly better for light colors than for dark. The primary factors—size ($F(10, 6741) = 58.2625$,$p < .0001$) and color difference along L^* ($F(1, 50756) = 8301.816$, $p < .0001$), a^* ($F(1, 50756) = 7819.245$, $p < .0001$), and b^* ($F(1, 50756) = 4974.221$, $p < .0001$)—all had a highly significant effect on response.

8.1.4 Predicting Discriminability Thresholds

The collected data was used to create a parameterized noticeable difference (ND) as a linear function of distance in CIELAB space for each tested size, using the same procedure as discussed in Section 7.5.1. For every tested color difference, a

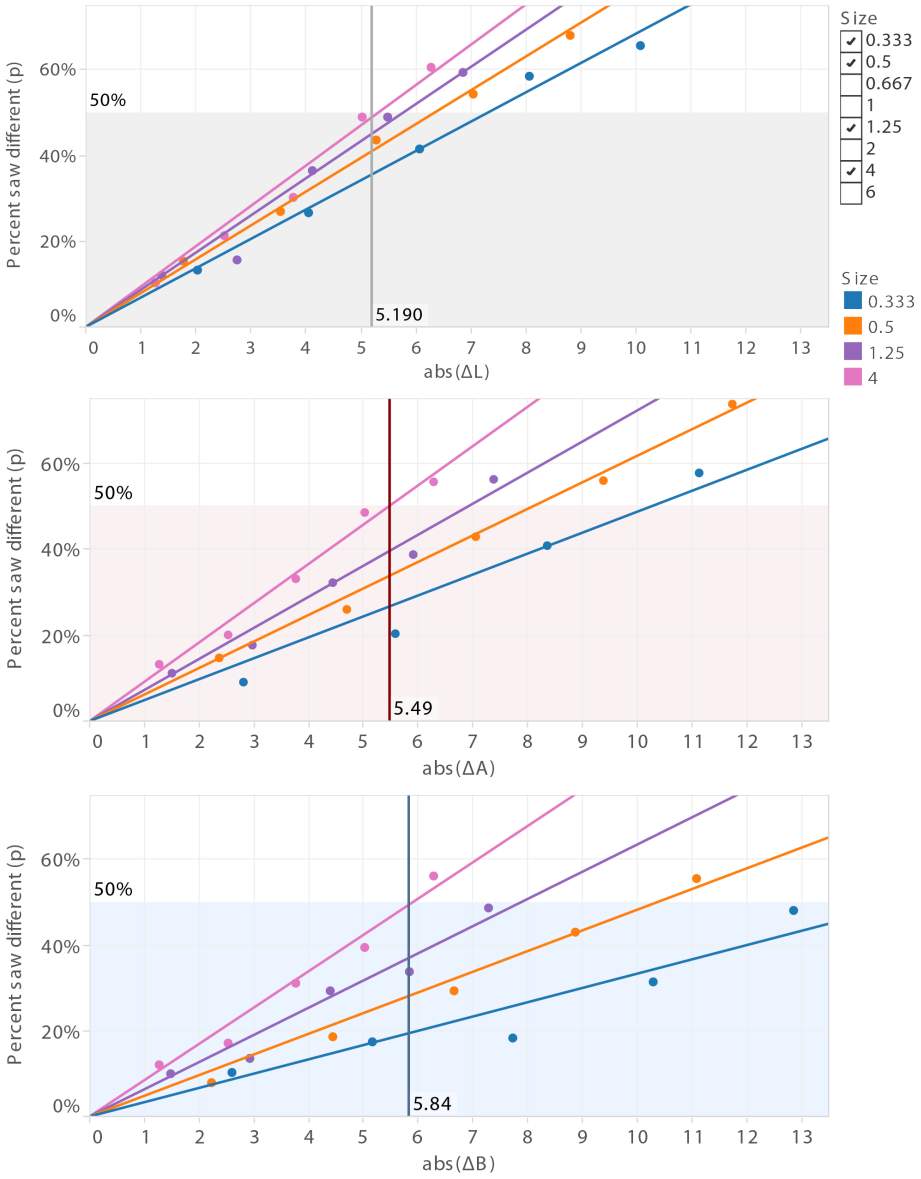


Figure 8.5: Discriminability changes linearly with color difference (colored lines show this fit for four tested sizes), but the slope of the linear fit decreases with size. The shaded box marks 50% discriminability. The point at which each line exceeds this bound is the ND(50) for each of L^*, a^* and b^* axis. The ND(50) for the 4-degree stimulus is indicated by a vertical black line. All models fit with $p < 0.0001$ except for Δb for size 0.33 ($p = 0.000189$).

	Size (s)										
Axis	0.333	0.4	0.5	0.667	0.8	1	1.25	1.625	2	4	6
L^*	0.068	0.069	0.078	0.081	0.090	0.083	0.089	0.085	0.100	0.096	0.090
a^*	0.051	0.054	0.062	0.067	0.064	0.073	0.073	0.072	0.085	0.091	0.097
b^*	0.034	0.042	0.050	0.051	0.055	0.061	0.064	0.066	0.073	0.086	0.086

Table 8.1: $V(s)$ for each size and axis

	Size (s)										
Axis	0.333	0.4	0.5	0.667	0.8	1	1.25	1.625	2	4	6
L^*	7.321	7.267	6.435	6.180	5.531	6.017	5.643	5.903	5.010	5.187	5.574
a^*	9.901	9.268	8.052	7.429	7.837	6.897	6.821	6.906	5.917	5.488	5.149
b^*	14.837	12.019	10.101	9.747	9.091	8.197	7.764	7.587	6.831	5.841	5.834

Table 8.2: ND for $p = 50\%$ for each size and axis

linear model through the origin was fitted to the proportion of correct "different" responses for each color difference. The resulting models were of the form:

$$p = V(s)\dot{\Delta}D + e \tag{8.1}$$

where s is the size, V is the vector of slopes for the linear models on each axis (L^*, a^* , b^*), D is the vector of color differences across each axis, and e is experimental and observational error. This is shown in Figure 8.5. Table 8.1 summarizes the slopes data.

Given Equation 8.1, $ND(p) = p/V$, with ND equivalent to the vector ΔD. For example, to compute the distance vector where 60% of the observers saw a difference, divide 0.6 by V to derive the set of color differences in LAB space that separate two colors with a 60% reliability. The utility of this p value is discussed in Section 7.5. Classically, a JND is defined as color difference where 50% of the observers identify the difference, or $ND(50)$. The collected data generate unique conventional JNDs for each size (Table 8.2). I can use this data to estimate the noticable color difference for a given size, or $ND(p, s)$, in two different ways.

8.1.5 Predicting Discriminability at a Given Probability

Given a fixed p (likelihood that the difference is detectable for the target population), a designer will want to predict $ND(p)$ as a function of size. Figure 8.6 plots $ND(50)$ against size. This figure shows that discriminability varies roughly inversely with

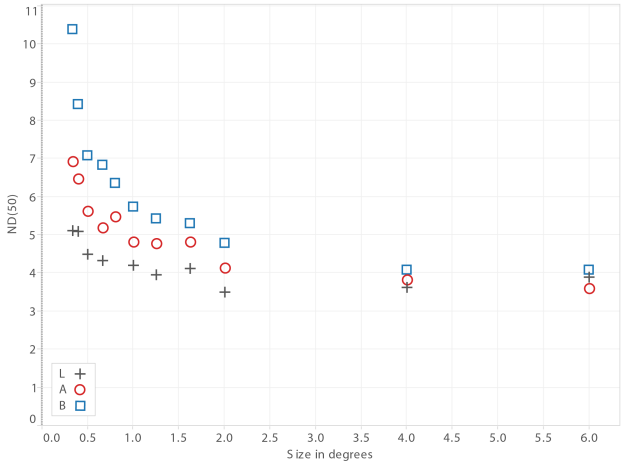


Figure 8.6: ND(50) plotted against size for each of our tested sizes for each axis. L^* is gray plus, a^* is red circle, b^* is blue square.

size (Figure 8.7). The resulting equation for this relationship is:

$$ND(50, s) = C(50) + K(50)/s \tag{8.2}$$

with the coefficients for C(50) (the asymptote ND(50) as size grows infinitely) and K(50) (the rate at which ND(50) increases with inverse size) computed using a linear regression of discriminability as a function of inverse size (Table 8.3 contains these coefficients for $p = 50\%$) .

Axis	C(50)	K(50)
L^*	5.079	0.751
a^*	5.339	1.541
b^*	5.349	2.871

Table 8.3: C and K coefficients for ND(50)

As size increases, the K/s term goes to zero, leaving a constant ND(50) of $(5.1, 5.3, 5.3)$. Visually, this means that JNDs eventually stabalize once marks are sufficiently large. As size decreases below 1, ND(50) increases more rapidly, meaning viewers abilities to distinguish between marks decays substantially as the marks approach a single pixel.

Changing the desired probability of a noticable difference p generates a series of C(p) and K(p) coefficients. This provides a two-step model for discriminability as a function of size. First, compute ND(p) for the desired p, then use linear regression to define the coefficients for size. To summarize, the model derived in

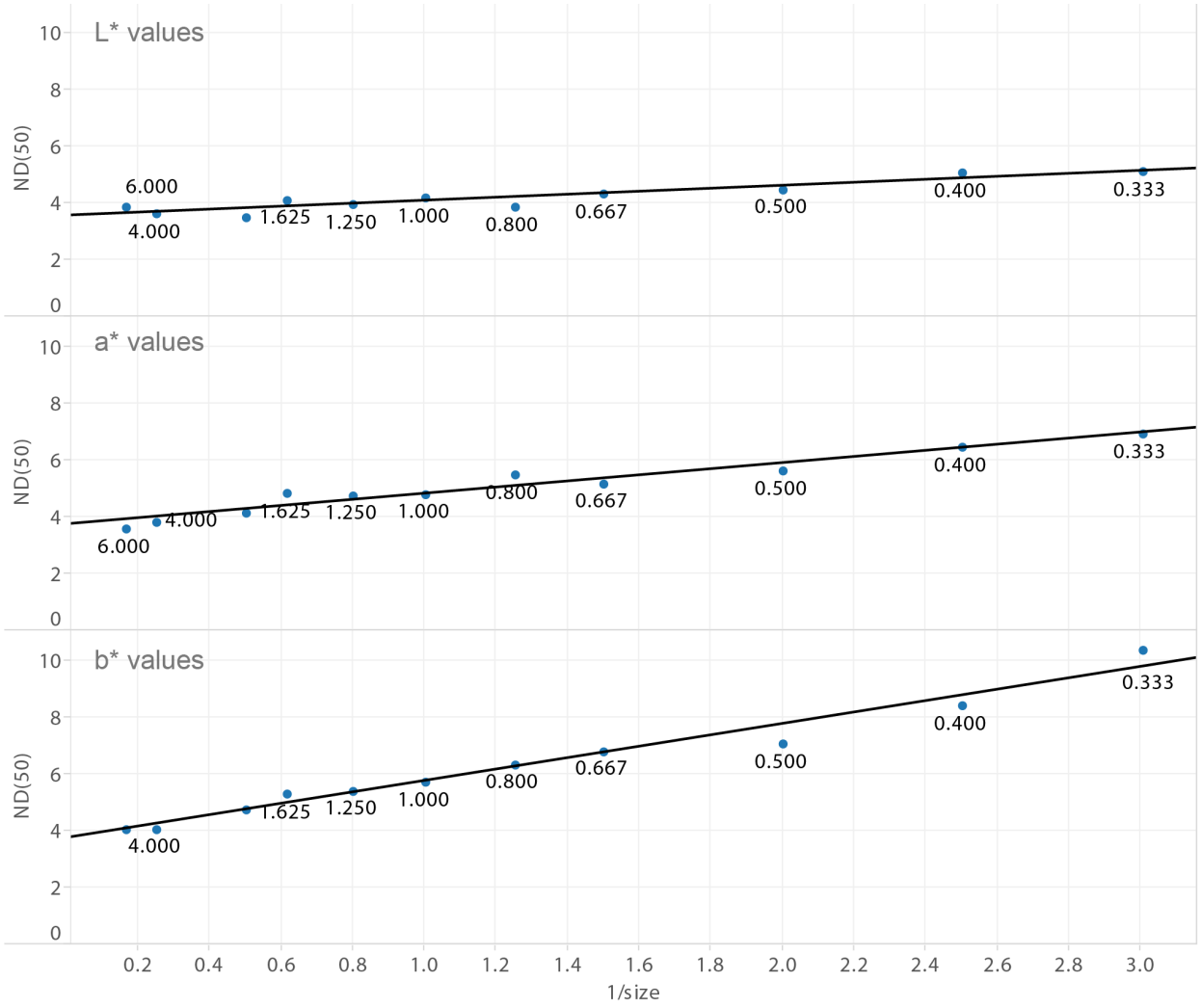


Figure 8.7: The plot of ND(50) for each of the 11 sizes vs. $1/size$ for each of L^*, a^* and b^*. ($R^2_L = .849696, p_L < 0.0001$; $R^2_a = .942234, p_L < 0.0001$; $R^2_b = .970395, p_b < 0.0001$)

this section shows that a p% just noticeable difference (ND(p)) for a fixed value of p varies with size (s) according to:

$$ND(p, s) = C(p) + K(p)/s \tag{8.3}$$

8.1.6 Generalizing the Model to Size

The previous model captures discriminability for a fixed probability of robustness. Designers may alternatively about models that vary in p over a fixed mark size. Based on the results in the previous sections, the solution will have the form:

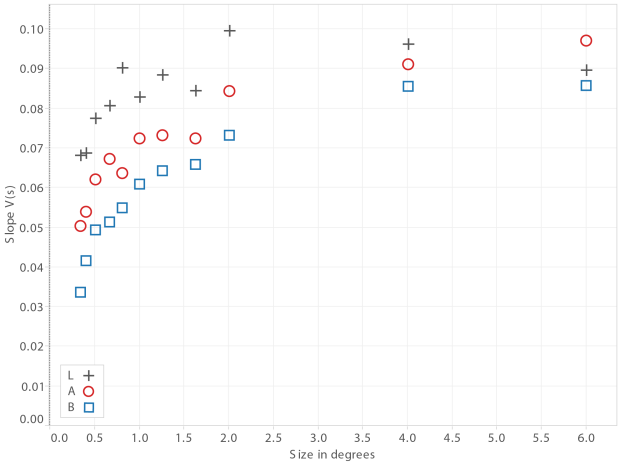


Figure 8.8: The distribution of the slope, V vs. size for our data. Gray cross is L^*, red circle is a^*, blue square is b^*.

$$V(s) = p/ND(p) = p/(C(p) + K(p)/s) \tag{8.4}$$

where $C(p)$ and $K(p)$ are the coefficients in Equation 8.3.

Plotting slope, V, as a function of size gives a non-linear distribution (Fig. 8.8) where the inverse slope varies inversely with size (Fig. 8.9). The resulting model has the form:

$$1/V(s) = A + B/s \tag{8.5}$$

which yields a general formula for $ND(p, s)$:

$$ND(p, s) = p(A + B/s) \tag{8.6}$$

where s is size in degrees, p is the probability of a detectable difference ($[0,1]$), and the values for A and B are shown in Table 8.4.

Axis	A	B
L^*	10.16	1.50
a^*	10.68	3.08
b^*	10.70	5.74

Table 8.4: A and B coefficients for Equation 8.5

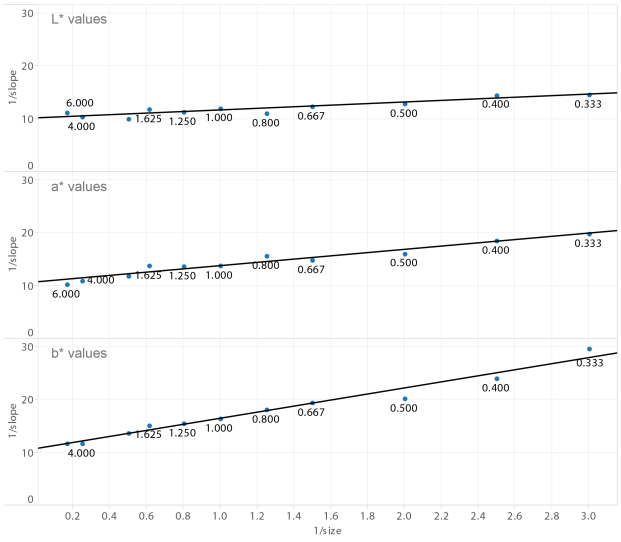


Figure 8.9: Linear fit to $1/V$ vs $1/size$ for each of L^*, a^* and b^*. ($R^2_L = .849696, p_L < 0.0001$; $R^2_a = .942234, p_L < 0.0001$; $R^2_b = .970395, p_b < 0.0001$).

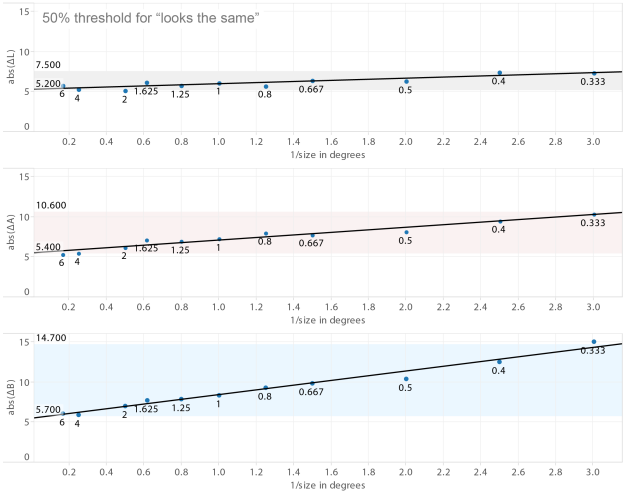


Figure 8.10: The figure shows the color difference step needed for 50% discriminability (ND(50)) for each axis as a linear model of $1/size$. Colored bands are labeled with the range of color difference values for each axis.

8.1.7 Discussion

The models generated in this section help designers reason about discriminability as mark sizes change. For example, Figure 8.10 shows the different $ND(p, s)$ regression lines for $p = 50$ for each axes. The shaded bands show the variation in ΔL^*, Δa^* and Δb^* over the range of sizes, with the band size increasing for L^* vs. a^* vs. b^*. Each of these models generates a single scaling parameter based on the size of a mark s and the desired discriminability level p.

Part of the challenge with designing for marks of difference sizes is that not only discriminability but overall appearance changes as colors get small—small stimuli appear less colorful. However, in practice, the models provided here allow designers to, at a minimum, ensure sufficient discriminability in encoding design. For example, in Figure 8.11, both the large and small patches are stepped according to the parameters of the current size model. Ideally, the color differences will seem the same independent of size. For comparison, the small patches are also shown with the same color steps as the large patches, and should appear less different.

8.1.8 Conclusion and Future Work

The work presented in this section offers a simple model for computing color difference as a function of size. While these results are preliminary, this sort of data-driven modeling shows strong promise for creating practical results. The data indicates that a minimum step in CIELAB of between 5 and 6 is needed to make two colors visibly different for large shapes (2-degree or larger), which matches well with the intuitions that my collaborators have developed through design practice. The asymmetry between L^* and the two axes that together define hue and chroma (a^* and b^*) also matches designer experiences (e.g. Samsel et al. [2015]).

In practice, these models are somewhat limited by their bind to symmetric marks: all studies were conducted using colored square patches. Future work will include studies to refine the model parameters, including a consideration of non-symmetric marks, and to explore the effect of background color on these judgments.

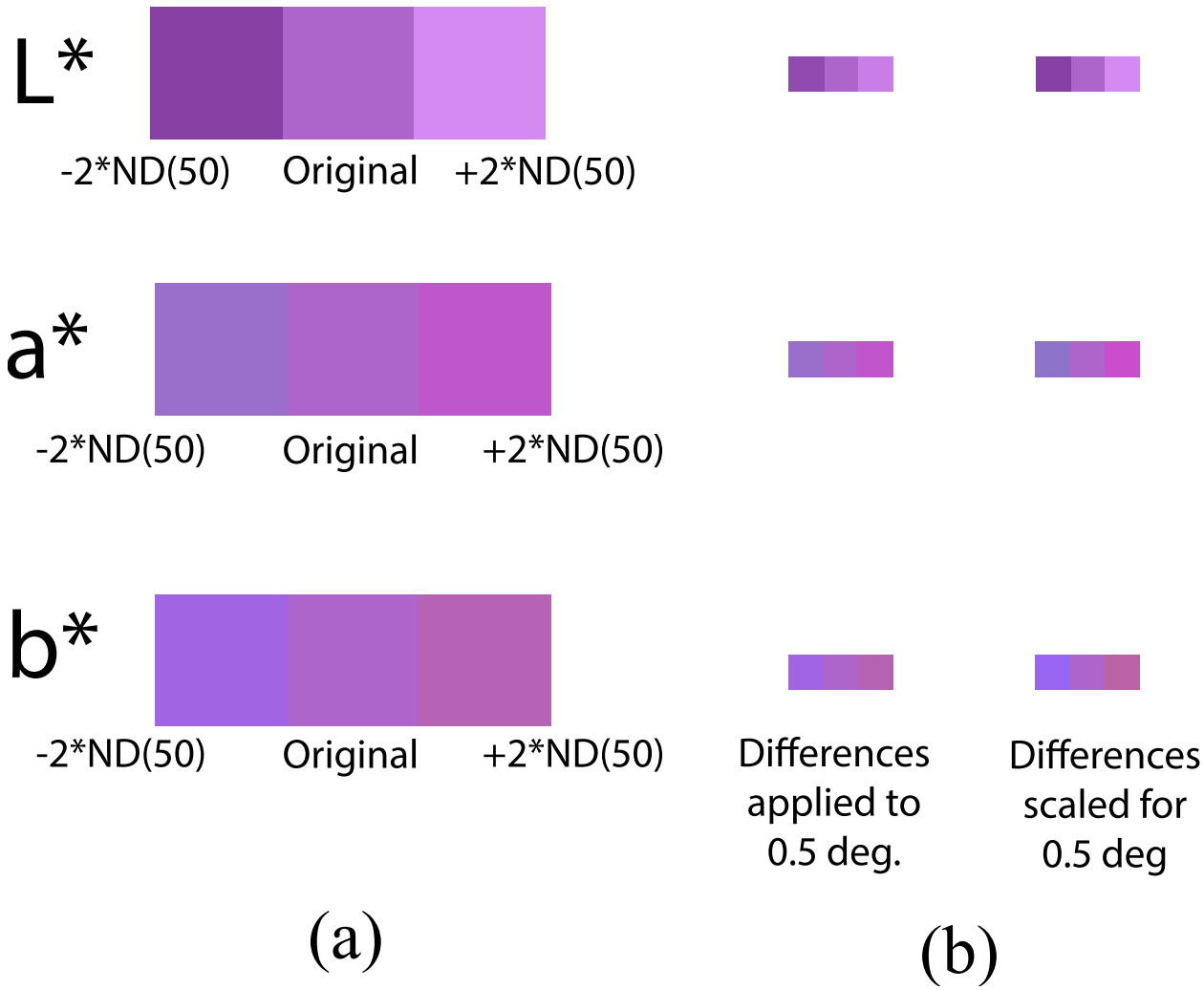


Figure 8.11: Assuming a viewing distance of 24 inches, the (a) large patches are 2 degree squares and the (b) small patches are 0.5 degrees. Horizontally adjusted patches are two ND(50) steps different as computed from the model formulas. For comparison, the 0.5 degree squares are also drawn with the 2-degree values. The differences are subtle, but important: the 2 degree color differences become more difficult to see when applied to smaller marks. Scaling color differences according to size results in color differences at 0.5 degrees that better match the differences for 2 degrees.

8.2 Color for Elongated Marks

The color-size model from this chapter provides metrics for creating color encodings for small marks of a uniform height and width. By designing for the smallest expected mark, the metric gives designers a lower-bound on how discriminable their encodings will be. In practice, marks often are not uniform: visualizations can use the length of a mark to encode data. For example, blocks within a tree map may vary in width and height. The length and height of scatterplot points can also be used to encode values.

In visualization, designers often constrain the valid range for the length and width of a mark to prevent marks from becoming too small or too large. Designers can use these constraints to create encodings that are robust for the *smallest* possible mark size they expect to occur (e.g. the lower-bound on a size encoding). Color encodings that are discriminable for the smallest allowed dimension of a mark will become more discriminable for larger marks—data mapped to different values in a color encoding will only become more discriminable as size along any given axis increases.

For some visualizations, designers may not be able to fully constrain the size of a mark. For example, bar charts vary the height of a bar to encode value, but this height is difficult to constrain—small values map to small heights. A designer does not always know ahead of time what length the bar will be, but constraints are often imposed on the width of a bar. Previous work suggests that increasing the length of two rectangles of different colors makes it easier to tell a difference between them, and that elongation, not area, is the primary factor in efficiently distinguishing between marks [Highnote, 2003]. This means that for cases where designing for the smallest dimension is not possible, instead designing for a known dimension can provide some guarantees of robustness—the longer dimension will be a primary predictor of discriminablity. Therefore, color encodings based on the minimum width of a bar will remain distinguishable for the majority of cases (below a certain ratio of side lengths, gains from elongation level out). If the height of the bar is smaller than the width, the width becomes the elongated size and should play a significant role in preserving discriminability.

However, this prior work measures how quickly different sets of marks with large color differences could be identified. In visualization, designers often care about encoding subtle changes between values using color, especially for large amounts of ordinal data, and analysts often have unlimited time to view a visualization.

The color-size model explains how size might influence color under a different measure (small color differences modeled based on accuracy).

In this section, I present an experiment verifying that the color-size model is robust for changes in bar height. In this study, I do not look to model how discriminability changes with height—the height of a bar is often unknown to a visualization designer *a priori*—but to show how modeling discriminability with respect to width can help ensure discriminability.

8.2.1 Methods

Discriminability for bars was measured using two experiments—one for each of two different bar widths—using the same methods as in Section 7.5. Participants saw bars of either 0.5 or 1 degree of visual angle. Bars were placed four degrees apart among mid-grey distractor bars of random heights (Fig. 8.13). Colors were drawn form 14 colors sampled uniformly across L^*, a^*, and b^*. Tested color differences were 0.25, 0.5, 0.75, 1, 1.25, and 2 $ND(50, w)$ units apart, where w corresponds to the width of the bars. Tested bar heights were 0.5, 1, 3, 5, and 7 degrees.

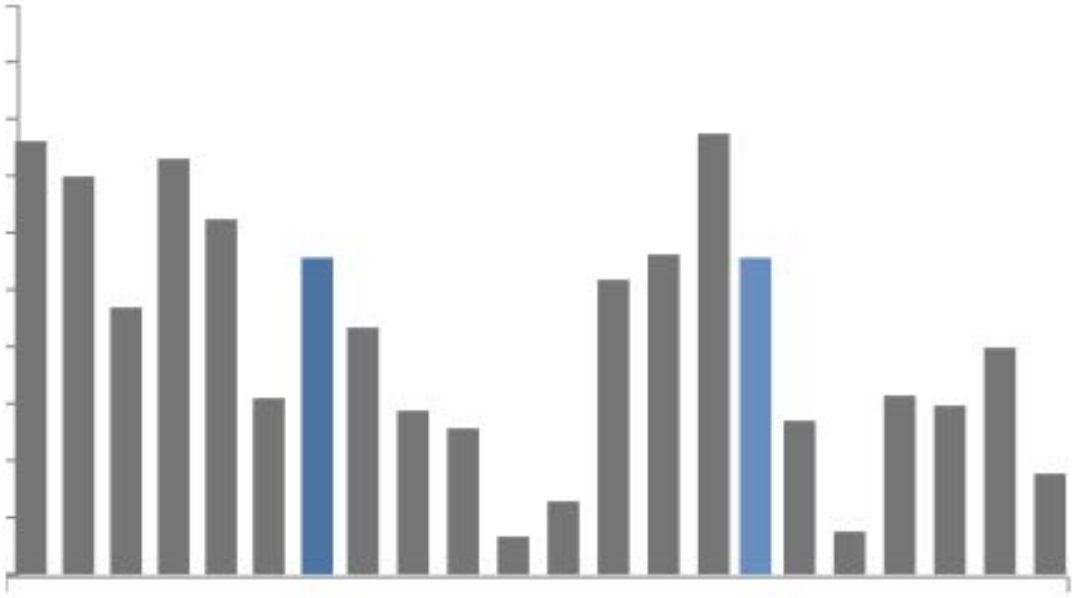
After completing three training stimuli to introduce participants to the definition of "same" and "different" colors, participants reported whether they perceived a color difference for 98 bar charts. Each participant saw each tested color six times and each combination of height $\times$ color difference three times. Three stimuli were mapped to large color differences and five to identical colors (one per tested height) for validation.

I analyzed performance for 140 participants (70 per condition; 53 female, 87 male; mean age 32.32, σ 9.33) of an original sample from 152 participants. Three were excluded for performance on the "very different" stimuli, and nine for performance on the equal color bars.

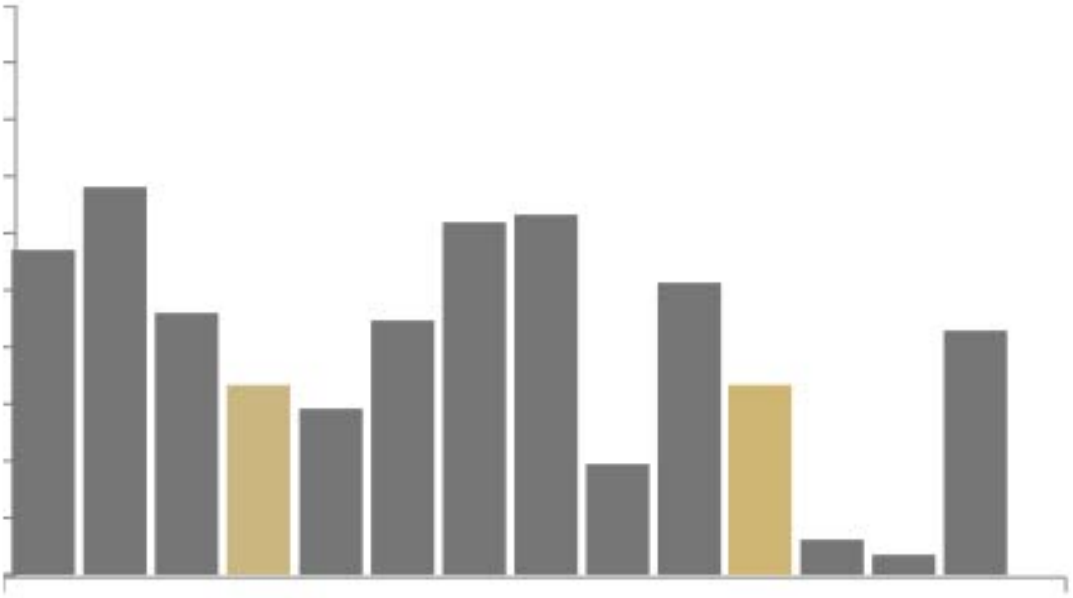
8.2.2 Results

I analyzed performance for each width condition using a three-way Chi Squared Test (height, the $L*$ of the color, and the color difference in ΔE) comparing participant responses. For both widths, discriminability increased with bar height ($\chi^2_1(1, N = 70) = 151.08, p_1 < .0001$, $\chi^2_{0.5}(1, N = 70) = 94.95, p_{0.5} < .0001$).

Since color discriminability (and tested step size) both vary with bar width, I compared performance across both conditions using a Chi-Squared with the



(a) Bars 0.5 degrees wide.



(b) Bars 1 degree wide.

Figure 8.12: Participants were asked to report whether or not two bars were the same color. Bars were placed four degrees of visual angle apart, and surrounded by mid-grey distractor bars to increase task validity.

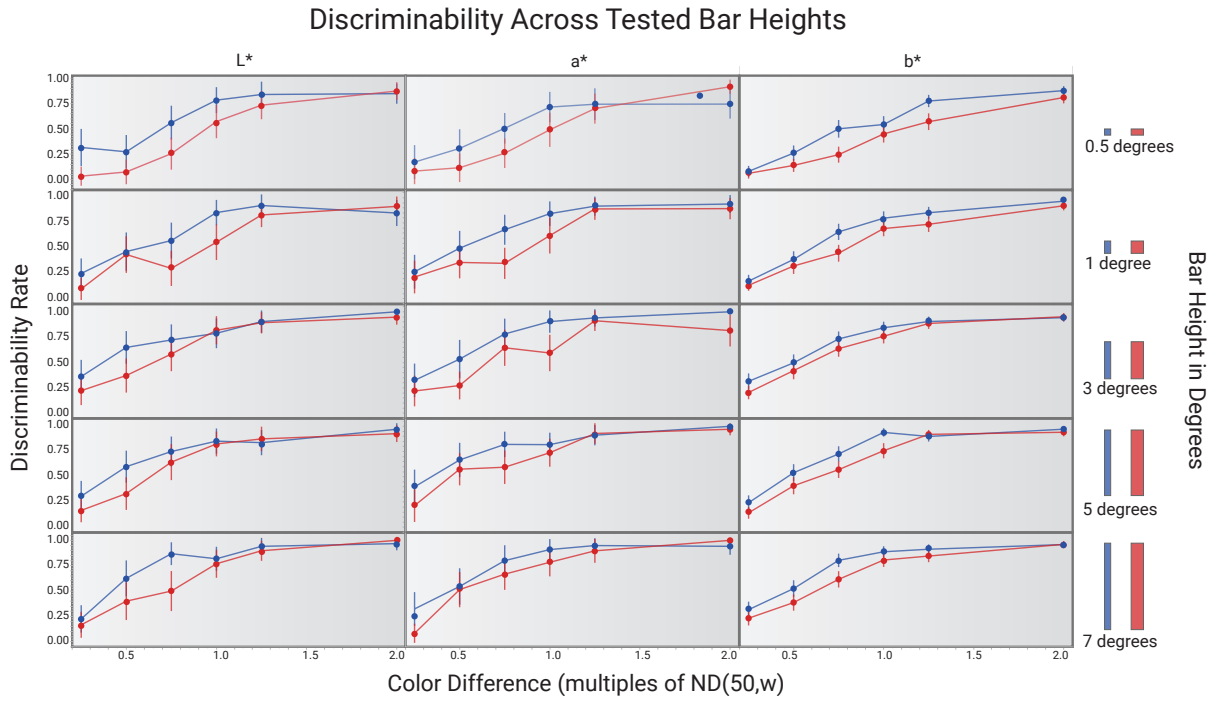


Figure 8.13: Breaking down performance by height and width shows comparable color matching performance for both 0.5 and 1.0 degree bars. Discriminability changed at roughly the same rate for both widths across all heights. Relative heights are shown at the right for comparison (height and width are scaled down approximately 75% to fit within the page). Error bars present 95% confidence intervals.

color difference expressed as multiples of a JND (e.g. a difference of .25 for an absolute color difference of .25 * 1 JND). I found a highly significant effect of width ($\chi^2(1, N = 140) = 153.78, p < .0001$), height ($\chi^2(1, N = 140) = 258.49, p < .0001$), L* ($\chi^2(1, N = 140) = 166.84, p < .0001$, Fig. 8.14), and color difference ($\chi^2(1, N = 140) = 3779.63, p < 0.0001$). I also found a significant interaction effect between color difference and height ($\chi^2(1, N = 140) = 24.86, p < .0001$) but not between color difference and width ($\chi^2(1, N = 140) = .2068, p = .6493$) or between height and width ($\chi^2(1, N = 140) = .0308, p = .8606$).

For both widths, discriminability significantly increased with height (Fig. 8.14). This suggests that as marks grow longer, discriminability increases. An alternative explanation of these results is that discriminability increases with area as area is correlated with height. While height was a significant predictor of performance, the area of a mark (height * width) did not appear to have a systematic effect on performance (Fig. 8.15). These results align well prior work [Highnote, 2003]: the longest edge of a bar is more important for discriminability than the area. This

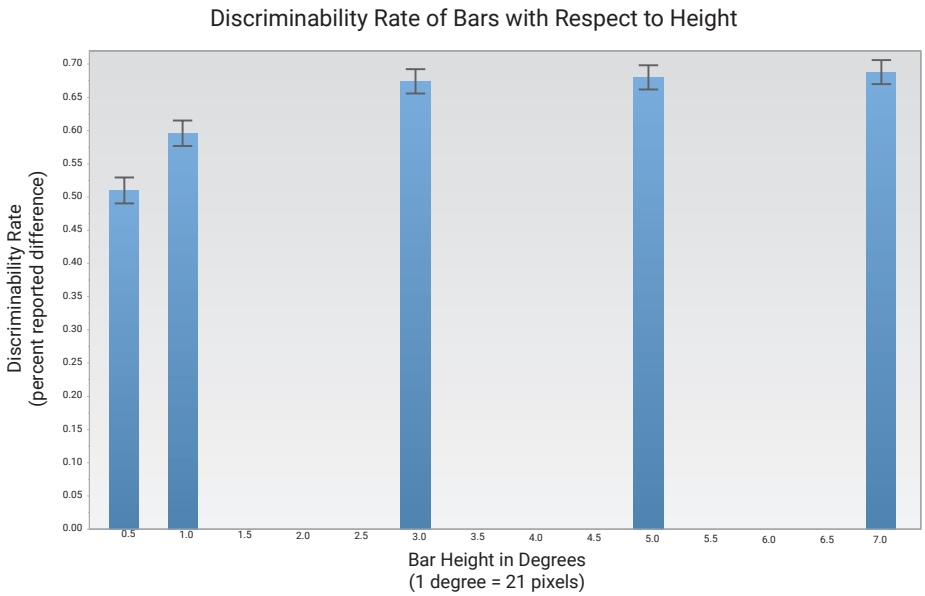


Figure 8.14: Detecting color differences between bars becomes easier as bars grow taller. This increased discriminability for longer bars appears to be limited: discriminability gains appear to level off for the larger tested heights.

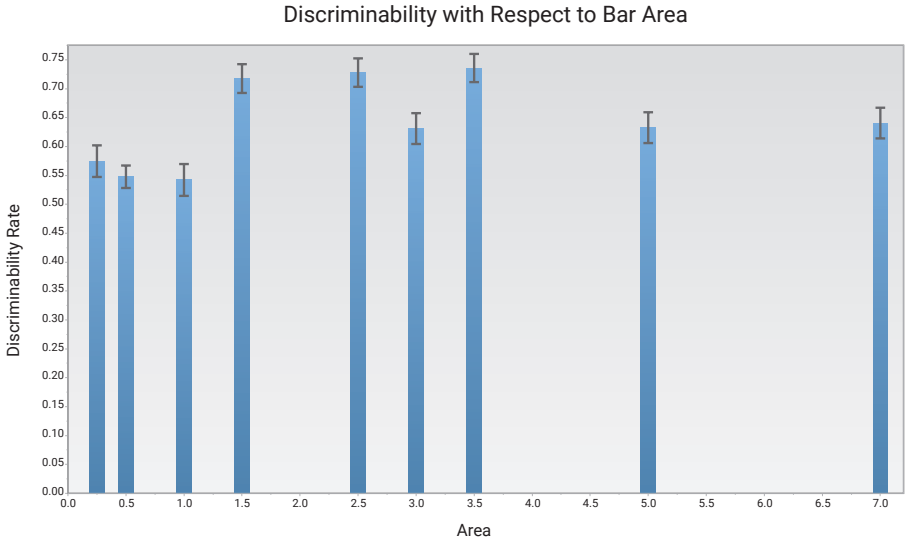


Figure 8.15: Changes in height are correlated with changes in area. However, area does not appear to systematically effect the apparent color differences between bars across the tested bar widths. This result confirms results from prior work: elongation is more important to discriminability than area.

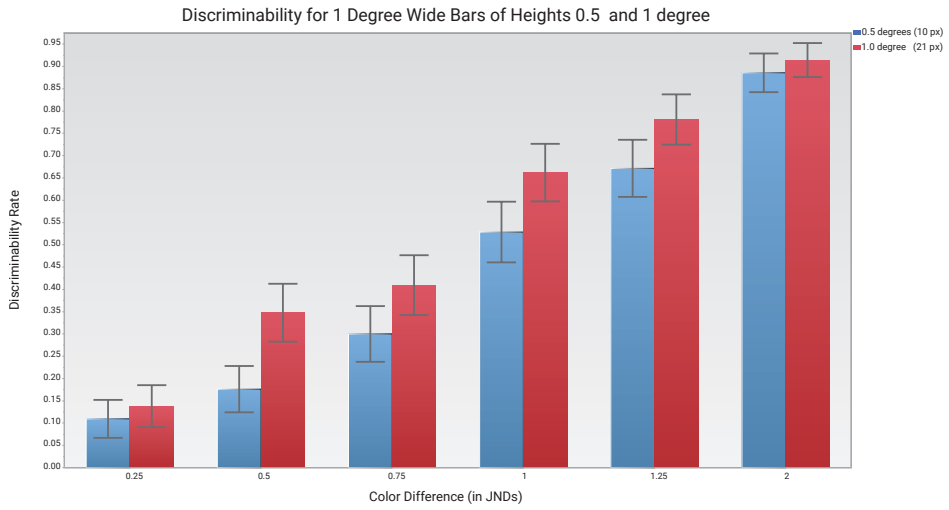


Figure 8.16: Performance at different color differences for short (1 × 0.5 bars, blue) and the equivalent square bars (1 × 1, red). The only significant difference in performance in for 0.5 JNDs. This suggests that discriminability based on length is reasonably robust for the shorter bar. This performance likely breaks down in extreme cases (e.g. bars of one or two pixels). Future work might explore discriminability as a function of the ratio of bar height to width.

also correlates with findings from the color-size model: discriminability varied inversely with length as opposed to area (length * length).

In one tested condition, the height of the bar was smaller than the width (1 degree width, 0.5 degree height, upper-left corner of Fig. 8.13). For this condition, I only found significant performance differences between the uniform mark size (1 × 1) and the shorter mark size (1 × 0.5) for a color difference of 0.5 JNDs (Fig. 8.16). This finding provides preliminary evidence that designing for known constraints in a visualization, like the minimum allowable bar width, can guide effective encoding design. The shorter bar had half the area but an equally large longest side and could be distinguished roughly as effectively. However, there are known limitations on this elongation effect: discriminability degrades significantly once the height is a fixed proportion smaller than the width [Highnote, 2003]. Better understanding this ratio will allow designers to better reason about the robustness of encodings developed with this model.

8.2.3 Discussion

These results provide initial evidence that robust color encodings for bar charts can be designed by considering only the width of a bar. These results show that as

a bar increases in height, discriminability increases. Area does not appear to be the primary factor in determining discriminability. Instead, in line with previous research, discriminability increases more closely with elongation.

This study provides preliminary evidence of how robust color encodings can be applied to elongated marks. It does so using only two sizes and a fixed number of heights. Further, in only one condition is the bar shorter than it is wide. As marks become very short (on the order of one or two pixels), discriminability is likely to degrade substantially regardless of the width of a mark. It is unclear, however, whether colors can be discerned with much fidelity at all at extremely small scales—the necessary color differences may be too large to effectively design for.

Future work should build on this study by exploring a wider range of widths, heights, and aspect ratios. This study could provide more robust guidance for designing color encodings as a function of size and could lead to new models of color difference for specific mark types.

8.3 Applications to Color Encoding Design

The experiments in this chapter show that size is an important parameter for designing effective color encodings. The resulting models can be used to design encodings that are robust at a certain size or evaluate how suitable existing encodings for different visualization designs. In this section, I briefly discuss how these results might be used to inform color encoding design for visualization to provide color encodings with uniform perceptual differences, to adjust encoding designs for different mark sizes, to tune encodings to support special values, and to design systems for tailoring encodings to a visualization.

8.3.1 Guiding Effective Color Encodings for Different Mark Sizes

The metrics introduced in this dissertation allow designers to create color encodings where the differences between each step in an encoding are perceptually uniform. For example, viewers can identify *control colors* in a sequential color ramp that represent the first and last steps in an encoding. The intermediate steps are computed by uniformly interpolating colors between these two points. Interpolating control colors in CIELAB ensures that the color differences between

(a) ColorBrewer Greens

(b) ColorBrewer Yellow-Green-Blue

Figure 8.17: Allowing designers control over ramps is a trade-off. Perceptually-uniform ramps can be interpolated between desired colors in CIELAB, but this flexibility requires designers to select visually pleasing combinations. Designers can use existing hand-generated ramps as guides for creating new encodings. For example, ColorBrewer ramps use small color shifts to introduce visual appeal. Cylindrical interpolation in CIELAB can produce encodings with uniform perceived distances but are less visually appealing. Allowing designers to introduce small shifts to the center color of a ramp and interpolating uniformly along the resulting curve can significantly enhance the visual appeal of an encoding.

any two subsequent steps are approximately perceptually equivalent. If designers know, for example, the minimum mark size for their visualization, they can use the color-size model presented in this chapter to pick step sizes that will be effective for their target visualization. They can also use the scaling parameters presented in these models to normalize color interpolation along L^*, a^*, and b^* to account for known variations in color perceptions between axes.

Interpolating in CIELAB is also beneficial as it allows designers to create perceptually smooth color encodings without extensive design expertise [Wijffelaars et al., 2008]. The metrics provided here can guide how finely the interpolation should sample CIELAB for steps to be sufficiently distinct. For example, visualizations commonly use ramps from ColorBrewer [Brewer et al., 2003b] to encode data.

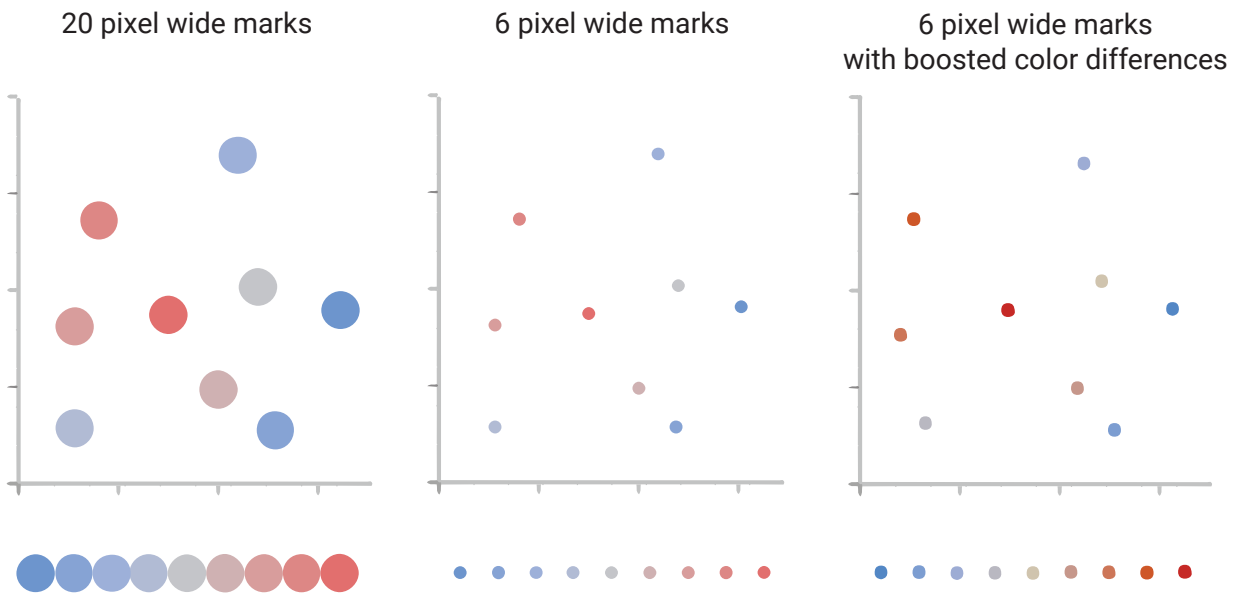


Figure 8.18: The color-size models presented in this chapter can be used to design encodings that are robust to mark sizes. For example, color encoding designed for 20 pixel wide marks (left) becomes more difficult to distinguish when it is applied to six pixel wide marks (center). By scaling differences in the original encoding using this model, the endpoints of the encoding are pushed further apart at smaller scales (right) to better match the perceived differences in encodings designed for larger marks and better support point tasks.

These ramps are visually appealing and used extensively in practice, but because they have been constructed by hand, designers using them must pick from a set of existing ramps. However, designers might need to generate encodings using specific colors not included in this set for several possible reasons, including aesthetic or semantic characteristics of the data [Lin et al., 2013]. Designers can interpolate between any two desired colors in CIELAB to generate a reasonable color encoding without the expertise required to handcraft effective encodings.

Handcrafted ramps can then be used for design inspiration. For example, ColorBrewer ramps often introduce small nonlinear variations between endpoints to improve visual appeal. These variations can be modeled in CIELAB by introducing third control color near the center of the ramp. By displacing the center color and interpolating along the curve through the three control colors (start, middle, and end), interpolated encodings can closely approximate ColorBrewer (Fig. 8.17).

The metrics generated in this dissertation help designers reason about the distances with which control points can be interpolated to remain effective for a

Figure 8.19: Color steps that are readily discriminable can be used encode special values within a visualization. The results from Section 7.7 can be used to generate encodings to represent outliers. For example, the crowdsourced ramp in Figure 1.4 encodes values using luminance. Outlier colors could be generated by extending the luminance interpolation on either end of this ramp by one readily discriminable knee step (top, first and last color). Adding an equivalent hue step in either direction (bottom, first and last color) generates outlier colors that appear even more distinct from the primary encoding but still preserve the perceived order of outlier values.

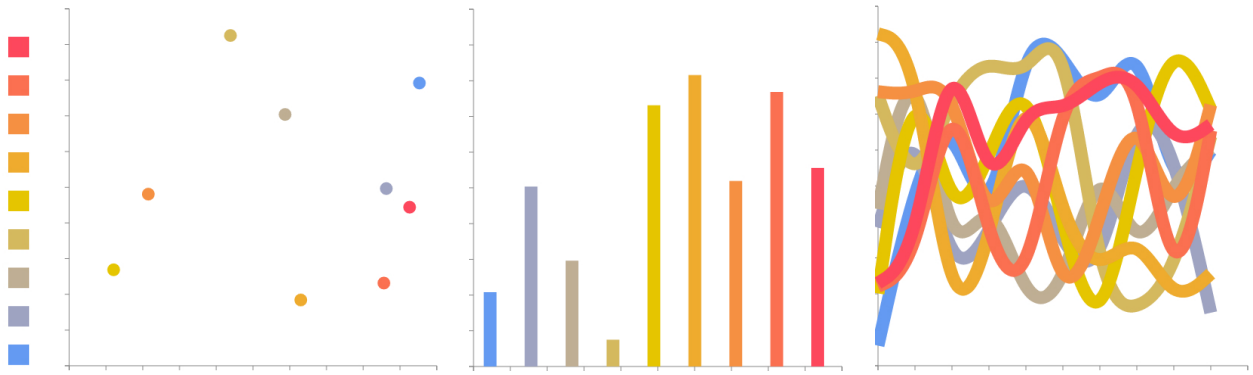
target design. Designers can use the color-size model discussed in this chapter to tailor encodings to anticipated mark sizes in two primary ways. First, designers can specify a mark size before selecting control colors. The number of possible steps in an encoding is then bounded by the number of sufficiently distinct steps between the control colors. Alternatively, designers can first create a color encoding and then adjust the control colors to scale perceived differences between values. This approach pushes colors apart or pulls them together to preserve the perceived differences between steps in an encoding as marks change in size. The differences between steps in the encoding can be scaled by:

$$\Delta E_{new} = \Delta E_{old} * \frac{ND(50, s_{new})}{ND(50, s_{old}))} \tag{8.7}$$

This reduces color differences as marks grow and boosts differences as marks shrink (Fig. 8.18). The new colors can be iteratively refined by manipulating the new control points of the encoding. The resulting encodings are probabilistically robust for a desired mark size: color differences that encode value are tuned to the visualization design to support point visualization tasks.

8.3.2 Designing Outlier Colors

Color encodings may also need to adapt to the data itself. For example, the distribution of a dataset can impact how effectively an encoding represents data. Extreme outliers or long-tailed distributions can bias color encodings—a small amount



(a) (b) A scatterplot with marks of a uniform size.

(c) A bar chart where the width of each bar matches the diameter of a scatterplot point.

(d) A line graph where line width matches the diameter of a scatterplot point.

Figure 8.20: Effective color encodings should be tailored to mark of different shapes and sizes. For example, colors that are useful for (b) scatterplots appear brighter for (c) bars of equal width. The encoding breaks down when mapped to (d) intersecting lines in a line graph. The work presented in this chapter provides first steps in understanding how designers can tailor color encodings to support different visualization designs.

of data consumes a large proportion of the range of an encoding. Visualizations can try to tailor how colors map to data value to account for these distributions [Tominski et al., 2008], but this requires knowing the data distribution in advance. It also requires the viewer to mentally compare values along less intuitive scales [Aigner et al., 2011]. In practice, analysts may instead want to bin these values and represent them using a single outlier color.

Encodings can be designed to include outlier colors to represent binned outliers. Outlier colors should be appear readily distinct from the rest of the encoding. This ensures that outliers will not be confused with data values in the primary encoding. Outlier colors should also be perceptually ordered with respect to the primary encoding: high value outliers should be perceived as higher than other values. Figure 8.19 shows how the "knee" metric (Section 7.7) can be used to generate effective outlier colors for quantitative data. Scaling knee distances for different mark sizes can ensure these colors remain readily discriminability in practice.

8.3.3 Considering Visualization Designs

Color encodings might need to account for size differently across mark types. For example, Section 8.2 shows how encodings for bar charts can be designed for bar widths, whereas scatterplots might instead be constrained by the smallest allowed mark size. Aesthetic concerns may also vary across mark size. For example, the colors in Figure 8.1 appear bright and saturate in bar charts but appear softer when mapped to a scatterplot. Likewise, the same color choice may be inappropriate for other mark types (Fig. 8.20). In practice, color encodings that are good for small marks are often bright and saturate, whereas visually appealing colors for large marks should generally be more subdued [Munzner, 2014].

Both visual appeal and discriminability are important for effective encoding design [Cawthon and Moere, 2007]. To better understand visual appeal and discriminability in practice, encoding design tools should provide designers with the tools to account for these properties when developing an encoding. For example, systems could tailor color encodings for web-based visualizations by allowing the designer to specify a set of desired control colors, the desired discriminability level (p) and the anticipated mark size (s). These parameters serve as inputs to the color-size metric which, as discussed above, can be used to generate an appropriate color encoding. The encoding can then be previewed using different mark types, such as points, bars, or lines, to refine the aesthetic properties of the encoding for a target visualization design (Fig. 8.20). These metrics allow systems to put full aesthetic control in the hands of the designer while providing probabilistic guarantees of how accurately encodings will be interpreted.

8.4 Discussion

This chapter discusses a model for determining color difference as a function of mark size. It also shows how this model can support effective encoding design for variable size marks. Designers can use these metrics in a number of different ways to construct color encodings that are robust to the mark sizes used in a visualization and better support point tasks.

This model assumes that to goal of good encoding design is to ensure data mapped to different values are sufficiently distinct. This choice is a trade-off: increasing discriminability in a color encoding increases the distance between colors. As there are only a finite number of pixel colors, this reduces the total

number of colors available to represent data. In some cases, a designer might prefer to map data to a larger number of less distinct colors.

The color-size model supports this trade-off by being probabilistic—the designer controls how important discriminability is using the p parameter. Higher p values privilege discriminability, whereas lower p values provide more usable colors. While this approach provides the designer with control over encoding design, it is unclear what settings of p are optimal. Better understanding of how this parameter can guide effective design is important future work.

This chapter also only considers two kinds of marks (squares and bars). Other mark shapes might introduce interesting variations. For example, lines tend to be much thinner than bars and also often curve. Wedges in a pie often have a known length, but an unusual shape. Measuring discriminability for these marks can guide effective color encoding design across different visualization types. For visualizations where sizes vary between marks, such as bar charts, changes in mark size influence the apparent difference between marks in the same visualization. For example, the color difference between two small bars might appear smaller than the same difference mapped to two large bars. Accounting for this phenomena would require tuning color encodings whenever a visualization is rendered; however, there are no known models that capture this phenomena. I do not anticipate that variable mark sizes will significantly hinder the utility of these models in practice, but verifying this is important future work.

The results in this chapter, as well as the other chapter in this half of the dissertation, focus on quantitative encodings, where visualizations present ordered data. The color-size model could also be used to improve encoding designs for categorical data, as in Figure 8.1. In categorical data, colors help viewers discriminate between different groups. Effective encodings for categorical data follow different rules than quantitative data [Brewer, 2006]. Unlike quantitative encodings, simply "pushing" color differences apart as categorical marks grow smaller might not be sufficient—because encodings are unordered, it is unclear in which direction color differences should be scaled. Understanding how to use these models to refine categorical encodings is important future work.

9 DISCUSSION

Color has a long history of use in visualization [Bertin, 1983]. Graphical perception suggests that color is of limited utility for tasks that require the viewer to estimate precise values [Cleveland and McGill, 1984]. As the amount of data being visualized increases, visualizations must consciously consider how viewers can use these displays to estimate high-level aggregate properties of the data effectively while still supporting analysts in understanding individual data values. Color has many desirable properties for such scalable visual aggregation. This dissertation shows how designers can leverage color effectively to provide high-level overviews of data, and can mitigate some of the limitations of color at lower levels through careful encoding design.

To understand how color might support visualization and visual aggregation as datasets grow in size and complexity, I have organized and applied findings from perception to help identify potential limitations in traditional approaches to one-dimensional data visualization. This organization illustrates the potential benefits of color over traditional encodings for visual aggregation tasks. At scale, color is limited by the number of available pixels and the perceptual processing power available to make sense of those pixels. To address this limitation, I introduce a method for task-driven aggregation of one-dimensional data that can help designers visualize longer data sequences while preserving local control. This method also introduces designs for encoding aggregate data to support different visual aggregation tasks as datasets scale beyond the available pixels.

To evaluate these theories and designs for visualization, I empirically measured how eight relevant aggregate encodings for time series data support visual aggregation. This evaluation explored performance with respect to what aspects of the data are visualized, how data are visually represented, and how data are mapped to that representation. In these experiments, I showed that these three properties of creating a visualization can predict performance for different kinds of visualization task. Most critically, it demonstrated that color better supports viewers perform tasks that derive new values from a collection of datapoints, whereas position better supports viewers in identifying important values in the distribution.

I demonstrated the scalability of these methods in three real world applications: sequence comparison, text analysis, and structural biology. The resulting systems supported aggregate analyses at scales significantly larger than previous systems.

In each case, domain experts were able to use these systems to generate new insights into their data. I hope that these designs will inspire future work not only in considering how to support visual aggregation at scale, but also in reconsidering the utility of color in visualization.

Visual aggregation is likely to become increasingly important as datasets increase in both size and complexity, but visualizations still must support point tasks in order to be effective for real world analyses. One way to support these tasks is to design other aspects of a visualization to better support point-level color interpretations. For example, in surface visualization, the structure of a surface is important to understanding data in context, but shading used to convey surface depth darkens encoded data. Through a series of experiments, I show how careful visualization design can improve (or impair) color identification performance for surface visualization.

Another way to improve performance for point tasks is to design encodings that map data to distinct colors. Distinct colors can help viewers select for interesting values or compare value differences between individual datapoints. Visualization designers generally rely on metrics to understand how different colors must be to appear distinct. Most of these metrics are based on findings from perceptual psychology which were designed for a different purpose—to model the sensitivities of the eye—and therefore are often impractical for visualization. I introduce a method for deriving color encodings under more practical conditions. Using this method, visualization designers can derive probabilistic models of color difference perception by sampling perceptions from a target audience. I used this method to derive a model of color perceptions for crowdsourced viewers that can inform encodings for web-based visualizations.

Viewers' abilities to distinguish between encodings are also influenced by the size of a mark. Unlike in surface visualizations, the size of a mark is often inherently bound to the data or display. I derive a model for color as a function of size. I show how this model can be applied to generate robust encodings even for designs where the minimum mark size may be unknown a priori, such as in bar charts. To demonstrate the utility of this model for visualization design, I present a system for authoring color encodings based on this model.

These findings collectively inform how visualization can leverage color to support complete data analyses by supporting both aggregate tasks at large scales and point tasks over specific values.

9.1 Issues & Limitations

There are many unexplored issues and limitations in this work. Many of these limitations are discussed within their component chapters. Here, I will outline additional limitations and issues, focusing on limitations of that span multiple projects. Some provide interesting avenues for future work, but do not necessarily reduce the contributions of the presented projects. Others are issues that should be addressed in order to improve the utility of these results for visualization.

Supporting Different Visual Aggregation Tasks: This discussion of visual aggregation in this dissertation focuses on a specific set of abstract tasks (extrema, range, mean, variance, and outliers) and on tasks from specific domains (genomics, text analysis, and structural biology). A broader consideration of the different kinds of visual aggregation tasks that can be conducted (and how visualizations might support them) is necessary. This will open new research questions for graphical perception and help designers reason about how to support these tasks in practice.

Performance Costs of Visual Aggregation: Visual aggregation can allow viewers to find and estimate high-level patterns in data. However, visually estimating this information likely comes at a cost. For example, some aggregate judgments may be infeasible or inefficient for the visual system to compute and would be better supported by explicit computation. Understanding when and why designers should support visual aggregation rather than computational methods is important for understanding its utility in practice.

Scalability Evaluation: The claims about scalability made in the first part of this dissertation are validated by proof-of-concept. While the techniques presented here help visualizations scale beyond existing limits, it is unclear how extensively these methods scale. A more extensive evaluation, both quantitatively evaluating how these designs support aggregate analyses and when they might break down, would provide a better understanding of how these designs support scalability.

Balancing High-Level and Low-Level Goals: The goals of the first and second half of this dissertation may be somewhat at odds–the first relies on the visual system combining color and the second on differentiating color. It is important

that a data encoding supports the appropriate perceptual goals. Continuous but discriminable colors will still support visual aggregation and be robust for more exacting judgments. Distinctly discriminable colors may impede perceptions of aggregate gradients in data. There is a trade-off between inherent in designs for each task. The presented models address this constraint by being probabilistic—they let the designer decide. Understanding the trade-offs involved in discriminability for both visual aggregation and point-level tasks will help designers to reason about this trade-off.

Continuous Multiscale Visual Analysis: This work focuses on visualization tasks at two levels: aggregate (visual aggregation) and point (value comparison). Visualization analysis, in reality, often necessitates that viewers can explore data across continuous levels of detail. For example, a biologist might compare large sets of genomes to understand evolutionary relationships across species. At an aggregate level, they can understand how similar a large set of genomes are overall and identify subsets of genes with interesting behaviors. They can then explore these subsets in more detail to find individual genes to explore at lower levels of detail. The work presented in this dissertation explores how visualizations might support the first and last pieces of this workflow. Understanding how visualizations might remain effective as analysis moves across different levels of detail is important future work. This includes testing the assumption that using a consistent encoding across all levels of detail reduces cognitive burdens on the user. It is alternatively possible that multiscale analysis should change visual encodings at different levels of detail.

Interaction: All of the studies presented in this work consider only static analyses. While interaction can complicate study design, it provides a number of potential benefits for completing visualization tasks. For example, instead of visually estimating an answer, the user can interact with data to compute exact quantities on the fly. Interaction may also provide methods for balancing aggregate and point-level goals in visualization, such as by refining encodings on the fly [Elmqvist et al., 2011]. A better understanding of how interaction can be used in practice to support aggregate and point tasks would substantially inform visualization design.

Mitigation Limitations in Color Perception: The presented methods for designing color encodings for point tasks focus on how designers can mitigate perceptual

limitations agnostic to data. As a result, designers consider only the worst case designs, meaning these models will generally overcorrect for potential error. Alternative methods have been proposed that instead correct colors after a visualization has been rendered [Mittelstädt et al., 2014]. This represents a trade-off. Data-agnostic methods are only need to be considered at design time but can only approximate corrections based on design constraints. Data-aware methods are computed every time the view changes, but can potentially provide more accurate perceptions of color. Better understanding this trade-off is important future work.

Color Vision Deficiencies: The work presented here assumes normal color vision. However, accommodating colorblind viewers introduces a new set of questions to consider. For example, how can color encodings be designed such that visual summaries are meaningful for colorblind viewers? How can we generate models of color difference perceptions that capture viewing needs associated with different forms of color blindness? I anticipate that the modeling procedures introduced in Chapter 7 can inform encoding metrics that account for color blindness in practice, but future work is needed to confirm this hypothesis.

Contextualizing Visualization Tasks: The models constructed in this work ask participants to answer intentionally specific questions (e.g. do the bars appear to be the same color?). These tasks are designed to simplify modeling by asking directly for relevant percepts. This choice may trade-off modeling simplicity for ecological validity: they may or may not capture the strategies used to compare values in a visualization. A deeper evaluation of how question framings influence color identification performance is necessary to understand this limitation.

Generalizability: The generalizability of the models presented in this work is only validated to a limited extent. For example, the color modeling method is only tested for crowdsourced viewers and for marks of different sizes. It is unclear how well these results translate to, for example, cell phone viewers or marks of different shapes. This provides some evidence in support of the design metrics developed here, but needs to be expanded to better understand how these metrics can be used in practice.

9.2 Future Work

This work represents first steps in many directions. Many potential research projects could build on this work.

Two-Dimensional Aggregation: The systems and techniques presented in the first part of this dissertation attempt to overcome horizontal limitations in screen space by aggregating along one dimension. I intend to extend these ideas to support the aggregation of multiple sequences together to further increase scalability by increasing the number of distinct sequences or series that can be visualized at any given time. I will represent these aggregate sequences by leveraging alternative visual features that might also support visual aggregation tasks, such as size.

Broader System Deployment: The systems in this work have only been deployed to a limited number of users. I hope to increase the value of these systems by making them available to a broader audience.

Color Appearance as a Function of Size: In this work, I consider how discriminability changes as a function of size. Part of the reason marks become more difficult to distinguish as they become smaller is that their general appearance shifts. By instead modeling this shift, designers can correct for this shift when data is displayed in order to ensure that all marks appear identically. Building a computationally-tractable model of color appearance as a function of size to use for these applications is important future work.

Considering Other Visual Encodings: This dissertation focuses on color. Many of the knowledge gaps addressed in this dissertation exist for other encodings as well. For example, orientation is well-studied in perception and has been shown to be effectively averaged by the visual system. In visualization, orientation is often dismissed as being less effective for encoding point information. A better understanding of how different encoding channels might support visual aggregation could inform novel visualization approaches.

These works collectively reframe how visualization designers can think about color in visualization. They present evidence for the utility of color for aggregate tasks and methods for reframing how designers reason about color for comparing individual values. I do not intend for the work presented here to be the final word

in considering visualization designs for aggregation, nor for understanding color in practice. I see this dissertation instead as guiding new conversations about color in visualization.

BIBLIOGRAPHY

E. H. Adelson. Perceptual organization and the judgment of brightness. *Science-AAAS-Weekly Paper Edition-including Guide to Scientific Information*, 262(5142): 2042–2044, 1993.

E. H. Adelson. Lightness perception and lightness illusions. *The new cognitive neurosciences*, page 339, 1999.

E. H. Adelson and A. P. Pentland. The perception of shading and reflectance. *Perception as Bayesian inference*, pages 409–423, 1996.

T. Agostini and A. Galmonte. Perceptual organization overcomes the effects of local surround in determining simultaneous lightness contrast. *Psychological Science*, 13(1):89–93, 2002.

W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):47–60, 2008.

W. Aigner, C. Kainz, R. Ma, and S. Miksch. Bertin was right: An empirical evaluation of indexing to compare multivariate time-series data using line plots. In *Computer Graphics Forum*, volume 30, pages 215–228. Wiley Online Library, 2011.

D. Albers, C. Dewey, and M. Gleicher. Sequence Surveyor: Leveraging overview for scalable genomic alignment visualization. *IEEE TVCG*, 17(12):2392 – 2401, 2011. doi: 10.1109/TVCG.2011.232.

D. Albers Szafir, M. Stone, and M. Gleicher. Adapting color difference for design. In *Proc. 22nd Color and Imaging Conf.*, November 2014.

R. Alfvin and M. Fairchild. Observer variability in metameric color matches using color reproduction media. *Color Research & Application*, 22(3):174–188, 1997.

S. R. Allred and D. H. Brainard. Contrast, constancy, and measurements of perceived lightness under parametric manipulation of surface slant and surface reflectance. *JOSA A*, 26(4):949–961, 2009.

G. Alvarez and A. Oliva. The role of global layout in visual short-term memory. *OPAM*, 2006. URL `http://visionlab.harvard.edu/Members/George/Publications_files/Alvarez-Oliva-2007-VisCog.pdf`.

G. Alvarez and A. Oliva. Spatial ensemble statistics are efficient codes that can be represented with reduced attention. *Proc. of the National Academy of Sciences*, 106(18):7345–7350, 2009.

G. A. Alvarez and P. Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological science*, 15(2):106–111, 2004.

G. A. Alvarez, T. Konkle, and A. Oliva. Searching in dynamic displays: Effects of configural predictability and spatiotemporal continuity. *J. Vis.*, 7(14):1–12, 12 2007. ISSN 1534-7362. URL http://journalofvision.org/7/14/12/.

B. L. Anderson and J. Winawer. Layered image representations and the computation of surface lightness. *Journal of Vision*, 8(7):18, 2008.

R. M. Andrei, M. Callieri, M. F. Zini, T. Loni, G. Maraziti, M. C. Pan, and M. Zoppè. Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics*, 13(4), 2012.

C. Andrews, A. Endert, B. Yost, and C. North. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization*, page 1473871611415997, 2011.

N. Andrienko and G. Andrienko. *Exploratory analysis of spatial and temporal data*. Springer Berlin, Germany, 2006.

D. Ariely. Seeing sets: Representation by statistical properties. *Psychological Science*, 12(2):157–162, 2001a.

D. Ariely. Seeing sets: Representation by statistical properties. *Psychol. Sci.*, 12 (2):157–162, 2001b.

R. Arnheim. The Perception of Maps. *Cartography and Geographic Information Science*, 3(1):5–10, Apr. 1976. ISSN 15230406. doi: 10.1559/152304076784080276. URL http://openurl.ingenta.com/content/xref?genre=article&issn=1523-0406&volume=3&issue=1&spage=5.

B. Balas, L. Nakano, and R. Rosenholtz. A summary-statistic representation in peripheral vision explains visual crowding. *Journal of Vision*, 9(12):1–18, 2009. ISSN 1534-7362. URL http://journalofvision.org/9/12/13/.

S. Baldassi, N. Megna, and D. C. Burr. Visual clutter causes high-magnitude errors. *PLoS biology*, 4(3):387, 2006.

B. Bauer. Does stevensâŁ™s power law for brightness extend to perceptual brightness averaging? *The Psychological Record*, 59(2):2, 2010.

J. T. Behrens, W. A. Stock, and C. Sedgwick. Judgment errors in elementary box-plot displays. *Commun. Stat.-Simul. C*, 19(1):245–262, 1990. doi: 10.1080/03610919008812855. URL http://www.tandfonline.com/doi/abs/10.1080/03610919008812855.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res.*, 28: 235–242, 2000.

R. Berns. *Bilmeyer and Saltzman's Principles of Color Technology, third edition*. Wiley-Interscience publication. Wiley, 2000. ISBN 9780471194590.

J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.

K. Bhoyar and O. Kadke. Color image segmentation based on jnd color histogram. *International Journal of Image Processing (IJIP)*, 3(6):283, 2010.

E. Boring. Urban's tables and the method of constant stimuli. *Am. J. Psychol.*, 28 (2):280–293, 1917.

M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE TVCG*, 17(12):2479–2488, 2011.

D. Borland. Ambient occlusion opacity mapping for visualization of internal molecular structure. *Journal of WSCG*, pages 17–24, 2011.

D. Borland and R. M. Taylor. Rainbow color map (still) considered harmful. *IEEE Comput. Graph.*, 27(2):14–17, 2007.

D. Brainard and B. Wandell. Asymmetric color matching: how color appearance depends on the illuminant. *JOSA A*, 9(9):1433–1448, 1992.

D. H. Brainard and W. T. Freeman. Bayesian color constancy. *JOSA A*, 14(7): 1393–1411, 1997.

M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2376–2385, 2013.

P. Bressan. The place of white in a world of grays: a double-anchoring theory of lightness perception. *Psychological review*, 113(3):526, 2006.

C. A. Brewer. Basic mapping principles for visualizing cancer data using geographic information systems (gis). *American Journal of Preventive Medicine*, 30(2):S25–S36, 2006.

C. A. Brewer, G. W. Hatchard, and M. A. Harrower. Colorbrewer in print: A catalog of color schemes for maps. *Cartogr. Geogr. Inf. Sci.*, 30, 2003a. doi: doi: 10.1559/152304003100010929. URL http://www.ingentaconnect.com/content/acsm/cagis/2003/00000030/00000001/art00002.

C. A. Brewer, G. W. Hatchard, and M. A. Harrower. Colorbrewer in print: a catalog of color schemes for maps. *Cartogr. Geogr. Inf. Sci.*, 30(1):5–32, 2003b.

M. Bruls, K. Huizing, and J. Van Wijk. Squarified treemaps. In *Data Visualization*, pages 33–42. Springer, 2000.

D. Buckley, J. P. Frisby, and J. Freeman. Lightness perception can be affected by surface curvature from stereopsis. *PERCEPTION-LONDON-*, 23:869–869, 1994.

M. Buhrmester, T. Kwang, and S. Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspect. Psychol. Sci.*, 6(1):3–5, 2011.

N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1172–1181, 2010.

S. K. Card and J. Mackinlay. The structure of the information visualization design space. In *Proc. IEEE Symp. InfoVis*, pages 92–99. IEEE, 1997.

M. Carpendale. Considering visual variables as a basis for information visualisation. 2003.

R. Carter and L. D. Silverstein. Size matters: Improved color-difference estimation for small visual targets. *J. Soc. Inf. Display*, 18(1):17, 2010. ISSN 10710922. doi: 10.1889/JSID18.1.17. URL http://doi.wiley.com/10.1889/1.1828693.

R. C. Carter and L. D. Silverstein. Perceiving color across scale: great and small, discrete and continuous. *JOSA A*, 29(7):1346–1355, 2012.

J. Cataliotti and A. Gilchrist. Local and global processes in surface lightness perception. *Perception & Psychophysics*, 57(2):125–135, 1995.

N. Cawthon and A. V. Moere. The effect of aesthetic on the usability of data visualization. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 637–648. IEEE, 2007.

S. C. Chong and A. Treisman. Representation of statistical properties. *Vision research*, 43(4):393–404, 2003.

H. Choo, B. Levinthal, and S. Franconeri. Average orientation is more accessible through object boundaries than surface features. *Journal of Experimental Psychology: Human Perception and Performance*, 38(3):585, 2012a.

H. Choo, B. R. Levinthal, and S. L. Franconeri. Average orientation is more accessible through object boundaries than surface features. *Journal of Experimental Psychology: Human Perception and Performance*, 38(3):585, 2012b.

G. Cipriano and M. Gleicher. Molecular surface abstraction. *IEEE TVCG*, 13(6): 1608–1615, 2007.

G. M. Cipriano, G. N. Philips, Jr, and M. Gleicher. Local functional descriptors for surface comparison based binding prediction. *BMC Bioinformatics*, 13(1):314, 2012.

W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

W. S. Cleveland, R. McGill, et al. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

R. B. Corey and L. Pauling. Molecular models of amino acids, peptides, and proteins. *Rev. Sci. Instrum.*, 24(8):621–627, 1953.

M. Correll, S. Ghosh, D. O'Connor, and M. Gleicher. Visualizing virus population variability from next generation sequencing data. In *2011 IEEE Symp. Bio. Data Vis. (BioVis)*, pages 135–142. IEEE, 2011.

M. Correll, D. Albers, S. Franconeri, and M. Gleicher. Comparing averages in time series data, May 2012a.

M. Correll, D. Albers, S. Franconeri, and M. Gleicher. Comparing averages in time series data. In *Proc. 2012 ACM Human Factors in Computing Systems*, pages 1095–1104. ACM, May 2012b.

M. A. Correll, E. C. Alexander, and M. Gleicher. Quantity estimation in visualizations of tagged text. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2697–2706. ACM, 2013.

B. Crawford. Colour matching and adaptation. *Vision Research*, 5(1):71–78, 1965.

A. C. E. Darling, B. Mau, F. R. Blattner, and N. T. Perna. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*, 14(7):1394–1403, 2004. URL `http://dx.doi.org/10.1101/gr.2289704`.

V. M. de Almeida, P. T. Fiadeiro, and S. M. Nascimento. Effect of scene dimensionality on colour constancy with real three-dimensional scenes and objects. *Perception*, 39(6):770, 2010.

D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. In *ACM T. Graphic.*, volume 22, pages 848–855. ACM, 2003.

W. L. DeLano. The PyMOL molecular graphics system. 2002.

K. Devlin, A. Chalmers, and E. Reinhard. Visual calibration and correction for ambient illumination. *ACM TAP*, 3(4):429–452, 2006.

J. Duncan and G. W. Humphreys. Visual search and stimulus similarity. *Psychological review*, 96(3):433, 1989.

C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proc. CHI 2013*, pages 3247–3256. ACM, 2013.

W. D. Dupont and W. D. Plummer. Density distribution sunflower plots. *Journal of Statistical Software*, 8(3):1–5, 2003.

M. Eckert and A. Bradley. Perceptual quality metrics applied to still image compression. *Signal processing*, 70(3):177–200, 1998.

N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE TVCG*, 16(3):439–454, 2010.

N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Color lens: Adaptive color scale optimization for visual exploration. *Visualization and Computer Graphics, IEEE Transactions on*, 17(6):795–807, 2011.

A. Endert, C. Andrews, Y. H. Lee, and C. North. Visual encodings that support physical navigation on large displays. In *Proc. of Graph, Interface 2011*, pages 103–110. Canadian Human-Computer Communications Society, 2011.

J. Enns. Seeing textons in context. *Perception & Psychophysics*, 39(2):143–147, 1986.

M. Fairchild. *Color appearance models*. J. Wiley, 2005.

M. Fairchild and R. Berns. Image color-appearance specification through extension of cielab. *Color Research & Application*, 18(3):178–190, 1993.

M. D. Fairchild. *Color appearance models*. John Wiley & Sons, 2013.

J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 117–124. IEEE, 2002.

A. Fiorentini. Differences between fovea and parafovea in visual search processes. *Vision research*, 29(9):1153–1164, 1989.

D. H. Foster. Color constancy. *Vision Res.*, 51(7):674–700, 2011.

S. Franconeri, D. Bemis, and G. Alvarez. Number estimation relies on a set of segmented objects. *Cognition*, 113(1):1–13, 2009.

J. Freeman and E. P. Simoncelli. Metamers of the ventral stream. *Nature neuroscience*, 14(9):1195–1201, 2011.

J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proc. CHI 2013*, pages 3237–3246. ACM, 2013.

S. Garlandini and S. I. Fabrikant. Evaluating the effectiveness and efficiency of visual variables for geographic information visualization. In *Spatial information theory*, pages 195–211. Springer, 2009.

A. Gilchrist and A. Jacobsen. Perception of lightness and illumination in a world of one reflectance. *Perception*, 13(1):5–19, 1984.

A. Gilchrist, C. Kossyfidis, F. Bonato, T. Agostini, J. Cataliotti, X. Li, B. Spehar, V. Annan, and E. Economou. An anchoring theory of lightness perception. *Psych. Rev.*, 106(4):795, 1999.

A. L. Gilchrist and V. Annan. Articulation effects in lightness: Historical background and theoretical implications. *Perception*, 31(2):141–150, 2002.

M. Gleicher. Explainers: Expert explorations with crafted projections. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2042–2051, 2013.

M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of average value in multiclass scatterplots. *IEEE TVCG*, 19(12):2316–2325, 2013a.

M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of average value in multiclass scatterplots. *IEEE TVCG*, 19(12):2316–2325, 2013b.

D. J. Graham, J. D. Friedenberg, and D. N. Rockmore. Efficient visual system processing of spatial and luminance statistics in representational and non-representational art. In *IS&T/SPIE Electronic Imaging*, pages 72401N–72401N. International Society for Optics and Photonics, 2009.

J. J. Granzier, E. Brenner, and J. B. Smeets. Can illumination estimates provide the basis for color constancy? *J. Vis.*, 9(3), 2009.

S. Grossberg and S. Hong. A neural model of surface perception: Lightness, anchoring, and filling-in. *Spatial Vision*, 19(2):263–321, 2006.

P. Grosset, M. Schott, G.-P. Bonneau, and C. D. Hansen. Evaluation of Depth of Field for Depth Perception in DVR. In *Proc. IEEE Pacific Visualization*, 2013.

H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE TVCG*, 13(6):1270–7, 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70623. URL `http://www.ncbi.nlm.nih.gov/pubmed/17968074`.

A. Hård and L. Sivik. NCS Natural Color System: a Swedish standard for color notation. *Color Res. Appl.*, 6(3):129–138, 2007.

L. H. Hardy, G. Rand, and M. C. Rittler. Tests for the detection and analysis of color-blindness. *J. Opt. Soc. Am.*, 35(4):268–271, 1945.

S. Haroz and D. Whitney. How capacity limits of attention influence information visualization effectiveness. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2402–2410, 2012.

L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics*, 2014.

M. Harrower and C. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, June 2003. ISSN 00000000. doi: 10.1179/000870403235002042.

C. Healey and J. Enns. Building perceptual textures to visualize multidimensional datasets. In *Visualization'98. Proceedings*, pages 111–118. IEEE, 1998.

C. Healey, K. Booth, and J. Enns. High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2): 107–135, 1996.

C. G. Healey and J. T. Enns. Attention and visual memory in visualization and computer graphics. *IEEE TVCG*, 18(7):1170–1188, 2012a.

C. G. Healey and J. T. Enns. Attention and visual memory in visualization and computer graphics. *Visualization and Computer Graphics, IEEE Transactions on*, 18(7):1170–1188, 2012b.

M. Hedrich, M. Bloj, and A. I. Ruppertsberg. Color constancy improves for real 3d objects. *J. Vis.*, 9(4), 2009.

J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 203–212. ACM, 2010.

J. Heer and M. Stone. Color naming models for color selection, image editing and palette design. In *Proc. CHI*, pages 1007–1016. ACM, 2012.

J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1303–1312. ACM, 2009.

S. Highnote. *Color Discrimination of Small Targets*. University of California, San Diego, 2003. URL `https://books.google.com/books?id=ixnGNwAACAAJ`.

K. Hornbæk and M. Hertzum. The notion of overview in information visualization. *Int. J. Hum. Comput. Stud.*, 69(7):509–525, 2011.

W. Humphrey, A. Dalke, K. Schulten, et al. VMD: visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38, 1996.

O. Irsoy, O. T. Yildiz, and E. Alpaydin. Design and analysis of classifier learning experiments in bioinformatics: Survey and case studies. *IEEE/ACM Trans. Comput. Biol. and Bioinform.*, 9(6):1663–1675, Nov. 2012. ISSN 1545-5963.

W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):927–934, 2010.

D. Jen, L. Larson, C. Stolte, D. DeCaprio, T. Allen, B. Birren, M. Koehrsen, and M. Henn. Comparative viral genome visualization. IEEE InfoVis Poster Proceedings, 2009.

S. Kaski, J. Venna, and T. Kohonen. Coloring that reveals high-dimensional structures in data. In *Neural Information Processing, 1999. Proceedings. ICONIP'99. 6th International Conference on*, volume 2, pages 729–734. IEEE, 1999.

D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):59–78, 2000.

D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: A visualization technique for very large multi-attribute data setsâŁ . *Information Visualization*, 1(1):20–34, 2002.

D. A. Keim, J. Schneidewind, and M. Sips. *Scalable pixel based visual data exploration*. Springer, 2007.

J. M. Kennedy and J. Bai. Cavanagh and leclerc shape-from-shadow pictures: Do line versions fail because of the polarity of the regions or the contour? *Perception*, 2000.

W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The Human Genome Browser at UCSC. *Genome Research*, 12 (6):996–1006, 2002. doi: 10.1101/gr.229102. URL `http://genome.cshlp.org/content/12/6/996.abstract`.

F. Kingdom and B. Moulden. A multi-channel approach to brightness coding. *Vision research*, 32(8):1565–1582, 1992.

F. A. Kingdom. Perceiving light versus material. *Vision Research*, 48(20):2090–2105, 2008.

F. A. Kingdom. Lightness, brightness and transparency: A quarter century of new ideas, captivating demonstrations and unrelenting controversy. *Vision Research*, 51(7):652–673, 2011.

A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456. ACM, 2008.

F. Kleiner. *GardnerâĽ™s art through the ages: The western perspective*, volume 1. Cengage Learning, 2013.

J. Krantz. Stimulus delivery on the web: What can be presented when calibration isn't possible. *Dimensions of Internet science*, pages 113–130, 2001.

M. Krone, M. Falk, S. Rehm, J. Pleiss, and T. Ertl. Interactive exploration of protein cavities. In *Computer Graphics Forum*, volume 30, pages 673–682, 2011.

M. Krstajic, E. Bertini, and D. Keim. Cloudlines: compact display of event episodes in multiple time-series. *IEEE TVCG*, 17(12):2432–2439, 2011.

T. Lammarsch, W. Aigner, A. Bertone, J. Gartner, E. Mayr, S. Miksch, and M. Smuc. Hierarchical temporal patterns and interactive aggregated views for pixel-based visualizations. In *Int. Conf. InfoVis*, pages 44–50, 2009.

O. D. Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 171–178. IEEE, 2011.

E. H. Land et al. *The retinex theory of color vision*. Scientific America., 1977.

H. Landis. Production-ready global illumination. *Siggraph course notes*, 16(2002): 11, 2002.

M. S. Langer and H. H. Bulthoff. A prior for global convexity in local shape-from-shading. *Perception*, 30(4):403–410, 2001.

M. S. Langer, H. H. Bülthoff, et al. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649, 2000.

C. I. D. L'Eclairage. Recommendations on uniform color spaces-color difference equations, psychomettic color terms. *Paris: CIE*, 1978.

B. Lee and F. M. Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379–IN4, 1971.

H. Legrand, G. Rand, and C. Rittler. Tests for the detection and anlysis of color-blindness i. the ishihara test: An evaluation. *Journal of the Optical Society of America*, 35:268, 1945.

Y. Li and Z. Pizlo. Depth cues versus the simplicity principle in 3d shape perception. *Topics in cognitive science*, 3(4):667–685, 2011.

S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. In *Computer Graphics Forum*, volume 32, pages 401–410. Wiley Online Library, 2013.

F. Lindemann and T. Ropinski. About the influence of illumination models on image comprehension in direct volume rendering. *IEEE TVCG*, 17(12):1922–1931, 2011.

Z. Liu and J. T. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):999–1008, 2010.

M. Livingston, J. W. Decker, et al. Evaluation of trend localization with multivariate visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2053–2062, 2011.

A. D. Logvinenko. Lightness induction revisited. *PERCEPTION-LONDON-*, 28: 803–816, 1999.

M. C. Lohrenz, J. G. Trafton, M. R. Beck, and M. L. Gendron. A model of clutter for complex, multivariate geospatial displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 51(1):90–101, 2009.

M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Res. Appl.*, 26(5):340–350, 2001.

A. M. MacEachren, R. E. Roth, J. O'Brien, B. Li, D. Swingley, and M. Gahegan. Visual semiotics & uncertainty visualization: An empirical study. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2496–2505, 2012.

M. Mahy, L. Van Eycken, and A. Oosterlinck. Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Res. Appl.*, 19(2): 105–121, 1994a.

M. Mahy, L. Van Eycken, and A. Oosterlinck. Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Research & Application*, 19(2):105–121, 1994b.

G. Mather and D. R. Smith. Combining depth cues: effects upon accuracy and speed of performance in a depth-ordering task. *Vision Res.*, 44(6):557–562, 2004.

M. Meyer, T. Munzner, and H. Pfister. Mizbee: A multiscale synteny browser. 15 (6):897–904, Nov. 2009. doi: 10.1109/TVCG.2009.167.

J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011. doi: 10.1126/science.1199644. URL `http://www.sciencemag.org/content/331/6014/176.abstract`.

S. Mittelstädt, A. Stoffel, and D. A. Keim. Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum*, volume 33, pages 231–240. Wiley Online Library, 2014.

S. Mittelstädt, D. Jäckle, F. Stoffel, and D. A. Keim. Colorcat: Guided design of colormaps for combined analysis tasks. *Computer Graphics Forum*, 2015.

M. Morgan, C. Chubb, and J. A. Solomon. A âĽ˜dipperâĽ™function for texture discrimination based on orientation variance. *Journal of Vision*, 8(11):9, 2008.

N. Moroney, M. D. Fairchild, R. W. Hunt, C. Li, M. R. Luo, and T. Newman. The ciecam02 color appearance model. In *Color and Imaging Conference*, volume 2002, pages 23–27. Society for Imaging Science and Technology, 2002.

K. Mullen. The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *The Journal of Physiology*, 359(1):381–400, 1985.

T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

S. Newhall, R. Burnham, and R. Evans. Color constancy in shadows. *J. Opt. Soc. Am.*, 48(12):976–984, 1958.

H.-C. Nothdurft. Saliency effects across dimensions in visual search. *Vision Research*, 33(5):839–844, 1993.

B. Oicherman, M. Luo, B. Rigg, and A. Robertson. Effect of observer metamerism on colour matching of display and surface colours. *Color Res. Appl.*, 33(5): 346–359, 2008.

A. Oliva. Gist of the scene. In Neurobiology of Attention, 2005.

A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

M. Olkkonen, T. Hansen, and K. R. Gegenfurtner. Color appearance of familiar objects: Effects of object shape, texture, and illumination changes. *J. Vis.*, 8(5), 2008a.

M. Olkkonen, T. Hansen, and K. R. Gegenfurtner. Color appearance of familiar objects: Effects of object shape, texture, and illumination changes. *J. Vis.*, 8(5): 13, 2008b.

M. Olkkonen, C. Witzel, T. Hansen, and K. Gegenfurtner. Categorical color constancy for rendered and real surfaces. *J. Vis.*, 9(8):331–331, 2009.

S. Papadimitriou, J. Sun, C. Faloutos, and S. Y. Philip. Dimensionality reduction and filtering on time series sensor streams. In *Managing and Mining Sensor Data*, pages 103–141. Springer, 2013.

L. Pauling, R. B. Corey, and H. R. Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A.*, 37(4):205–211, 1951.

T. Peeters, M. Fiers, H. van de Wetering, J.-P. Nap, and J. J. van Wijk. Case Study: Visualization of annotated DNA sequences. *Eurographics*, 2004.

K. Perlin. An image synthesizer. *Comput. Graph. (SIGGRAPH'85)*, 19(3):287–296, 1985.

E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004.

S. Pizer and F. Chan. Evaluation of the number of discernible levels produced by a display. In *Inf. Process. Med. Imaging*, pages 561–580. Editions INSERM, Paris, 1980.

C. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *Selected Areas in Communications, IEEE Journal on*, 16(4):525–539, 1998.

J. B. Procter, J. Thompson, I. Letunic, C. Creevey, F. Jossinet, and G. Barton. Visualization of multiple alignments, phylogenies and gene family evolution. *Nature Methods*, 7(3):S16–S25, Mar. 2010. doi: doi:10.1038/nmeth.1434.

R. A. Rensink. On the prospects for a science of visualization. In *Handbook of Human Centric Visualization*, pages 147–175. Springer, 2014.

R. A. Rensink and G. Baldridge. The perception of correlation in scatterplots. In *Computer Graphics Forum*, volume 29, pages 1203–1210. Wiley Online Library, 2010.

F. M. Richards. Areas, volumes, packing, and protein structure. *Annu. Rev. Biophys.*, 6(1):151–176, 1977. doi: 10.1146/annurev.bb.06.060177.001055. URL `http://www.annualreviews.org/doi/abs/10.1146/annurev.bb.06.060177.001055`. PMID: 326146.

P. Rizzo, A. Bierman, and M. Rea. Color and brightness discrimination of white leds. In *International Symposium on Optical Science and Technology*, pages 235–246. International Society for Optics and Photonics, 2002.

A. Robertson. Historical development of cie recommended color difference equations. *Color Res. Appl.*, 15(3):167–170, 2007.

B. E. Rogowitz and L. A. Treinish. Data visualization: the end of the rainbow. *Spectrum, IEEE*, 35(12):52–59, 1998.

R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: a measure of display clutter. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770. ACM, 2005.

R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *J Vis*, 7(2):17.1–1722, 2007. doi: 10.1167/7.2.17. URL `http://dx.doi.org/10.1167/7.2.17`.

R. E. Roth. Cartographic interaction primitives: Framework and synthesis. *The Cartographic Journal*, 49(4):376–395, 2012.

M. E. Rudd. How attention and contrast gain control interact to regulate lightness contrast and assimilation: a computational neural model. *Journal of vision*, 10 (14):40, 2010.

A. I. Ruppertsberg, M. Bloj, and A. Hurlbert. Sensitivity to luminance and chromaticity gradients in a complex scene. *J. Vis.*, 8(9), 2008.

M. Rutherford and D. Brainard. Lightness constancy: A direct test of the illumination-estimation hypothesis. *Psych. Sci.*, 13(2):142–149, 2002.

F. Samsel, M. Petersen, T. Geld, G. Abram, J. Wendelberger, and J. Ahrens. Colormaps that improve perception of high-resolution ocean data. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 703–710. ACM, 2015.

M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.

J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. Moorhead. A user study to compare four uncertainty visualization methods for 1d and 2d datasets. *IEEE TVCG*, 15(6):1209–1218, 2009.

A. Sarikaya, D. Albers, J. Mitchell, and M. Gleicher. Visualizing validation of protein surface classifiers. In *Computer Graphics Forum*, volume 33, pages 171–180. Wiley Online Library, 2014.

A. Sarkar, L. Blondé, P. Le Callet, F. Autrusseau, P. Morvan, J. Stauder, et al. A color matching experiment using two displays: design considerations and pilot test results. In *Proceedings of the Fifth European Conference on Color in Graphics, Imaging and Vision*, 2010.

H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *IEEE TVCG*, 19(12):2366–2375, 2013.

B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of IEEE Symp. Visual Languages*, pages 336–343. IEEE, 1996.

B. Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 3–12. ACM, 2008.

S. Silva, B. Sousa Santos, and J. Madeira. Using color in visualization: A survey. *Computers & Graphics*, 35(2):320–333, 2011.

L. Silverstein and R. Merrifield. Color selection and verification testing for airborne color crt displays. In *AACD Symp.*, pages 39–81, 1982.

M. Singh and D. Hoffman. Constructing and representing visual objects. *Trends in Cognitive Sciences*, 1(3):98–102, 1997.

J. Slack, K. Hildebrand, T. Munzner, and K. John. SequenceJuxtaposer: Fluid navigation for large-scale sequence comparison in context. In *German Conference on Bioinformatics*, pages 37–42, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.2900&rep=rep1&type=pdf`.

A. Slingsby, J. Dykes, and J. Wood. Configuring hierarchical layouts to address research questions. *IEEE TVCG*, 15(6):977 –984, 2009. doi: 10.1109/TVCG.2009.128.

S. V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sens. Environ.*, 62(1):77 – 89, 1997. ISSN 0034-4257.

M. Stokes, M. Fairchild, and R. Berns. Precision requirements for digital color reproduction. *ACM T. Graphic.*, 11(4):406–422, 1992.

M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet-sRGB. *Microsoft and Hewlett-Packard Joint Report*, 1996a.

M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet-sRGB. *Microsoft and Hewlett-Packard Joint Report*, 1996b.

M. Stone. Color balancing experimental projection displays. In *9th IS&t/SID Color Imag. Conf.*, volume 7, 2001.

M. Stone. Color in information display principles, perception, and models. In *ACM SIGGRAPH 2004 Course Notes*, page 21. ACM, 2004.

M. Stone. In color perception, size matters. *IEEE Computer Graphics & Applications*, 32(2):8–13, Mar. 2012. ISSN 0272-1716. URL `http://dl.acm.org/citation.cfm?id=2360758.2361073`.

B. Swihart, B. Caffo, B. James, M. Strand, B. Schwartz, and N. Punjabi. Lasagna plots: A saucy alternative to spaghetti plots. *Epidemiology*, 21(5):621–625, 2010.

M. Tarini, P. Cignoni, and C. Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE TVCG*, 12(5):1237–1244, 2006.

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

C. Tominski, G. Fuch, and H. Schumann. Task-driven color coding. In *Information Visualisation, 2008. IV'08. 12th International Conference*, pages 373–380. IEEE, 2008.

C. Vehlow, J. Heinrich, F. Battke, D. Weiskopf, and K. Nieselt. ihat: Interactive hierarchical aggregation table. In *Biological Data Visualization (BioVis), 2011 IEEE Symposium on*, pages 63–69. IEEE, 2011.

L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1739–1754, 2008.

C. Ware. *Information visualization*. Morgan Kaufmann, 3 edition, 2000.

C. Ware. *Visual Thinking for Design*. Morgan Kaufmann, 2008.

M. Wattenberg and F. Viegas. Beautiful history: Visualizing wikipedia. In J. Steele and N. Illinsky, editors, *Beautiful Visualization*, page 416. O'Reilly Media, Inc., 2010. URL `http://books.google.com/books?hl=en&lr=&id=TKh6fdlKwfMC&pgis=1`.

C. D. Wickens and C. M. Carswell. The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors*, 37 (3):473–494, 1995.

M. Wijffelaars, R. Vliegen, J. J. Van Wijk, and E.-J. Van Der Linden. Generating color palettes using intuitive parameters. In *Computer Graphics Forum*, volume 27, pages 743–750. Wiley Online Library, 2008.

J. Wolfe and S. Bennett. Preattentive object files: Shapeless bundles of basic features. *Vision research*, 37(1):25–43, 1997.

J. M. Wolfe. Asymmetries in visual search: An introduction. *Perception & Psychophysics*, 63(3):381–389, 2001.

J. N. Yang and S. K. Shevell. Stereo disparity improves color constancy. *Vision Res.*, 42(16):1979–1989, 2002.

X. Zhu, S. S. Ericksen, and J. C. Mitchell. DBSI: DNA-binding site identifier. *Nucleic Acids Res.*, 41(16):e160, 2013.

S. Zuffi, C. Brambilla, G. Beretta, and P. Scala. Understanding the readability of colored text by crowd-sourcing on the web. Technical report, External HPL-2009-182, HP Laboratories, August 6 2009, http://www. hpl. hp. com/techreports/2009/HPL-2009-182. html, 2009.