IS THAT RIGHT? DATA VISUALIZATIONS FOR SCIENTIFIC QUALITY CONTROL

by

Devin Lange

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

August 2024

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Devin Lange**
has been approved by the following supervisory committee members:

Alexander Lex, Chair(s) __
Date Approved

Paul Rosen, Member __
Date Approved

Kate Isaacs, Member __
Date Approved

Nils Gehlenborg, Member __
Date Approved

Hanspeter Pfister, Member __
Date Approved

by **Mary W. Hall**, Chair/Dean of
the Department/College/School of **Computing**
and by **Darryl P. Butt**, Dean of The Graduate School.

ABSTRACT

Scientific inquiry is difficult. Collecting data on a phenomenon can reveal new insight into it. However, collecting and interpreting such data requires human expertise. In particular, distinguishing between a pattern in data is the result of the underlying phenomenon being measured or an error in the data collection requires a careful review of data while incorporating expert knowledge and human judgment. This dissertation describes two tensions that contribute to the difficulty of this review process. It then claims that visualizations are a valuable tool for easing these tensions.

To back up this claim, this dissertation describes three main contributions that provide specific examples of where visualizations have helped with the quality control process: (1) a cell microscopy visualization system that uses exemplar data points to bridge the gap between detailed individual data points in a large dataset, (2) composite visualization techniques that improve the interpretability of multi-modal data, (3), a visualization system for reviewing datasets for manipulation that embeds knowledge on common patterns found in manipulated data and guides the user through this knowledge without being prescriptive.

For Regina.

CONTENTS

CHAPTERS

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude first to my advisor, Alex Lex. Alex has been an exceptional mentor who has helped me grow as a scientists and a person. He has helped me strive for academic excellence.

To my committee, Paul Rosen, Kate Isaacs, Nils Gehlenborg, and Hanspeter Pfister, thank you for your advice, guidance, and ideas, and thank you for challenging me. To members of the Zangle Lab and Judson-Torres Lab, working with you all has been an amazing experience. Thank you for providing me the opportunity to learn about cell biology and microscopes.

I have been fortunate to have many amazing academic communities. Thank you all the members of the Visualization Design Lab, the SCI Institute, the University of Utah, and the broader visualization research community. It is amazing to have so many people I can discuss visualizations with, commiserate in our setbacks, and celebrate our successes.

To my friends and family thank you for all your support in life and your sometimes real, sometimes feigned interest in my research. Lastly, to my wife Regina, thank you for everything.

CHAPTER 1

INTRODUCTION

In scientific research, data is often used to represent some underlying reality. Thus, making observations and collecting data about a phenomenon of interest is a precondition for learning about the phenomenon. In practice, however, the data never perfectly represents reality. At best, it is a useful, close approximation of the underlying phenomenon. At worst, it can misrepresent reality. If conclusions about the phenomenon of interest come from collected data, but the data can be wrong, how can we make progress and improve our understanding?

This dissertation focuses on the issue of determining the quality of scientific data — data quality control. Data quality must be "good enough" to make sound judgments based on it, but how do we know if the data is good enough? For instance, say a cell biologist collects data on cell behavior with an automated microscope, and the resulting data shows an unexpected value. Does this mean the microscope has malfunctioned and produced an incorrect reading, or has some unknown cell behavior just been discovered? Determining which of these two possibilities is real takes considerable effort and expertise.

Several strategies can be employed to determine the quality of scientific data. For instance, a cell biologist can review the images a microscope produces. A researcher trained on the equipment will be able to determine if the microscope is out of focus or if the imaging location is drifting over time. If the images look reasonable, the derived attributes based on those images might be reviewed. Some microscopy workflows use algorithms to compute attributes of the cells, such as their size, or track their movement over time. These algorithms are essential for high-throughput approaches resulting in large datasets. They often perform reasonably well but aren't as good as humans at certain tasks. Therefore, cell biologists can compare the output of these algorithms with their expected output based on the images, validating the algorithms or identifying needed improvements. Finally, a high-level analysis of experimental results is required. For instance, if the data appears sound but unexpectedly indicates that all the cells are dying, why is that? Is it because the drug those cells are exposed to is killing everything? Or is it because the light from the microscope was too intense and is frying the cells? Making this determination requires a deep understanding of the experimental process, cell biology, and the details in the dataset.

In general, data is not self-evident and can not be assumed to be correct [1]. Data must be interpreted to have meaning, requiring knowledge beyond what is directly encoded. In other words, human expertise is required to contextualize data. Even in the case where an unknown phenomenon is being measured, an expert must have knowledge of how the data was collected and use their judgment to determine if the results are plausible.

These are just some examples of how the quality of scientific data is evaluated. However, some common *tensions* are present in various scientific quality control approaches. The first tension, *Trees-and-Forest*, is between the detail-focused analysis of data and the broad synthesis of different knowledge. As the saying goes, you should avoid *missing the forest for the trees*; in other words, don't get lost in the details (trees) and forget the larger situation (forest). However, to fully understand a forest, the trees cannot be ignored completely. So, how do you balance the detailed data analysis with the broader interpretation of that data? For instance, evaluating an image analysis algorithm that calculates derived values of cell images requires both detailed and broad thinking. The detailed analysis involves looking at individual images and comparing them to the derived values. The broad thinking requires a deep understanding and reflection of both cell biology and image processing algorithms.

The second tension, *Flexible-Structure*, is between a structured and flexible exploration for quality control. Airplane pilots famously review a comprehensive checklist before each take-off. Researchers can also benefit from a structured approach to reviewing data quality. However, scientific experiments are more varied than flights — this necessitates a more flexible approach where researchers can also explore threads of inquiry into their data. Data analysts, in general, constantly transition between a directed and exploratory analysis of their data [2].

The main claim of this dissertation is that data visualizations are well-suited to ease these tensions. The reasons for this are multifaceted and will be explored in detail, but the short explanation is that visualizations can be designed with these tensions in mind. We will provide three in-depth examples of visualizations we have developed to illustrate how visualizations can provide rich detail of individual data points while still contextualizing them in the larger dataset (*Trees-and-Forest*), and how visualization systems can provide structure while still supporting exploration (*Flexible-Structure*).

Althrough these tensions are conceptualized with data quality control in mind, there is overlap with other data analysis tasks. The same tools that can be used for quality control can also be used to garner insights about the data. Thus, it is impossible to completely separate the processes of data analysis and quality control. Visualizations are well known for their general capabilities to help people characterize their data [3]. This characterization includes the ability to quickly understand the distributions and trends of data, spot patterns quickly, and identify outliers. Furthermore, visualizations can be designed to include an overview of data combined with detailed information [4], and the idea of providing a flexible yet structured analysis of data through visualizations is not new [5], [6]. These strengths contribute to visualizations' ability to help research perform quality control, but this dissertation will focus more specifically on those that relate to the two quality-control tensions we have defined.

1.1 Contributions

This dissertation contains five contributions. The first two are specific to this dissertation, and the final three are related to the individual papers that comprise this dissertation.

1.1.1 Dissertation Level Contributions

The dissertation-level contributions are the result of reflection on the work conducted for this dissertation. The first contribution is the introduction of the **two tensions that describe how scientific quality control is difficult (*Flexible-Structure* describes the desire for both flexibility and structure and *Trees-and-Forest* covers the tension between detailed analysis and broad reflection)**. These tensions were identified across multiple projects, and their characterization provides a helpful reference for reasoning about scientific quality control, especially when developing tools to facilitate the process.

The second contribution is a **set of design guidelines for visualizations for quality**

control (show relationships of the data; automate sampling of data points, but allow arbitrary selection, and provide guidance). These guidelines were developed by pulling out the common themes of visualization design across projects and generalizing on why those designs are helpful. These guidelines can aid other visualization designers who are creating visualizations for scientific quality control.

1.1.2 Paper Level Contributions

This dissertation is composed of three projects that all improve the quality control experience of scientists. These projects contribute to the higher-level dissertation contributions, but each project also has more specific contributions. The first two projects, Loon and Aardvark, are the result of a collaboration with cell biologists studying cancer. These design studies illustrate how visualization systems can be developed to help scientists ensure their data is of high quality before sharing their results. The last project, Ferret, explores how visualizations can help reviewers guard against authors who work in bad faith and manipulate their scientific tabular datasets. These projects resulted in the three main contributions that comprise this dissertation:

- **Exemplar Visualizations.** When reviewing large datasets, visualizations are often used to review aggregated data, such as univariate distributions. However, for multimodal datasets where data is difficult to summarize, such as imaging data, it is essential also to review the raw data points. Loon [7] is a design study for cell microscopy visualizations. One technical contribution of this project is an exemplar visualization view. This view selects representative cells, or exemplars, and visualizes them together with metadata about those cells. In general, exemplars can be useful for any large dataset where individual data points are highly detailed in a way that is difficult to summarize. Exemplars help to ease both tensions, *Flexible-Structure*, and *Trees-and-Forest*. To provide *structure*, Exemplars are selected across experimental conditions and imaging locations to review a good sample of cells. At the same time, any cell can be selected to view in the exemplar view, allowing a *flexible* way to investigate interesting cells. Since the images of the cells are shown alongside the metadata related to those cells, this helps alleviate the analytical task of aligning data elements of different modalities together, thus relieving the tension between the

Trees-and-Forest.

- **Composite Visualization of Different Data Modalities.** Aardvark [8] is another design study based on our continued collaboration with cell microscopy researchers. This study develops visualization views that combine different data modalities (images, trees, and time-series) into a single space of composite visualization views. We describe general design principles for creating composite views and instantiate them in new views for Loon that combine three different data types together. These views are again designed so the data can be understood more holistically instead of requiring researchers to do the mentally taxing task of combining disparate pieces of data together. As a result, these views help ease the *Trees-and-Forest* tension between deep reflection of the data and analytical reasoning.
- **Guided Review of Artifacts of Manipulations.** Loon and Aardvark dive deeply into how scientists review their own data where they know everything in the process. However, quality control is also necessary when the person doing the checking is not as familiar with the data as in these cases: for example, when reviewing work done by other researchers. In this case, the information available is limited by what the other researcher provides. When reviewing the data provided, some quality inspection may be performed. However, this inspection has a blind spot when reviewers assume the submitter is acting in good faith. The checks they perform might spot common accidental errors in data processing but may miss when authors intentionally falsify or fabricate their dataset. Some of the patterns in falsified datasets are not the same as those in data with other quality issues. For this work, we categorized common patterns found in retracted datasets. These artifacts of manipulation provided the structure that the visualization tool Ferret [9] was designed around. In Ferret, different visual analyses are provided to highlight these common artifacts. This approach gives users some helpful guidance when doing this unfamiliar review but does not limit the user's exploration. This system provides another example of a visualization system with a structured and flexible solution (*Flexible-Structure)*.

The remainder of this thesis is structured as follows: Chapter 2 provides relevant background and work related to quality control of data with visualizations. The main contributions of this thesis are included in Chapters 3–5, which correspond to the published papers Loon [7] (Chapter 3), Aardvark [8] (Chapter 4), and Ferret [9] (Chapter 5). A broader discussion of how all three of these relate to this dissertation is included in Chapter 6, and a concluding summarization is in Chapter 7.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we discuss existing work related to scientific quality control of data. First we provide a high-level definition for high-quality data and describe different approaches for reasoning about data quality. We position how this thesis is similar to and different from these approaches. In particular, we discuss the difference between uncertainty and errors, as well as explicit and implicit quality. Next, we provide more details on how statistics and error metrics can be useful, but we also describe the ways in which they are limited. This dissertation is focused on how visualizations can overcome these limitations and provide utility for scientific quality control. Thus, we will describe how visualizations have already been used for his purpose, with a particular focus on cell biology data since it relates to two of our three projects. Finally, we will end with a discussion on how visualizations can be designed to guide user analysis since it relates to how our visualizations guide users through their quality control analysis.

2.1 Data Quality

First, it is important to discuss what high-quality data is or, conversely, what low-quality data is. For this dissertation, we will limit our scope to data that is recorded in order to observe a phenomenon of interest. For this case, at a high level, one way to consider data quality is by the difference between recorded values and the phenomenon of interest. Datasets that closely match the phenomenon are high quality, and datasets that do not match the phenomenon well are low quality. There are different ways to reason about how well data matches a phenomenon. In this section, we will discuss the distinction between uncertainty in data and errors in data and also distinguish between explicit quality and implicit quality.

Although **uncertainty** has many different interpretations [10], the distinction we will make in this dissertation is that uncertainty deals with the inherent limitations in data collection. Data will always have uncertainty, whether it is actively considered or not, but data with uncertainty can still provide insights into a phenomenon of interest. One common characterization of uncertainty is a measurement that provides a range of possible values a phenomenon could have instead of a single definitive one.

In contrast to uncertainty, **errors** are the result of a flaw or mistake in the data collection process. Erroneous data usually does not provide useful information about the phenomenon of interest.

There is another dimension of data quality that is tangential to uncertainty vs. errors. For this thesis, we will describe this as *explicit* vs. *implicit* quality [11]. **Explicit quality** is an expression of the data quality that has been explicitly recorded with the data or calculated based on the data. For instance, statistical measures like confidence intervals provide a range of possible values the measured phenomenon falls within, and statistical hypothesis tests can determine the amount of deviation from a distribution, for instance, to check for deviation from Benford's law [12], [13].

Such explicit definitions of quality can be powerful tools; however, they do not cover the full gamut of quality evaluation. One aspect that is missing is the human evaluation of data, specifically, the process that occurs when a researcher who is familiar with the phenomenon of interest and data collection methods evaluates how well the data represents the phenomenon. We consider the judgment a researcher makes on the quality of the data to be **implicit quality**. This distinction between explicit and implicit quality has been made before using other terms to describe it. Padilla et al. [14] describe *direct quantitative* and *indirect qualitative* uncertainty, which correspond to our explicit and implicit quality; McCurdy et al. [11] use the term implicit error for errors that are not recorded but are known by people observing the data; and Lin et al. coined the term *Data Hunch* [1] to capture the human knowledge of how and why data diverges from the phenomena it measures, which describes mismatches between the phenomenon of interest in the data based on implicit quality issues.

Figure 2.1 organizes these terms into four quadrants, defined by these two dimensions. It is helpful to separate explicit uncertainty from explicit errors since these are usually specifically designed to describe uncertainty or uncover a specific type of error. On the other hand, the distinction between implicit uncertainty and implicit errors is less useful

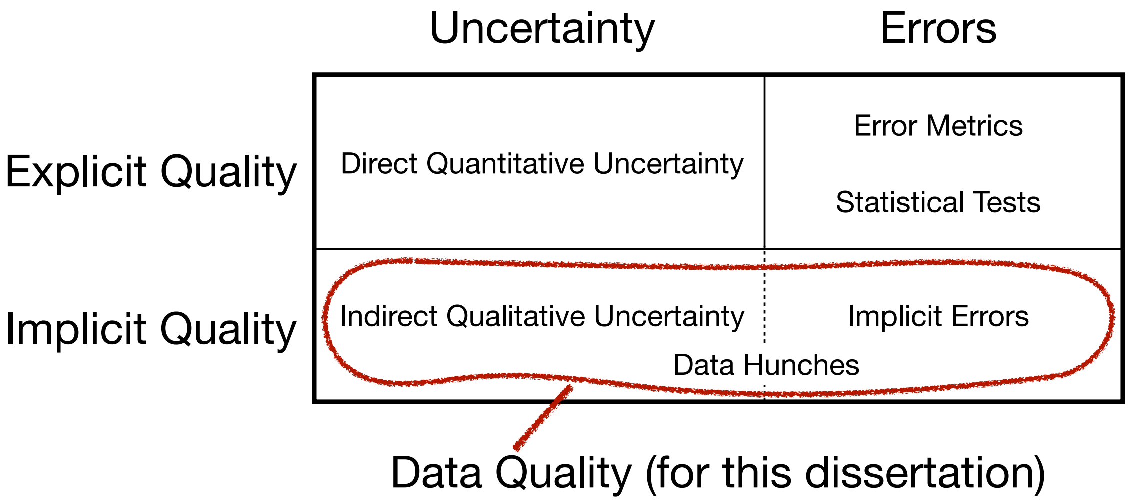


Figure 2.1: Different dimensions for reasoning about data quality with existing terms placed within. Explicit quality is defined within the data, or is the result of a function run on the data, whereas implicit quality includes human knowledge and judgment. The second dimension is between uncertainty and errors. This distinction is more fuzzy for implicit quality because it is primarily determined by the judgment the human makes.

for our purposes, which is to provide tools to help researchers evaluate their data. The difference between these types of measures is in the judgment that the human makes about the data. In other words, as they are reviewing the data, do they think, "I'm not so sure about these results. I know the device can fluctuate a bit when measuring this..." (uncertainty), or do they think "those combinations of values are not possible, something must have gone wrong here" (error). To make either of these judgments, the researcher must be able to effectively investigate their data while considering their background knowledge. This dissertation focuses on how visualizations can help scientists work with and understand their data so that they can apply their external knowledge to judge if it accurately represents a phenomenon. Thus, the focus will be on implicit quality and will generally use data quality and implicit data quality interchangeably.

2.2 Error Detection and Statistics

Work has been done outside the visualization community to more effectively detect errors in data. Specifically, for tabular data (spreadsheets), the detection of unintentional errors is a well-researched problem [15], [16]. Sukhobok et al. [17] identify different data anomalies in tabular data. The detection of formula errors [18] is useful when working with spreadsheets. Detecting structures that can lead to errors in tables [19] shares commonalities with our work done in Ferret since table clones are one of these structures. Beyond these structures, other methods for detecting errors in spreadsheets exist [20]–[24]. However, these methods have fairly narrow use cases. Specifically, unintentional errors (the target of all of these methods) do not always produce the same artifacts as intentional manipulations, so these techniques are typically not suitable for detecting manipulated datasets.

An alternative to our interactive visual system for finding data manipulations is to inspect data for statistical anomalies. Rules such as Benford's law [12], [25], which states that the leading digit is more likely to be 1 and then 2 than the large digits 7, 8, 9, have been used in domains such as accounting [26]. In these settings, financial fraud has been spotted by observing that Benford's Law was violated over a series of transactions. Statistical hypothesis tests can be leveraged to assess the significance of deviation from this expected distribution of leading digits [12],[13]. However, to apply Benford's law, assumptions must be made on the background (null) distribution. In particular, the distribution must span multiple orders of magnitude, which applies to some data, e.g., in astronomy or finance, but not in many others. For instance, the time in seconds to run a mile in a professional competition will almost surely start with a 2 (the current world record is 223 seconds).

Similar concerns exist in applying any statistical hypothesis testing method to look for anomalous patterns in data. All these methods start with an assumed background (null) distribution and look for a fixed type of pattern that may deviate from it. However, the *choice* of the background distribution requires domain knowledge and human judgment. Hence, one should not automatically or generically apply tests such as those for Benford's law. In the tabular datasets we analyzed, we found that we rarely could apply such tests.

Beyond statistical tests, there are numerical tests that do not check for statistically unlikely data but rather numerically impossible data. Notably, StatCheck [27] checks for internal consistency of statistical measures. StatCheck is used in some peer-review processes. However, it has received criticism due to concerns for its accuracy and its automated testing of papers [28]. The convenience of these automatic systems carries the risk that they will be used without providing authors the opportunity to respond to claims.

Furthermore, if the raw dataset has been manipulated before a correct statistical analysis is run, StatCheck cannot identify any errors. Another algorithm for detecting manipulation is described in Park et al.'s work [29] on detecting (and recovering) integer data when it has been multiplied by a nonintegral real number and has been rounded. Such techniques can be useful in the right situation, but they are limited to identifying a narrow set of problems.

2.2.1 Detecting Duplicated Data

One specific type of error in data is the duplication of parts of a dataset. This can occur both when data is copied as part of a data processing workflow and when data is being manipulated. The detection of text plagiarism is an active research field [30]. Even though plagiarism detection is not a solved problem, progress has been made, as is evident by the use of plagiarism-checking tools in many journals' review processes.

Detecting software plagiarism is a similar problem. The most widely used tool for determining software similarity is MOSS [31]. The authors of MOSS argue that it should not be used as an automated tool but rather as a way to surface potentially questionable data to reviewers, which is consistent with our motivations. Duplication of software can also occur when programmers copy/paste regions of code and then modify them. Detecting such copies is of interest to software engineers [32]. Similarly, in spreadsheet programs, an analyst may copy and paste a table, and just like in code, when one is updated, copies possibly should be as well. Hence, methods to detect such copies exist [33], [34].

Some forms of image manipulation create duplicated regions, such as the use of Adobe Photoshop's Clone Stamp tool. Image manipulation through duplication is a common problem in science [35]. Even though much of the work on identifying manipulations remains manual, recent work relies on machine-learning techniques such as CNNs [36]–[42]. CNNs are effective for images, where large datasets can be acquired or generated. However, tabular datasets are more varied in their structure. More importantly, the context associated with the data is also critical for interpretation. A tabular dataset could be completely plausible given one context and obviously manipulated in another. Such contextual understanding is difficult to encode in machine learning models today, especially with limited data, as is the case for manipulated tabular datasets. Therefore, we believe that a human-in-the-loop approach is needed when detecting manipulation in tabular data.

2.3 Visualizations for Data Quality Control

All of the explicit approaches for identifying errors in data can be useful in the right situation, but they cannot always be applied, and in the wrong situation, they can easily be misapplied, resulting in incorrect findings with the appearance of a definitive answer. John Tukey [43] identifies this weakness by claiming "Far better an approximate answer to the right question which is often vague, than an exact answer to the wrong question, which can always be made precise." This is one of the strengths of visualizations for exploratory data analysis — visualizations can provide good representations of data (approximate answers) so that researchers can synthesize that data with their background knowledge and ask the right questions. Exploratory data analysis is broader than just quality control, but this implicit judgment on the quality of data is certainly part of the process.

Research specifically on using data visualizations to help with data quality control has also been conducted for over a decade [44]–[46] in various domains. However, only in recent years has more careful thought gone into the strengths and weaknesses of visualizations to identify errors in data. Correll et al. suggest that sometimes univariate distribution visualizations look reasonable at first glance when, in reality, errors exist within the data [47]. McNutt et al. refer to these failures as visualization mirages and discuss ways to identify such mirages [48]. An important idea from these works is that different visualizations have different strengths and weaknesses when it comes to identifying errors in data. Correll utilizes this idea by creating raincloud plots that combine multiple univariate distribution plots that can complement each plot's strengths and weaknesses to reduce the likelihood that errors in data are missed [49]. Although this work is informative, it is focused on fundamental statistical charts. For this dissertation, detecting errors in microscopy data requires combining multiple types of data (such as images and metadata) that represent the same underlying phenomenon. The work of Zhang et al. [50] combines medical data from different sources into a single visualization to help clinicians treat patients with diabetes. This combination of data modalities into a single view helped those clinicians idenfity errors in the data. At a high level, this matches our approach in Loon (Chapter 3); however, the specifics of the data are completely different.

Many tools use visualizations to conduct exploratory data analysis. Even if quality control is not the sole design goal of these systems, it is often a component of it. It is useful to discuss general visualization systems designed for the types of data discussed in this dissertation. Specifically, cell biology data in Chapters 3 and 4 as well as tabular data in Chapter 5.

For tabular data, many visualization systems exist.The Table Lens [51] and Taggle [52] use a tabular layout to display aggregate tabular data as well as detailed data points with visualizations. Similarly, Domino [53] and SMARTexplore [54] link tabular data to visualizations. Finally, Taco [55], is a system for comparing similar tabular datasets.

Many visualization tools are tailored to the life sciences. For instance, genomics [56]–[58], connectomics [59]–[61], and histology [62]–[65] have utilized visualizations. Kerren and Schreiber argue that integrating diverse data modalities into a single visualization system is valuable [66]. One common data type from biology, imaging data, is well-suited for visualizations due to the visual nature of the data [67]. Visualizations of imaging data have been applied to subcellular structures [68], [69]. Visualizations for single-cell analysis can be utilized in different ways. Polyphony [70] uses visualizations to help researchers annotate cell types. This work illustrates why fully automated approaches are insufficient for this domain. Single-cell analysis can also be applied to drug screening by measuring the differences in cell populations when exposed to different drugs. Screenit [71] is a visualization approach for doing this type of high-content screening. More general frameworks for analyzing and visualizing single-cell data are also available. CellProfiler [72] is a system for cell image analysis, and CellProfiler Analyst [73] extends the base system to support data exploration with visualizations. SpatialData [74] links images of cells and their positions to their derived attributes. Finally, Vitessce [75] provides multiple linked visualizations for exploring and analyzing single-cell data.

Live cell microscopy tracks cell development under a microscope. The idea of visualizing these tracks has been around for at least two decades [76]. In the last decade, automation of data collection and tracking has become more prevalent [77], [78]. Our collaborators use commercial systems — such as Livecyte [79] and HoloMonitor [80] — as well as code developed within their labs. These systems use quantitative phase imaging (QPI) to measure the mass of individual cells [81]. Pretorius et al. have defined six classes of visualizations for live cell microscopy data [82].

Some cell microscopy visualization systems focus on studying how individual cells grow into organisms or the process of embryogenesis. The mechanisms that drive how humans and other multi-cellular organisms grow and develop have fascinated scientists for centuries; visualizations have been part of this journey from early on and continue to this day [83],[84]. Meyer et al. developed MulteeSum [85] to compare the embryo development of fruit flies. However, cell lineages are not collected or visualized in this work; instead, the spatial relationship of cells is utilized. Still, domain scientists value visualization tools for showing the actual tree information of the cell lineage [86]. CeLaVi [87] is one such tool that links a lineage diagram with 3D cell positions. LineageD [88] is another tool that also supports editing cell lineage labels. Finally, Pretorius et al. utilize cell lineages for analyzing cancer growth [89].

2.4 Guidance in Visualization

Guidance in visualization is related to how the visualization systems are designed to support users through their analysis [90]. There are many goals for such guidance systems, such as informing the user, mitigating bias, or reducing cognitive load [91]. Each of these goals aids in this dissertation's larger goal of helping researchers in their quality review of data. This guidance is especially related to the *Flexible-Structure* tension. In, Ceneda et al.'s characterization of guidance, they provide terms for the degree of guidance. A low degree of guidance is when a system doesn't suggest any actions but is still *orienting* the user. A system that is *directing* the user toward possible actions has a moderate degree of guidance. Finally a high degree of guidance is when a system is *prescribing* the actions of a user. Many visualization systems already exist that provide some degree of guidance [92]. Recently, progress has been made to facilitate the creation of guided vis systems [93] and evaluate them [94]. In this dissertation, we will discuss how the ability of visualizations to guide researchers can be beneficial for scientific data quality control.

CHAPTER 3

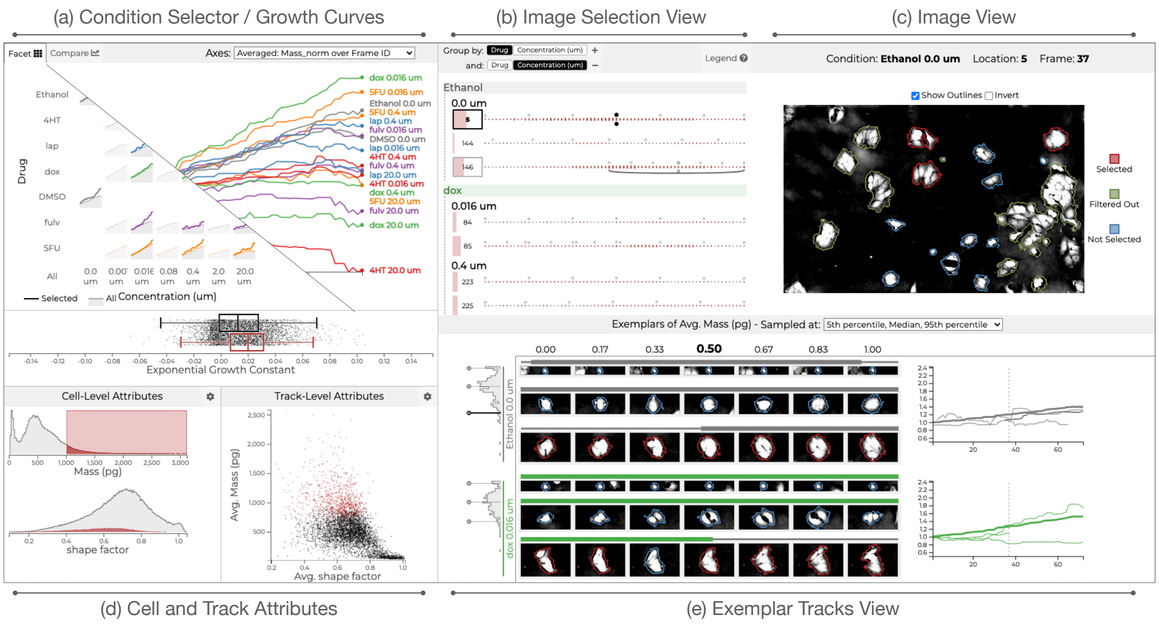
LOON: USING EXEMPLARS TO VISUALIZE LARGE-SCALE MICROSCOPY DATA

This work is based on our previous published work [7]. In this design study, we created a visualization system for cell microscopy data: Loon. Loon contains a variety of useful visualization views for our collaborators who study cell biology. The cell exemplar view (Figure 5.3e) is especially interesting for this dissertation since it employs several strategies for easing tensions while reviewing the quality of the cell microscopy data.

3.1 Introduction

Automatically acquired, large-scale microscopy data is an increasingly important tool in life-science research and medical practice. Areas such as brain connectomics create high-resolution images using electron microscopes of neurons and use segmentation to reconstruct the connectivity of the brain [59], [95]. Similarly, high-throughput screening to observe the effect of drug candidates on their cell lines is frequently used in pharmacological research [71]. For these approaches, the bottleneck has moved from acquiring image data to processing and analyzing the data. With the many thousands if not millions of images captured, these pipelines rely heavily on automatic image analysis processes, such as segmentation, to derive datasets of interest. However, completely automated setups have proven elusive, not least due to the heterogeneity of the images and biological structures captured [59], [82]. Analysts face many challenges, from conducting quality control of automatic processes (checking segmentation and tracking) to adapting the analysis for an experiment. To address these challenges, analysts need support through interactive visual analysis systems.

In this chapter, we introduce Loon, a visual analysis tool for a novel type of application: screening specific patient's tumor samples for a variety of cancer drugs using quantitative phase imaging (QPI) data. Our collaborators are developing this novel technology and


(a) Condition Selector / Growth Curves

(b) Image Selection View

(c) Image View

(d) Cell and Track Attributes

(e) Exemplar Tracks View

Figure 3.1: Overview of the Loon Visualization Tool. (a) The Condition Selector visualizes cell growth rates for different drugs at different concentrations using small multiple line charts in a matrix. Analysts can pick conditions that show interesting behavior for detailed analysis. (b) The Image Selection View is used to navigate images and visualizes aggregate cell and track (cells tracked over time) properties. (c) The Image View shows a selected microscopy image and the segmentation of cells. (d) The Cell and Track Attributes View shows distributions of and correlations between attributes of cells and tracks and serves as the primary means to define selections and filters. (e) The Exemplar Cells View shows cells extracted from the images and samples from a user-specified distribution. It also shows the growth curves for the condition and the selected cells.

the associated data analysis pipeline. The goal of these experiments is to rapidly (i.e., within 1–2 days) determine which known cancer drug inhibits cell growth, thereby taking a leap toward personalized medicine. The heterogeneous nature of the data — for different tumor types taken from different patients — requires a flexible visual analysis solution.

Our contribution is twofold: on the one hand, we contribute a design study based on a detailed analysis of the domain problem, which results in a functioning and deployed software tool. On the other hand, we also contribute a novel technique to visualize representative exemplars of cells by sampling cells along user-specified data dimensions. This approach enables analysts to both quickly validate preprocessing steps such as segmentation and tracking, and analyze the properties of cells in different conditions, thereby making visual analysis of microscopy data feasible even for large datasets. We argue that this technique can be applied broadly to other imaging/segmentation problems and beyond.

We validate our design using two case studies —– one focused on quality control, the other applied to data analysis — and through examples with two datasets.

3.2 Related Work

Our work is most closely related to visualizing cell microscopy data which is discussed in Chapter 2. Here, we briefly discuss other approaches to visualizing representative samples in datasets.

3.2.1 Exemplars in Different Contexts

Using representative examples to visualize classes of objects is a powerful approach. Many clustering algorithms, for example, can provide a "typical" member of a cluster that can then be used to represent a class. For example, Bach et al. [96] aggregate groups of temporal networks represented as adjacency matrices into piles based on clustering and show an example to represent the pile. Lekschas et al. [97] take a similar approach for genome interaction matrices, but also project the piles and the associated examples into 2D space based on similarity. In the context of computer vision, Carter et al. have developed an activation atlas that uses a structured layout of "activation images" and projects them into 2D space [98], leveraging exemplars to aid in the understanding of how a neural network sees.

These examples illustrate that exemplars are a powerful way to aid analysts in understanding a large dataset, but they are quite distinct from our approach and from our application area.

3.3 Biological Background

Cancer is a complex and dynamic disease with individual tumors presenting substantial genomic and transcriptomic heterogeneity, which makes it difficult to select the appropriate therapy for treating an individual cancer patient. Functional precision medicine seeks to guide treatment decisions using assays that measure the response of patient-derived tumor cells to candidate therapies [99]. Our collaborators are demonstrating the use of quantitative phase imaging as a method to measure the growth rate of individual cancer cells in response to chemotherapy. Technically, they use the exponential growth

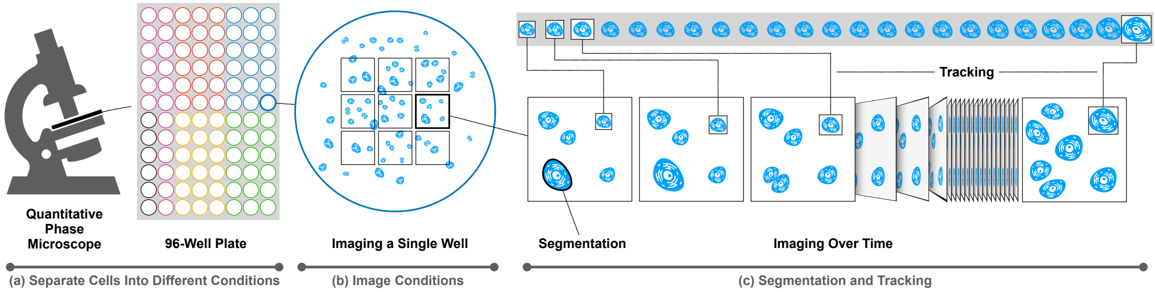

Quantitative Phase Microscope
96-Well Plate
Imaging a Single Well
Segmentation
Tracking
Imaging Over Time
(a) Separate Cells Into Different Conditions
(b) Image Conditions
(c) Segmentation and Tracking

Figure 3.2: The data acquisition process. (a) Cancer cells are plated in wells, and each well is treated either with a control substance or one of several cancer drugs (indicated as colors) at different concentrations. (b) For each well, multiple segments are imaged. (c) Individual cells are segmented. By repeatedly imaging each location as time passes, tracks of cells can be computed. The cell segmentation and tracking information can then be used to compute attributes such as mass and growth rates.

constant. For simplicity, we use the term growth rate in this chapter, which is similar conceptually. Effective chemotherapies reduce cell growth rates at low concentrations, indicating sensitivity. This work is therefore a step toward a functional precision assay.

QPI measures the phase shift of light as it passes through and interacts with cell mass. This phase shift is proportional to cell mass [100]. QPI has previously been applied to rapidly (within 5–10 h) measure chemotherapy sensitivity [101] and changes in cell phenotype associated with metastatic dissemination with single-cell resolution [102], [103]. QPI is, therefore, an ideal method for assessing the response of cancer cells to potential therapies.

In a typical experiment, illustrated in Figure 3.2, a cancer cell line derived from a tumor sample is plated at low density (cells covering 20%-50% of the cell culture surface) in a plate containing 24–96 individual wells. Each well is then treated with different drugs (or controls) at different concentrations. Then, multiple imaging locations for each well are selected. These locations are imaged repeatedly, once every 10–30 minutes, for a period of 1–5 days. We refer to an image in a time series as a **frame**.

To process the resulting phase images, our collaborators first use a rough segmentation to separate the cells from the background and average the background from 1000 images in order to do background correction. Next, they use a segmentation algorithm to segment cells and compute the mass of segmented cells by integrating the phase shift over the area of each cell. Cells are then tracked from frame to frame (see Figure 3.2) by minimizing the distance between cells in frame n relative to frame $n+1$, in a three-dimensional vector space such that the linearly independent components are distance traveled in x, distance traveled in y, and change in mass.

3.4 Dataset Description

The Zangle Lab and the Judson-Torres Lab are currently using Loon to analyze their datasets. In this chapter, we focus on one dataset from each lab as examples, although Loon is applicable to many similar datasets.

The main dataset we use to demonstrate Loon in this chapter and in the accompanying video and live demo is a breast cancer dataset. Worldwide, breast cancer is the most common cancer for women. Finding the right treatment for breast cancer could help millions of women who are diagnosed with breast cancer next year. In one experiment, the Zangle Lab exposed breast cancer cells to five drugs at six concentrations, as well as two control groups. This experiment used a multi-well plate to image cells at 864 locations. Each location was imaged once every 20 minutes for 24 hours, resulting in a total of 62,208 images at a resolution of 1920x1200. For this experiment, they chose not to segment individual cells but focus on cell clusters instead, as overall growth can also be tracked by observing the growth of clusters, and the segmentation is simplified. In total, this dataset contains 1,132,850 cell clusters and 166,015 cell tracks.

The second example dataset we describe in a case study is provided by the Judson-Torres Lab and is designed to investigate metastatic melanoma — skin cancer that has spread to other parts of the body. A tumor can contain subpopulations of cells that are either resistant or sensitive to treatment by a particular drug. In this experiment, the Judson-Torres Lab models this heterogeneity by combining two human melanoma cell lines, one that is resistant to drugs, and one that is sensitive. With this approach, the lab researchers can carefully control the ratios of the two cell types. In their experiment, they used a multi-well plate to separate four groups of cell mixture ratios (100%, 80%, 20%, 0% resistant cells). They exposed each group to the drug vemurafenib at five concentrations, as well as a control group with no drug. These conditions were imaged at 28 locations every 30 minutes for about 48 hours, resulting in a total of 2,716 images (at resolution 768x768), 439,699 segmented cells, and 111,151 cell tracks.

In general, these datasets contain thousands or tens of thousands of **images** (with resolutions between 400x300 to 1920x1200) that are assigned to **conditions**. Conditions are typically a combination of drug types and their concentration, or controls. Each image contains dozens or hundreds of **cells**. In a preprocessing step, our collaborators compute segmentation labels and derived numerical attributes, such as the area of a cell, the estimated mass of a cell, its position, attributes describing morphology (“shape factors”), etc. The position and mass of the cells are then used to track cells at the same location over time — resulting in **tracks** — for which the change in attributes can be measured and visualized. We supplement this data with derived data computed on the fly at both the cell (normalized mass) and track level (growth rate).

3.5 Collaboration and Domain Goals

Over the last 18 months, we collaborated with the Zangle Lab and the Judson-Torres Lab that use the same imaging methodology, but have slightly different focus areas; members of both labs are also coauthors of this chapter. Our primary collaborators are researchers in chemical engineering; their objective is to develop and validate quantitative phase imaging for the purpose of cancer drug screening. Their lab develops custom hardware setups and analysis software for QPI data analysis. Our secondary collaborators are researchers in a cancer center who use commercial hardware and software to analyze their data and who are primarily interested in the biomedical research questions that can be answered with this technology.

Eliciting domain goals is a key challenge in application-oriented work. Sedlmair et al. [104] discuss that “just talking to users [..] is typically not sufficient”. We used two complementary methods to understand user needs: first, we used contextual design methods [105] to develop an understanding of the domain and the analysis problems of our collaborators. The first author of this chapter embedded himself in the research group by attending all lab meetings for over a year, and by closely collaborating with the graduate students and the PIs. As the project progressed, the first author regularly collaborated with the domain experts to update their analysis code and aided them in unifying their processes across experiments, in particular between the different labs. Second, to jump-start our understanding of the domain problem, we used a creative visualization opportunity workshop [106] with five analysts for half a day. As a result of this workshop, we identified two high-level needs for our collaboration. First, they wanted to improve the quality control process of the lab, especially the segmenting and tracking of cells, since it is a labor-intensive task shared across the lab. The next high-level need we identified was comparing experimental conditions, that is, identifying differences in cell behavior when exposed to different drugs. Although not an explicit goal, the importance of cell growth curves in their analysis also became apparent at this workshop.

The outcome of this process is the codesign of the Loon visualization tool, which we developed to address the main domain-specific analysis goal, which is is to judge the effect of different drugs at different concentrations on patient-specific tumor cell lines. Together with other information, such as the clinical history of a patient and the side-effects of drugs, this information is then used for making treatment decisions. However, especially given that these systems are novel technology and given the heterogeneity of experiments and cancer, quality control and data preparation are also essential domain goals.

The quality control domain goals are concerned with validating various aspects of the experiment and the preprocessing:

- **QC-Segmentation**: Given the heterogeneity of tumor tissue samples and the variety of experiments, no two experiments are quite alike. In addition to different cell sizes and morphologies, our collaborators sometimes track clusters of cells, but sometimes also individual cells. All segmentation algorithms require parameterization, and human checks are necessary. A rough validation is possible just by looking at a few examples, but systematic errors that occur in edge cases are more difficult to spot with traditional methods, such as checking a sample of images. Hence, a key QC goal is to validate segmentation for a representative and diverse sample of images.

- **QC-Tracking**: Based on segmentations of individual cells, our collaborators compute tracks of cells (or cell clusters) that follow a cell over time. This process can also lead to errors, as the tracking can be lost if, e.g., the cell moves too much. Again, different parameterizations can lead to different results, in turn leading to the QC goal of validating cell tracking for representative and diverse samples of cells.

- **QC-Attributes**: Derived attributes, such as mass, size, or location in the experiment, etc., can also serve to validate all aspects of the experimental pipeline. To validate derived attributes, our collaborators need to explore these attributes and check for plausibility.

In addition to the quality control domain goals, we have identified the following data analysis domain goals:

- **DA-Cleaning**: As a preprocessing step in the analysis, the data needs to be cleaned of data points that do not correspond to cells that can be tracked. Common artefacts our collaborators want to remove are dust specks or debris that are mistakenly identified as small cells. Also, some tracks can be short, and our collaborators want to remove these since calculating growth rates is more accurate for longer tracks. Setting a minimum track length also reduces errors due to counting cells more than once in the subsequent analysis. However, these filters cannot be done completely automatically but instead have to be individually specified based on visual inspection.

- **DA-Conditions**: A typical experiment will analyze control conditions (no drug) and several drugs, resulting in dozens of experiments. As a first step, analysts want to analyze growth rates for all these conditions and pick the ones that look promising in terms of inhibiting growth at low concentrations, for subsequent detailed analysis.

- **DA-Growth**: The most important goal of the domain experts is to analyze growth indicators of cells of the different conditions. This analysis supports the ultimate goal to select medications that inhibit cell growth at low drug concentrations. Depending on the experiment, the experts might want to study normalized or absolute mass or area over time, while comparing different conditions. They also might analyze examples of fast-growing or dying cells in each condition.

- **DA-Heterogeneity**: Cancer cells are known to be heterogeneous in their genetic makeup and their responsiveness to therapy. A particular drug might be effective against one set of cells, but not against another set of cells, both of which can be found in the same tumor. To identify such cases, our collaborators need to study cells that both grow and shrink over time with respect to their attributes and their appearance.

- **DA-Communicating**: Since our collaborators are evaluating the overall method, they also want to communicate their findings in papers and presentations, including sharing plots of growth rates and images of exemplar cells.

3.6 Loon Visualization Design

We designed Loon to address the domain-specific goals discussed in the previous section. Ultimately, we can translate these goals into five visualization tasks. First, analysts need an interface to **select conditions** (Figure 3.1a) based on cell growth rates and contextual information about the conditions, which directly maps to the selecting condition requirement (**DA-Conditions**).

Second, for quality control goals, including evaluating the quality of segmentations (**QC-Segmentation**) and tracking (**QC-Tracking**), we need to **visualize the images and cells** (Figure 3.1b, c, and e). Visualizing cells is also essential for the analysis of heterogeneity (**DA-Heterogeneity**) and for communicating the effect a drug has on cells (**DA-Communicating**). The challenge with this task is scale: we need methods to visualize good representative subsets of tens of thousands of cells and images.

Third, we need the ability to **visualize tracks** (Figure 3.1b and e), both for quality control of the tracking (**QC-Tracking**) and to analyze cell growth (**DA-Growth**). Since tracks are deeply connected to the imaging data, visualizing both together is natural. Images of tracks are also difficult to curate manually; hence, a visualization of tracks based on cells also served the communication goal (**DA-Communicating**).

Fourth, we need the ability to **visualize cell growth rates** (Figure 3.1a), such as mass over time, area over time, or growth rates, and compare them between conditions in service of the main analysis goal: analyze cell growth for selecting drugs (**DA-Growth**).

Finally, we need the ability to **visualize distributions of and correlations between derived data items** (Figure 3.1d), to enable quality control for derived attributes (**QC-Attributes**), and to serve as an interface to **dynamically filter and select items**, in service of the data-cleaning goal (**DA-Cleaning**) and the ability to explore heterogeneity in cell lines (**DA-Heterogeneity**).

A deployed version of Loon can be accessed at `http://loon.sci.utah.edu/`, including a publicly available demo dataset and an option to log-in to access more datasets for our collaborators.

3.6.1 Selection View

Understanding the growth of cells under different conditions is a key component of our collaborators' workflow (**DA-Growth**). However, a typical experiment consists of many conditions that turn out to be of little value: of the dozens of conditions and controls, only a few will show responses to the drug. As data for individual conditions cannot be sensibly aggregated, filtering to conditions based on how promising they are is useful since it allows analysts to focus on the relevant data and alleviates scalability concerns (**DA-Conditions**).

Figure 3.3 shows a compact overview of all possible conditions, using a small-multiple grid view with area charts, inspired by Meyer et al.'s curvemap [107].

The area charts themselves plot cell growth over time; the charts are arranged in a matrix that plots drugs or control vs. concentrations so that charts on the right correspond to experiments with high concentration. We use color and vertical position to redundantly encode the type of drug. The colors are used consistently throughout the tool.

This view serves two purposes: to give an overview of the effectiveness of the drugs (a low or downward slope indicates inhibition) and to select conditions for a detailed analysis in other views. Analysts want to select conditions that inhibit growth at a low concentration, as these drugs are likely to be effective. Conditions can be selected individually, or in bulk by row or column.

We considered several alternatives to this view, including a line-chart plotting all conditions in the same plot, because such a chart would have a larger drawing area and would make comparisons between two lines more precise. However, we found that even the 32 conditions in our sample dataset lead to significant overplotting and that the structure of drugs and concentrations is hard to understand.

3.6.2 Comparison View

Complementary to the selection view, we provide a comparison view that can be used to plot various metrics, such as normalized mass, mass, shape factors, area, intensity, etc., over time (or frame). These metrics can be either aggregated (averaged) by conditions or plotted for each track individually. Figure 3.4 shows an example of the normalized mass plotted against frame ID inline charts, a key metric to address the "analyze cell growth" goal (**DA-Growth**). We use the drug-specific color scheme, and show labels for every line.

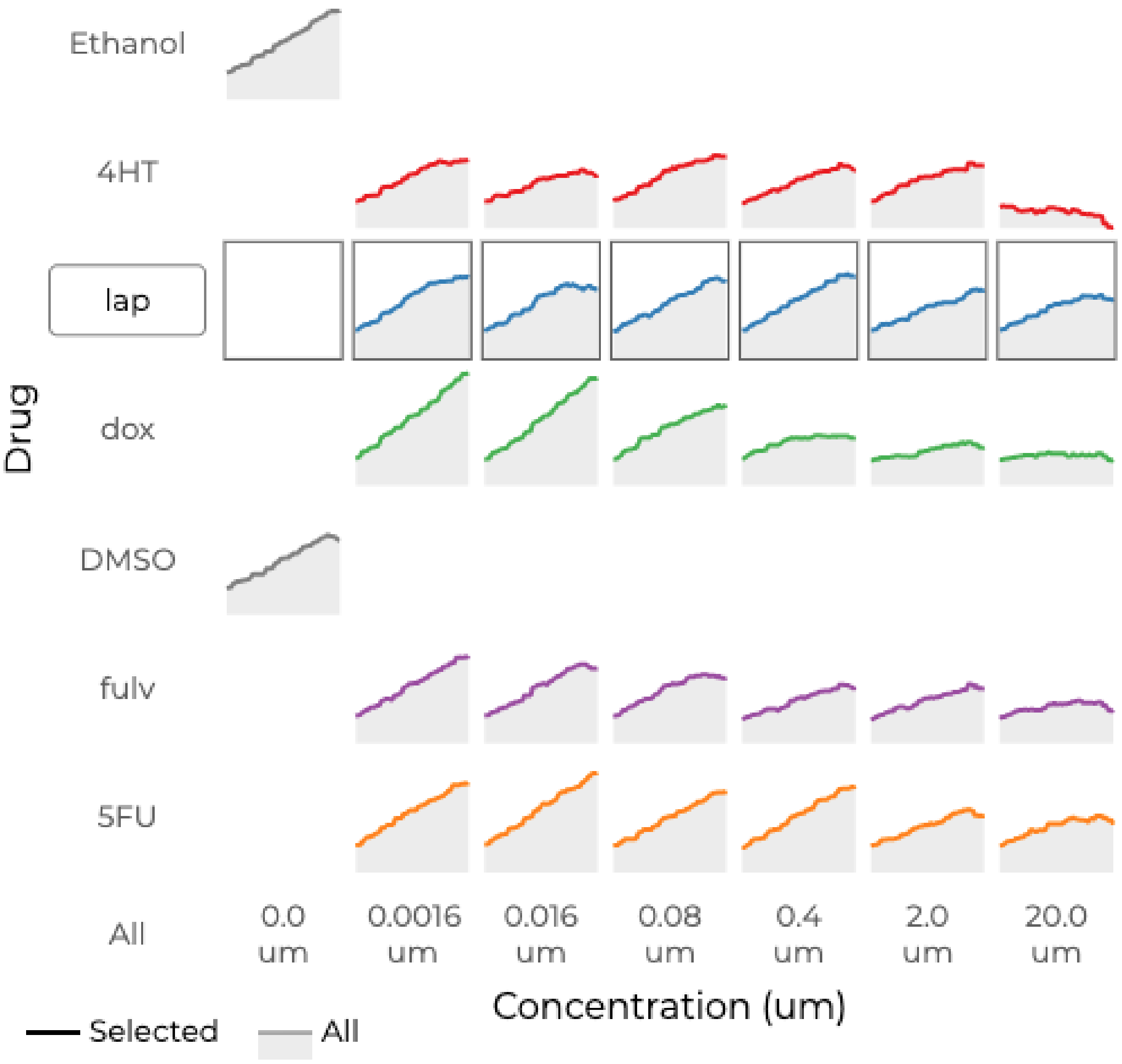


Figure 3.3: Condition Selection View. Small area charts show the growth rate for individual conditions. The charts are arranged by drug (4HT, lap, dox, fulv, and 5FU) or control (ethanol and DMSO) type vertically, and by concentration horizontally. Drugs are color coded; controls are shown in gray. Each condition can be selected individually, or conditions can be selected in bulk by row or column.

Compared to the selection view, this view enables a more detailed comparison between conditions, due to the larger screen area and the overlaid lines. This approach, however, comes at the cost of scalability and understanding the relationships between the trends and conditions.

By default, we apply a median filter to smooth out the appearance of the curves. Due to differences in focus settings, which are updated after every frame, the derived metrics, such as mass, can fluctuate, although this fluctuation is typically consistent between wells.

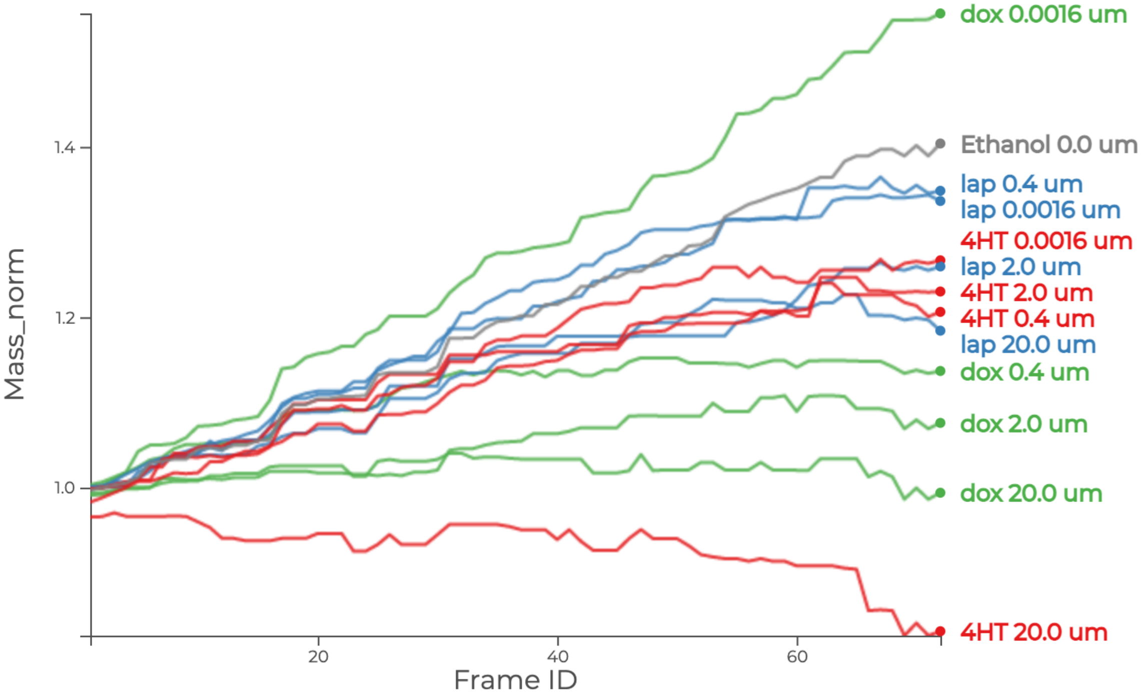


Figure 3.4: Comparison view showing frame ID, a proxy for time, plotted against normalized mass for a selected subset of conditions.

3.6.3 Image Viewer and Exemplars

A key component of Loon, and also our main technical contribution, is an interface that supports the ability to view and navigate microscopy images efficiently and to get an overview of representative cells for different conditions. This view is in service of the segmentation and tracking quality control goals (**QC-Segmentation**, **QC-Tracking**) but also for the analysis of cell heterogeneity (**DA-Heterogeneity**). The image analysis capabilities are made of three tightly coupled components: the image selection interface, the single image viewer, and the cell tracks display (Figure 3.1b, c, and e).

3.6.3.1 Image Selection

The image selection view, shown in Figure 3.5, enables analysts to navigate images by frame and experimental conditions. Each row displays the location ID and an array of ticks, where each tick mark corresponds to an image recorded at that location. Clicking on the tick loads that image. The size of the bar on the left and the size of the tick marks are scaled based on the number of cells at the location or the image.

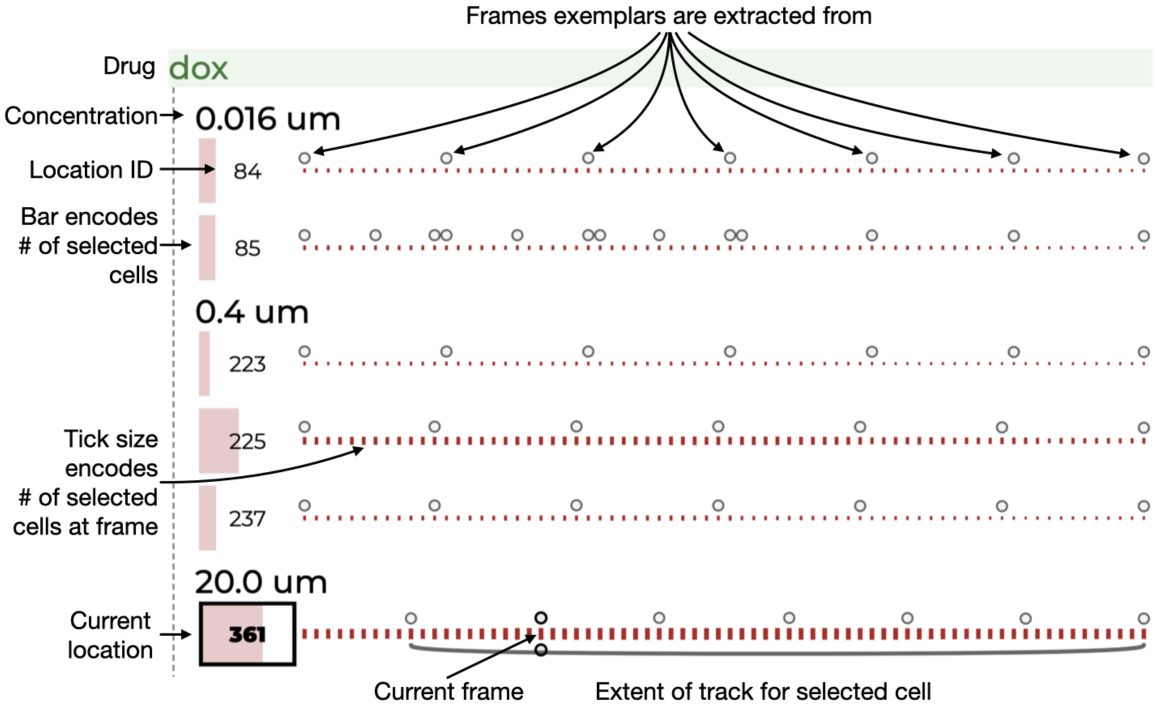


Figure 3.5: The image selection view enables analysts to browse images by condition (drug and concentration) and frame. It also displays the number of cells in each frame, and visualizes track lengths for selected cells.

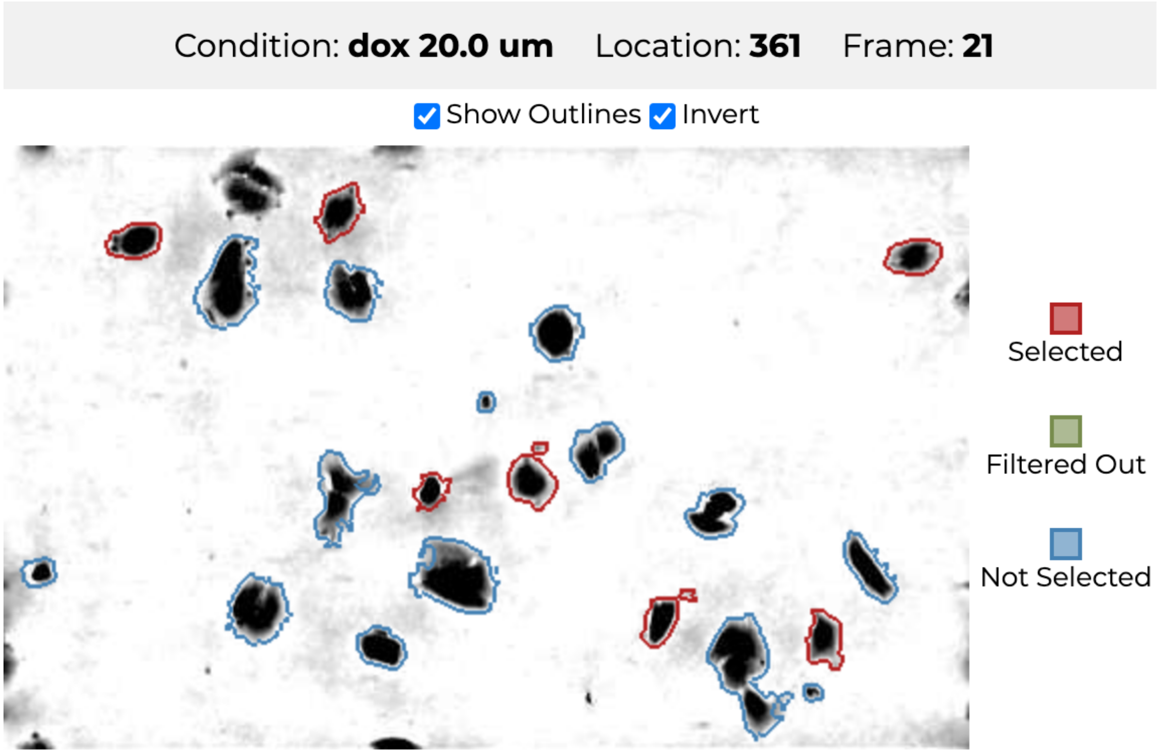
Small circles above the tick marks indicate that an exemplar has been extracted from that frame. A pair of larger circles above and below a frame indicates the currently selected frame. As cells are extracted into tracks, we show an arc from the first to the last time a cell appears in a track for selected cells.

3.6.3.2 Image Viewer

The image viewer (Figure 3.6) shows the raw image data of the cells along with color-coded segmentation boundaries. Red boundaries indicate that a cell is part of the current selection, blue are not. Green cells have been filtered out of the current dataset. Metadata about a cell is shown in a tool-tip. The images can be inverted, as illustrated in Figure 3.6 —– which can result in better discriminability between cells and background for certain datasets. Image frames can be navigated with keyboard controls.

3.6.3.3 Visualizing Exemplars of Cells.

The images can be accessed via the selection view and analyzed in the image view, but doing so is not scalable, neither for quality control nor for data analysis tasks. To address



Condition: **dox 20.0 um** Location: **361** Frame: **21**

☑ Show Outlines ☑ Invert

Figure 3.6: The image viewer, showing a color-inverted image and segmentation outlines. The segmentation outlines are color-coded by whether they are selected or filtered out.

this problem, we introduce exemplar cells that we sample from the large set of cells based on user-chosen attributes. Exemplar cells are extracted from the original image based on their cell region to create thumbnails. Although exemplars of cells have been used before, [71], [73], we are the first, to our knowledge, who sample exemplars systematically based on a user-chosen attribute.

Figure 3.7 illustrates the process. Given cells that have been assigned to tracks in a preprocessing step, we first group the tracks by condition. Then we sort within each group by a user-specified attribute, e.g., the growth rate, or the average mass. Next, we sample the tracks, for example, to show the median track and the tracks at the 5th and 95th percentile, thereby reducing all tracks for a group to a few representatives: a typical example and two outliers, in this case. Finally, in an optional step, we also sample exemplars over time, so that we always have the same number of cells for each track, no matter how long the track is. This approach results in a desirable compact view, but understanding the length of a track is an important aspect of the analysis, and, hence, we encode the time-frame over which the track is present in a bar on top of the exemplars.

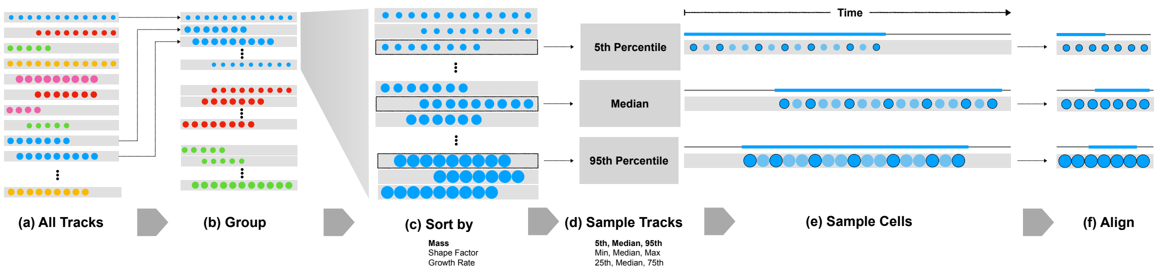


Figure 3.7: The process of sampling exemplar cells based on a user-selected attribute. (a) Tracks with cells as extracted during preprocessing. (b) We first group cells by conditions, indicated as colors. (c) Next, we sort within each group by a user-selected attribute. We use mass in this example, but other attributes such as growth rate or a shape metric are equally useful. (d) We then sample tracks based on a user-specified rule. In this example, we extract the median track, and the tracks at the 5th and 95th percentile according to the metric specified in the previous step. (e) Optionally, tracks can also be sampled over time. (f) The sampling makes it possible to reduce tracks to a fixed number of cells, which results in a compact layout. The "time-window" of the track is shown above the cells with a filled-in bar.

Figure 3.8 shows an example of how this concept is implemented in Loon and introduces several complementary views. By default, seven images are evenly sampled from the cell's lifetime, although analysts have the option to show all the cells in a track. The extent of a track's lifetime is visualized with the color-coded line above each track. On the left, we show a histogram that visualizes the distribution of the tracks with respect to the user-selected attribute (average mass in this case). A set of pins indicates the position of the exemplars in that histogram. Analysts can also retrieve exemplars at any location of that histogram; an example for a very large cell group is shown in Figure 3.8, indicated by the push-pin. This feature enables analysts to interactively check tracks anywhere in the distribution, e.g., to analyze outlier cells (**DA-Heterogeneity**), or to check the segmentation in edge cases (**QC-Segmentation**). Finally, a line chart to the right of each group shows the average growth rate for the condition and the growth rates of the exemplar cells. This view enables analysts to judge which cells might be responsible for growth or shrinkage. Analysts can also bookmark cells they encounter at any point in their analysis, which are then shown as exemplars at the top.

We found that this condensed display of exemplar cell tracks is on a sweet-spot with regard to spatial efficiency and supporting the goals of our collaborators. In earlier iterations, we explored other ways to present these cell tracks, which are still accessible in the interface and may be useful in special cases. One alternate mode samples tracks but does not sample cells over time. We quickly found that the temporal sampling rate is too fine-grained and that cells mostly do not change much between two frames, and, hence, a view like this shows too much detail for most analysis tasks. We also initially attempted to visualize all cells in all tracks, aligned by absolute time, as shown in Figure 3.9. This alignment leads to more whitespace and tedious scrolling, as well as a significant slowdown with regard to the rendering performance.

3.6.4 Interaction and Metadata Views

Various domain tasks rely on the analysis of metadata, including quality control (**QC-Attributes**), data cleaning (**DA-Cleaning**), and analysis (**DA-Growth**). Our example datasets contain metrics for mass and size, as well as a shape factor that describes the "circularity" of a cell, in addition to track-level attributes such as growth.

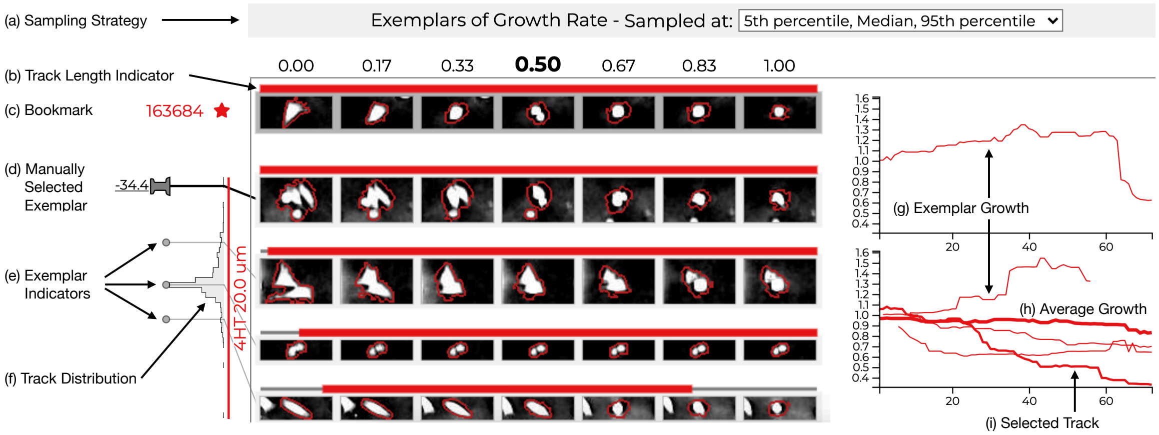


Figure 3.8: Visualizing Exemplars. This exemplar view shows one bookmarked track (c), one manually selected exemplar (d), and three automatic exemplar tracks (e), sampled from all tracks based on a user-defined attribute and sampling strategy. (a) This example is sampled by the growth rate. (f) To the left of the exemplars, we show a histogram visualizing the distribution of tracks for the selected attribute (mass in this case). (e) Pins indicate the position of the exemplar tracks in this distribution. (d) Analysts can also click the histogram to manually select an exemplar and add it to the view. Such exemplars are shown as a push-pin in the histogram. (b) Colored bars on top of the exemplar tracks indicate the length of the track. The line charts on the right shows the (h) average growth of the condition, and (g) the growth rates of the exemplars. (c) Finally, any cell can be bookmarked. Bookmarked cells are shown as an exemplar on top, independent of their group/condition. Several cells here are shrinking in mass. The pinned track (d), drawn from the extreme end of the distribution, shows an example of a cell cluster breaking apart, which contributes to this extreme drop in mass (i). All other cancer cells seem to be shrinking naturally.

Figure 3.9: An early attempt to visualize tracks of cells without sampling.

Loon provides several standard statistical plots, such as scatter-plots, histograms, kernel-density estimate plots, and dot plots with box plots overlaid, along with a data selection interface, to visualize these data dimensions (see Figure 3.10a). Loon also provides the ability to facet each plot by condition. These types of plots can be used to visually inspect if the derived attributes have a reasonable distribution (**QC-Attributes**).

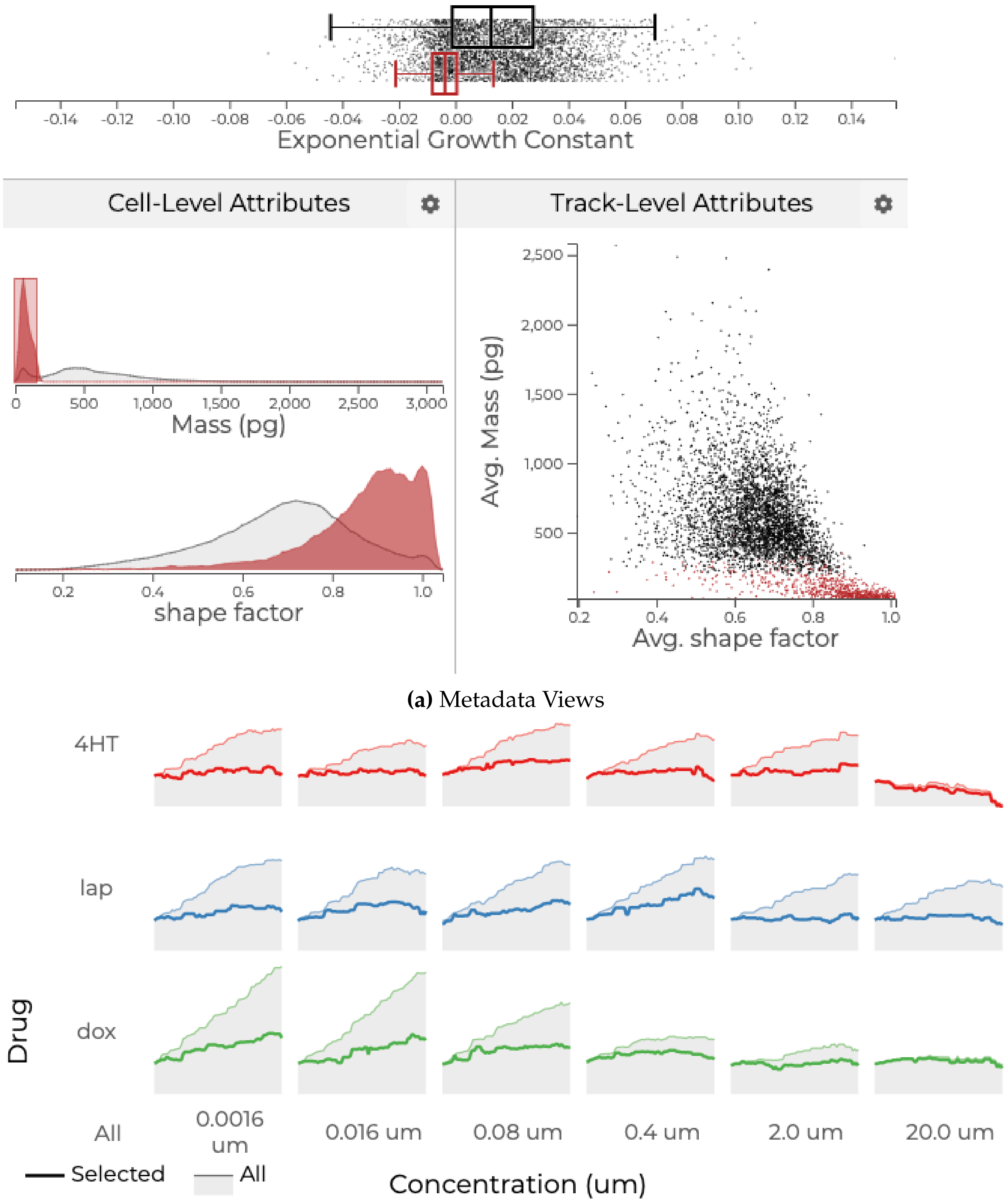
In cases where the data is showing unexpected values, these plots serve as a selection and filtering interface to inspect those values more closely. Using 1D and 2D brushes will update all other plots to indicate the current selection. For example, Figure 3.10a shows a brush made on the growth rate, which is also applied to the growth curves in Figure 3.10b as overlaid curves. At any point, a selection can be converted into a more permanent filter. Filters completely remove data outside the current selection. Importantly, this conversion will refer only to selections at the track level. In other words, it will not filter out parts of a track; tracks are either fully included or excluded.

Since interactions between multiple views are tightly coupled, we currently show all windows (with the exception of the condition selector/growth curve). In the future, we plan on adding support for window management, so that users can customize their layout.

3.7 Implementation

Loon is implemented as an open-source web-based tool. The source code is available at `https://github.com/visdesignlab/loon/`. Loon uses a Flask server as its backend. The various charts were custom-built with the D3 JavaScript library [108].

Our collaborators store their processed data, including the images and derived datasets, as MATLAB files on Google Drive, which provides free unlimited storage to educational organizations. Since a single experiment can produce in excess of 500GB of data, and



(a) Metadata Views

(b) Effect of Brushes.

Figure 3.10: Metadata views and selections. (a) Histograms, dot plots, and KDE plots visualize metadata. A brush on mass is reflected in all other plots, including in (b) condition selection views' growth curves. Here, a new growth curve, only for the selected elements, is overlaid.

the goal is to eventually run experiments daily, we were hesitant to duplicate this data on a dedicated server. Instead, we decided to dynamically query the data through the Google Drive API, but the performance of retrieving images is problematic. To address this problem, we implemented several strategies. First, we found that for visual representation, we do not need the lossless image files. Slight downscaling and conversion to JPG reduced the size by about 100-fold. Next, we introduced three levels of caching. First, all metadata is cached on our webserver and automatically updated it if the data on Google Drive has changed. Second, the IndexedDB Web API is used to store data on a web client for both image metadata and the images. To minimize the amount of data transferred, this caching is done dynamically as data/images are requested. A downside of this approach is that an initial load of the page will take on the order of minutes, but a reload will be on the order of seconds. Similarly, actions inside the tool that require more images will be slower, but reloads with the same data are fast. The final level of caching occurs in JavaScript memory for images. Since some image processing/extraction happens on the fly, we use image Blob objects and keep a set number in memory, which allows us to achieve frame navigation at rates that appear to animate cell growth, for instance.

3.8 Case Studies

To demonstrate the utility of our tool, we provide two case studies, which were conducted by domain scientists and with real data and real analysis problems [104]. Our collaborators were closely involved in the design of Loon and are also coauthors of this chapter. Given issues with experimental demand characteristic effects that arise in close collaborations [109], we refrain from reporting subjective assessments of our collaborators and rather report factually on two analysis scenarios and how they were conducted with Loon, and describe the difference in capabilities relative to our collaborators' previous process.

Our collaborators used successive iterations of Loon over 18 months and codeveloped their analysis pipeline to both provide all the necessary data items for Loon and address analysis problems that were discovered when using Loon. For example, when visualizing the physical location of cells, our collaborators found positions far outside the expected bounds as well as points with a shape factor above one, when one should be the maximum

(**QC-Attributes**). This problem turns out to be an error in the data preprocessing and needs to be corrected in the pipeline.

3.8.1 Comparing Drug Effects on a Breast Cancer Case

We report here on an analysis of the breast cancer dataset that was conducted by Dr. Zangle, while the first author observed remotely and recorded the session. A detailed record of all stages of the case study is shown in the supplementary material. Initially, Dr. Zangle viewed the normalized mass growth curves in the condition selection matrix (**DA-Growth**). There he identified a dramatic change for the drug 4HT between 2 and $20\mu M$ (see Fig 3.3). He commented that this change appears to indicate that 4HT is cytotoxic at $20\mu M$, but has little to no effect at lower concentrations. However, the cells with negative growth at concentration $20\mu M$ may not actually be dying, but instead the problem may be one of segmentation or tracking problem (**QC-Segmentation**, **QC-Tracking**).

To explore this question. Dr. Zangle selected 4HT at concentration $20\mu M$ (**DA-Conditions**). Looking first at the single image view, he identified cells that appeared to be dying. After bookmarking a particularly interesting cell, he confirmed with the mass over time curve that the cell was dying at the end of the track (**DA-Growth**), as shown in Figure 3.8.

Next, he changed the track attribute to view exemplars sampled from the growth rate. Then, within the exemplar tracks view, he inspected various cells with negative growth rates from the automatically selected exemplars as well as by manually pinning other cells with negative growth rates from the histogram. While Dr. Zangle inspected the negative mass over time plot, he became curious to find out whether the cell was dying. To answer this question required analyzing the image data and segmentation. In one case, Dr. Zangle viewed the segmentation of a cell cluster, shown in Figure 3.8, and noticed that one cell drifted away from a cluster until it was no longer considered part of the same region by the algorithm. He noted that this complication in segmentation explained part of the dramatic cell mass decrease (**QC-Segmentation**). Other cell clusters, however, showed clear segmentation and negative growth rate, and the cell morphology clearly indicated cell death. All of this combined does seem to indicate that 4HT is cytotoxic at $20\mu M$. When communicating this insight, for instance in a paper, our collaborators would include selected exemplar images to illustrate what is happening to cells exposed to that drug

(**DA-Communicating**).

Another question Dr. Zangle's team frequently asked is what is causing differences in the average growth rate between conditions. The differences may be due to every cell growing more slowly, or from a heterogeneous response, with some cells dying, while others are unaffected (**DA-Heterogeneity**). Dr. Zangle was interested in this question for 4HT at $2\mu M$. Here we see a much more moderate decrease in average growth rate (**DA-Growth**). To begin analyzing the heterogeneity, he selected three conditions, 4HT at $2\mu M$, $0.0016\mu M$, and DMSO, the control for this drug. Being interested in the effect of growth rate on the cells, he chose growth rate as the attribute driving the sampling of exemplars. In the exemplar tracks view, he first looked at the growth rate histograms for each of these conditions (**DA-Heterogeneity**). He found no significant difference among these three in the shape of this distribution. Next, he changed the sampling strategy to include more exemplars per condition. These exemplars, along with some manually pinned exemplars, provided a small group of cells to inspect when looking for cell death. When trying to identify dying cells, Dr. Zangle looked at the image characteristics as well as the growth rate to determine if a cell was actually dying or not. After inspection of exemplars from these three conditions, he found no noticeable difference in the number of cells dying (**DA-Heterogeneity**). After this analysis, he concluded that the likely explanation is that these three conditions do not exhibit significant heterogeneity.

3.8.2 Refining The Tracking for a Melanoma Dataset

In recent years, great strides have been made in the treatment of metastatic melanoma — the deadliest of skin cancers. These advances include targeted therapies against common oncogenic mutations and immunotherapies. Both therapies can be effective, but neither works with 100% efficacy [110] due to the heterogeneous nature of these types of cancer cell populations. Determining which therapy will ultimately be most successful in preventing metastasis requires a better understanding of the effect drugs have on these different subpopulations of cells. A functional precision medicine approach that predicts response to targeted therapies could aid oncologists in this difficult and critical decision. To answer these questions, Dr. Judson-Torres' lab is running QPI experiments on their commercial platforms.

They started a collaboration with Dr. Zangle's lab to also integrate their dataset with their analysis pipeline and with Loon. Since this is a new dataset for our collaborators' data processing workflow, a key initial task is quality control. This case study describes a multiweek process, where the lab analyzed the data, found deficiencies, refined it, etc.

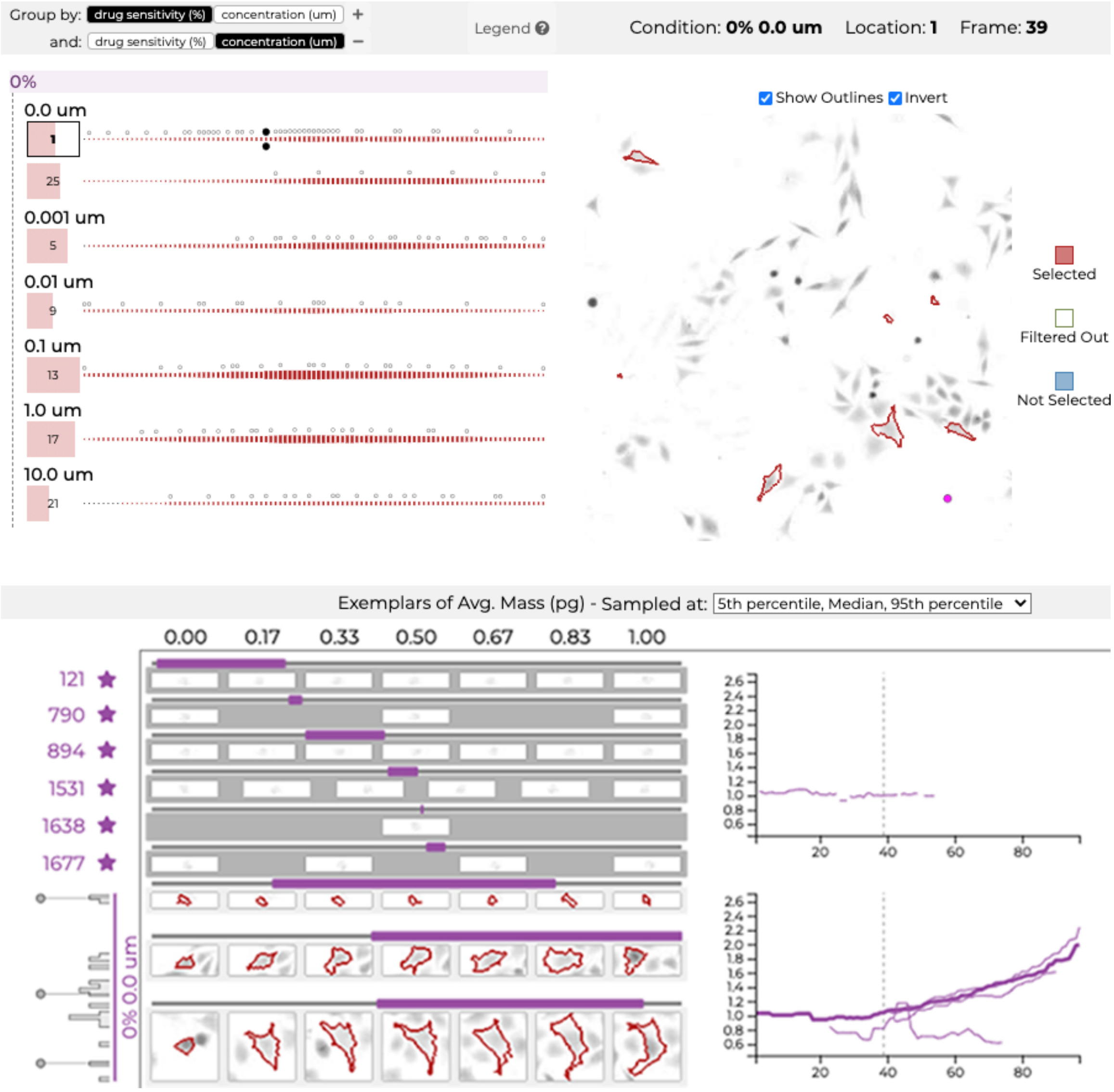
As a first data cleanup step in Loon, Dr. Zangle filtered to show only longer tracks (**DA-Cleaning**). In the single image view, he noticed that only a few cells were being tracked for a long duration, as indicated by the color-coded outlines. Such short tracks are indicative of tracking problems. He then selected a few locations with the image selection view and compared tick sizes across all locations to ultimately conclude that cells were commonly tracked only briefly (**QC-Tracking**). To improve tracking, our collaborators have many parameters they can adjust. For instance, if a cell moves "too far", or changes mass "too much" in a single frame, it will not be tracked. This threshold of "too much" can be adjusted for different experiments. The image view is useful here since it shows the position and mass of these cell segments. Our collaborators use this view to find cells that are not tracked and then adjust their thresholds (**DA-Cleaning**). They use a similar threshold to distinguish between debris that has been segmented in the image and actual small cells.

Adjusting these thresholds improved the tracking for this dataset, but the improvement did run into an upper limit. When filtered to longer tracks, Dr. Zangle noticed that the cells in these tracks did not move much, whereas the cells that were not tracked well were more motile (**QC-Tracking**). This particular dataset is challenging due to the variety of cell density and motility. He concluded that they would have to make larger changes to their tracking algorithm, possibly by connecting tracks after the first pass of the original algorithm.

For this QC task, our collaborators commented that the bookmarking feature was especially useful for keeping track of shorter tracks and comparing values of endpoints (**QC-Tracking**). Figure 3.11 shows six tracks that ideally should be connected into a single one.

3.8.3 Loon Feedback

We asked both labs to briefly explain how Loon changes their workflow. Dr. Zangle responded:



Group by: drug sensitivity (%) concentration (um)
and: drug sensitivity (%) concentration (um)

Condition: **0% 0.0 um** Location: **1** Frame: **39**

☑ Show Outlines ☑ Invert

Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile

Figure 3.11: Loon showing tracking problems in the melanoma dataset. The bookmarked short tracks are in reality a single cell that has not been tracked adequately. This cell would be more accurately represented as a single longer track.

> Loon enables us to view cell segmentation and morphology alongside mass over time tracks, which would be extremely tedious with our previous approach. This capability improves QC because we can quickly spot-check segmentation and find errors and how they interact with cell mass measurements. This also helps with interpreting data because we can see how cell morphology and motility are related to tracked parameters.

Dr. Judson Torres similarly commented:

> The workflow we find most useful in Loon is the ability to quickly visualize the mass, tracking information, and exemplar images of object groups based upon feature filters. The dry mass, track length, and, of course, the actual image of the segmented objects are all critical for conducting quality control — identifying which objects represent cells or colonies of interest and which are artifact from the segmentation. This process has previously been extremely tedious with our existing approaches, so much so that we had started to just accept a certain amount of noise in our data due to segmentation artifacts. In addition, one of our main uses of QPI approaches is to identify groups of phenotypically distinct cells within heterogeneous populations. Our previous approaches first required analyses of feature-level data to identify distinct groups of objects followed by cross-referencing representatives of each group in the images to determine whether the observed feature clusters were real or artifact — a time-intensive process. The ability to select groups based upon features and, in real time, visualize exemplars from each group enables us to rapidly identify and verify distinct groups of cells.

3.9 Lessons Learned and Reflections

3.9.1 Transferable Contribution

Even though we present a design study — a visualization tool that addresses the important but narrow problem of aiding in selecting cancer drugs based on microscopy experiments — we argue that our main technical contribution, sampling exemplars along selected data dimensions, transfers to a broad range of problems.

In general, we argue that visualizing through exemplars could potentially be applicable once two criteria are met. First, the dataset has to be large enough that it is infeasible to manually inspect every data point. Second, the data items have to contain rich contextual information that cannot be summarized. In our case, we consider our cell images as these data points. However, the same basic idea could be applied to many situations.

Obvious related applications that fulfill these criteria are in other image segmentation and microscopy scenarios, such as in connectomics or radiology, where both our patch-extraction and sampling could be applied. Different examples include time-series data, where periodic signals, such as heartbeats or electric data measured with an oscilloscope [111], are likely amenable to the same approach. Given the right metrics, even the analysis of cliques or other structures in a large network could benefit from exemplars [96]. Finally, avid bikers may be interested in looking back at their rides over the past few years. A traditional visualization approach would show distributions of ride duration, distance, and speed. An exemplar-based approach would pull out specific rides based on some criteria, such as distance traveled, and then present entry points to inspect the entire ride. With the richer context, our bikers are able to observe different aspects of the data — like see a familiar bike route, or recall an exceptionally grueling ride.

3.9.2 Line Chart Design

Line charts are commonly considered a scalable solution for plotting multiple time series at the same time. However, we found that an overlaid line-chart was unsuitable when the goal was to make a series of comparisons based on categorical information associated with the lines, such as the drugs and concentrations, in our case. Here, faceted small multiples were much more useful, especially as the faceting can be done based on the categorical variables.

3.9.3 Codesign of Method and Visualization

This design study was unique in that it contained strong aspects of codesign between the visualization researchers and the domain experts. QPI for drug screening is a novel method, and the Zangle lab is at the forefront of developing this technology. As illustrated in the melanoma case study, the lab is not treating the visualization as an end-point, but rather as a means to understand their data **and** their process. Just as the visualization tool developed over this 18-month collaboration, so too did the analysis process. We found that the immersive approach we took over this time was especially valuable.

3.10 Discussion

3.10.1 Performance

Loon can be slow when a dataset is used for the first time. Even though we implement various caching strategies (see Section 3.7), a large amount of data needs to be transferred from a relatively slow data source. To address this problem, we will likely have to abandon our initial design goal of retaining the data on our collaborator's Google Drive so that Loon integrates nicely with their workflow. Image-pyramid-based approaches have been successfully used for microscopy visualization [112], and combined with other precomputations and efficient data structures, we expect to achieve significantly improved performance.

3.10.2 Sampling Strategies for Exemplars

Our current sampling strategies for selecting exemplars are simple and thus easily understood and explainable. However, we believe adding more sophisticated sampling strategies would be valuable, based on multiple attributes at once, or maybe using the depth measure from contour boxplots [113] to sample based on growth functions. We could also use representative examples based on clustering algorithms, or extreme examples based on archetypal analysis [114]. Also, we could explore sampling strategies that do not rely on derived attributes of segmentations. These strategies could be especially useful in scenarios where segmentations are invalid, but have reasonable derived attributes.

3.10.3 Evaluation

We considered other evaluation strategies, in addition to our case studies, that demonstrate the value of our contribution. However, we came to the conclusion that other approaches would be problematic with respect to their validity, or test aspects that are not relevant to our contribution. For example, whereas usability testing would be useful to identify usability problems and bugs, it does not result in generalizable knowledge about our contribution. A comparative evaluation, either based on insights or quantitative measures, is infeasible with a specialized tool, since the number of users is limited, and participants will possibly be biased because they know the experimenter and the desired outcome. A user satisfaction survey likely suffers from similar biases. Hence, we chose case studies to demonstrate that our tool is useful for the domain goals we claim and use careful arguments to justify our technical contribution. However, in the future, we believe that an empirical evaluation of our main technical contribution (exemplars) could be designed, but such a design would require carefully untangling our approach to exemplars from the larger interface. We argue that case studies are a good first approach to validate that the idea is compelling, but that an empirical study of exemplars and alternatives is beyond the scope of this work and should be conducted systematically by an outside party [115].

3.11 Conclusion

In this chapter, we have presented Loon, a visualization tool for analyzing the effectiveness of cancer drugs on individual cell lines based on QPI microscopy data. In addition to the tool, we also contribute the concept of analyzing large datasets with complex item characteristics by exemplars. We argue that our tool is a successful design study in a high-impact area and that our technical contribution transfers to various other scenarios.

These exemplars are one way to help improve the quality control of data. Showing the rich detail of individual data points while embedding them in their distribution provides a balance between detailed information and broader understanding (*Trees-and-Foreset*). Automatically selecting reasonable data points while supporting the free-form selection of data points provides a direction for review without being prescriptive (*Flexible-Structure*). Finally, providing different modalities of data together in the same chart facilitates the synthesis of complex data (*Trees-and-Foreset*). This idea of combining different data modalities into the same visualization will be expanded on in the following chapter.

CHAPTER 4

AARDVARK: COMPOSITE VISUALIZATIONS OF TREES, TIME-SERIES, AND IMAGES

This chapter is based on our work describing Aardvark [8], to appear in IEEE VIS 2024. This design study is a continuation of the previous collaboration with microscopy and oncology researchers. However, the focus of this project is on the combination of different data modalities into composite visualizations. These composite visualizations help synthesize the different data modalities for a more effective review of data quality.

4.1 Introduction

Time-series cell microscopy is an essential method to understand the behavior of cells in the life sciences and plays a large role in the study of cancer. Cancer is a disease

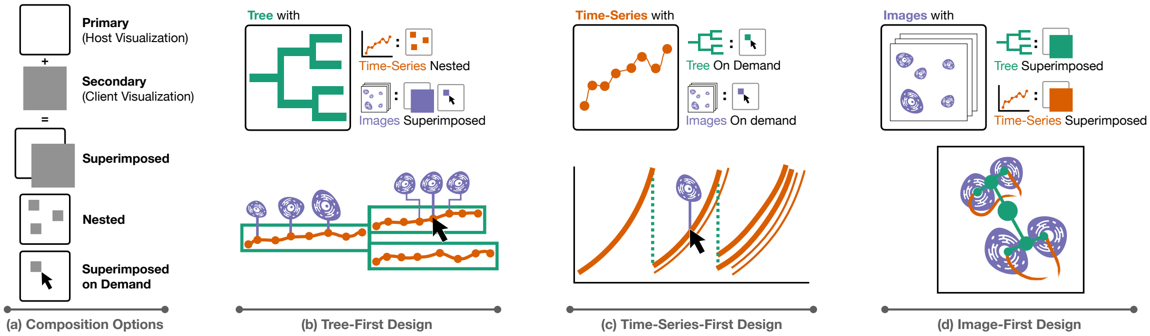


Figure 4.1: Composition options and designs for cell microscopy visualizations. (a) Composition options for integrating visualizations in one view. First, a primary data type and visual encoding are chosen as the host visualization, and then additional data is added via composition as client visualizations. (b) The Tree-First Design uses a node-link diagram for the primary encoding (green), nests time-series in the nodes (orange), and superimposes cell image data at regular intervals and on demand (violet). (c) The Time-Series-First Design employs a line chart for the time-series data (orange) and superimposes topological data (green) and image data (violet) on demand. (d) The Image-First Design superimposes tree data and time-series data (cell movement) in the same coordinate system as the images.

caused by the unregulated growth and division of cells. One way to better understand this disease is to collect data with a microscope on cancer cells as they grow and divide. Microscopic **images** of the cells growing and dividing form the raw data of many derived analysis methods. Once cells are segmented to find their boundary and tracked over time, **time-series** data of derived cell attributes can describe how cells change over time. Finally, identifying cell divisions and recording the parent-child relationship produces an inheritance **tree** that describes the lineages of cells.

Collecting such data can be challenging; it requires expertise in constructing and programming automated microscopes, physically preparing experiments in the lab, implementing machine learning and algorithms to perform image analysis, and analyzing the derived data. As with any complex data acquisition and processing pipeline, things can go wrong at any of these steps. Consequently, determining if and where errors exist, analyzing the results, and communicating those results can be challenging.

For this complex multivariate data, it can be difficult or even impossible to form conclusions using one type of data. For example, if a sudden decrease in cell size is observed, it is impossible to determine if that change represents a real biological phenomenon (the cell dying) or an error in the data (a segmentation failure) without reviewing the image data. Although an expert can often distinguish between these possibilities when reviewing the image data, matching the observed effect in the time-series data to the right cell in the right image is a tedious process with current methods. When constructing tree visualizations for lineage data, we run into a similar issue; matching nodes in the tree to time-series or image data is difficult. In our work, we tackle these issues.

There are two main contributions in this chapter. First, we describe design principles for composite visualizations of images, time-series, and tree data by choosing a primary encoding and embedding the secondary data types within it. Next, we apply these design principles to our data (cell images, time-series data, and lineages) and implement it in an open-source visualization system we call Aardvark. Aardvark includes three novel visualization designs that make the correspondence between data types easy to understand. We demonstrate the utility of this technique for data exploration, quality control, and communication with three case studies of cancer cell microscopy data.

4.2 Related Work

Our work is related to visualizations of cell microscopy data (discussed in Chapter 2), multivariate trees, and imaging data. To our knowledge, no prior work combines all three of these modalities into one framework.

4.2.1 Tree Visualizations

The use of trees in visualizations are a well-studied and active area of research [116]–[119]. In addition to the various methods for visualizing trees, work has been done to combine network and tree visualizations with other dimensions of data [120]. These approaches share some similarities with Aardvark, but all differ in key ways.

First, Beham et al. incorporate images of generated geometries into a radial tree visualization [121]. However, the nodes of the tree lack time-series data, and while there are some non-temporal attributes, these are not included directly in the tree visualization.

Visualizations that combine trees with other attributes are more common in the literature. Nobre et al. visualizes genealogy data with related attributes by aligning nodes of the tree with rows in a table [122]. A similar approach is taken in Juniper for a more general-purpose tree plus attribute visualization [123]. Dendrograms linked with heatmaps [124]–[126] also associate trees with attributes. In contrast to the trees we consider, dendrograms augment the heatmap by indicating the similarity between rows of attribute data. Phylogenetic trees are also sometimes visualized alongside a heatmap [127]–[129]. Here, the tree displays the relationships of species to compare them with the measured attributes. However, for both dendrograms and phylogenetic trees, no time-series or imaging data is visualized.

There has been some work to combine temporal data with trees. There are several approaches for visualizing tree structures that change over time [119], [130]–[132]. However, we are interested in a single tree structure where attributes on each node vary over time. Icicle plots are a classic tree visualization technique that is often used for function call profiling [133], but the node's temporal relationship differs from ours. Shreck et al. organize time-series charts in a space-efficient TreeMap layout [134]. This layout may be appropriate for hierarchically clustering similar time-series data. However, our work requires a more direct encoding of parent-child relationships. There are a few techniques that are more similar to our own. Burch et al. align a tree with time-series data [135],[136] Similarly, Nobre et al. also show time-series data in a tree layout [137]. Although both of these resemble our approach, neither incorporates imaging data.

4.2.2 Image Snippets

We use the term *image snippet* or *snippet* to refer to a region of an image or video that is cut or *snipped* from its source and displayed in other contexts. Snippets can be useful for extracting the most interesting part of an image or video, especially if the original image is large or complex.

Small but important regions can be identified and extracted from high-resolution images. Lekschas et al. apply this principle to high-resolution images [138] and genome interaction matrices [97]. Ghani et al. do the same with network visualizations [139]. The image datasets we work with are not high-resolution, however, extracting individual cells into snippets enables combining them with other data types.

For complex multichannel images, there may be too many channels of information to display simultaneously, requiring an interactive approach. For Scope2Screen, Jessup et al. use the concept of a scope to interactively display a snippet at a higher resolution with different channels highlighted [65]. In contrast, our approach is focused on combining images with their derived metadata, not understanding many different image channels at once.

When interacting with data to either correct errors or modify designs, snippets can serve as a preview for possible user actions. Choi et al. show snippets of classification recommendations for microscopy images [68]. Similarly, Coffey et al. use preview snippets to show potential design alternatives for medical devices [140]. However, our approach is focused on the display of data currently present and does not incorporate modifications to the data.

When there are many images, exemplars can be used to represent an entire group. Lekschas et al. group similar images into *piles* and show an exemplar image on top to represent that pile [141]. In Loon, exemplar cells are selected for each experimental condition by sampling from metadata distributions [7]. In our work, we focus on designing a fundamental detailed visualization for a single lineage.

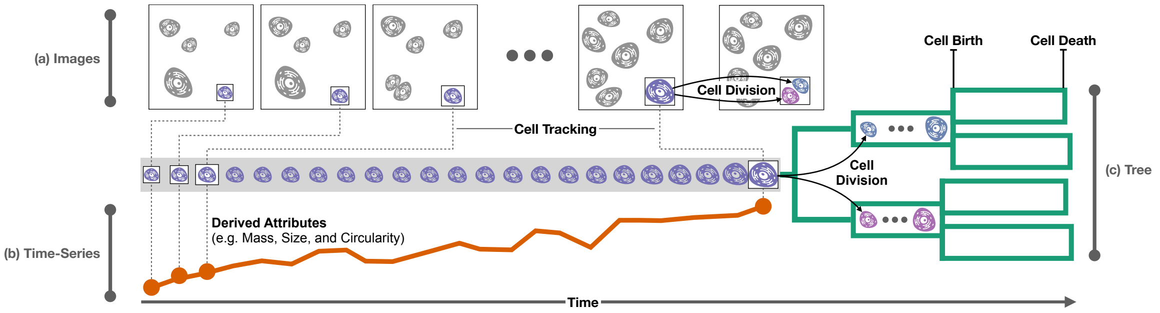


Figure 4.2: Illustration of the data acquisition pipeline. (a) Images of cells (violet) over time are the input data type for the pipeline. (b) Cell segmentation produces outlines of cells and various derived attributes (orange), such as the area, mass, or shape of the cell. (c) Lineages (family trees, green) of cells are constructed by observing cell divisions and matching the daughter cells with the parent cell.

For long videos, selecting individual frames can produce a static array of images that summarize the most important developments of the video. Lue et al. use this idea to create a visualization of news videos to summarize the important stories in a day [142]. Similarly, Yang et al. identify key frames in surveillance videos to quickly summarise the events in a video [143]. In our work, we identify key frames in a cell's development to summarize it.

4.3 Data

In this section, we describe the data Aardvark was designed to visualize. We provide essential details on how data is collected and processed in a time-series microscopy experiment (Figure 4.2).

Our collaborators take several steps to collect and process cell lineage data. First, **Images** of cells are taken repeatedly over the course of hours to days (Figure 4.2a). Next, cells are segmented to find their boundaries, resulting in *segmentations*. Cell segmentations are then tracked over time, resulting in *cell tracks* that ideally capture the lifetime of a cell. Attributes, such as mass, area, shape descriptors, fluorescence, and spatial positions are derived from these segmentations. These derived attributes over time and cell movement comprise the **Time-Series** data (Figure 4.2b). Finally, when a parent cell divides into two daughter cells, that division is identified, and the relationship between the cell tracks is recorded. This **Tree** of cell relationships includes the start and end time at each node. We refer to a tree as a *cell lineage* (Figure 4.2c).

We illustrate our work with three datasets that utilize cell lineages to inquire about fundamental scientific questions. Each dataset contains 2,000–100,000 cells, 30–4,500 cell tracks, and 4–156 lineages. The largest lineages for each dataset are tracked 5–13 generations and contain 9–219 cell tracks. The imaging data contains 200–3,000 imaging frames at a resolution between 160x160 and 800x800. The total size of the datasets range between 380 MB and 1.25 GB. We will describe the individual datasts in more detail in Section 4.8.

4.4 Domain Tasks

This work is the result of a four-year and ongoing collaboration with the Zangle Lab and the Judson-Torres Lab [7]. The first author attended group meetings and worked closely with graduate students and post-docs in these labs, formatting data and debugging code during that time. Furthermore, while designing and implementing Aardvark, the authors met on a recurring basis to discuss ideas and feedback on earlier versions of the tool. A static version of this visualization has already been included in work submitted for publication [144]. Throughout this collaboration, we identified the following domain goals.

4.4.1 Quality Control Tasks

Ensuring data quality is a difficult task that is critical to get right. We identified four sub-tasks, which are useful to discuss separately, yet are inherently intertwined, as issues with one aspect of the data tend to propagate into others.

- **QC-Segmentation**: Although cell segmentation techniques are improving, it is still an ongoing challenge. This is largely because cancer cells are highly heterogeneous — they come in many shapes and sizes, which makes a one-size-fits-all algorithm difficult to construct. Cells can also overlap, and debris can influence tracking. Therefore, domain experts need to review the segmentations of their data to ensure that the current segmentation algorithm is performing within expectations.

- **QC-Attributes**: Once cells are segmented, attributes about the cells can be derived. The types of attributes that can be measured depend on the particular instrument. Common attributes include mass, fluorescence, or a measure of roundness. These attributes are used, for instance, to see if different drugs affect cells' change in mass. Ensuring that the attributes are calculated correctly is essential for sound conclusions based on these analyses.

- **QC-Tracking**: To measure the change in a cell over time, a segmented cell has to be tracked across frames. This can be challenging as cells can move out of frame or divide. Common issues include when two different cells are recorded as one incorrect *merged track* or when tracking is temporarily lost for a cell, resulting in *broken tracks*.
- **QC-Lineage**: To study how cells proliferate, it is necessary to track cell lineages by identifying cell divisions and recording the correspondence between the parent and daughter cells. All the issues with segmentation and tracking can influence lineage tracing, but there are additional hurdles. For example, an algorithm can miss a cell division or record an incorrect division.

4.4.2 Discovery Tasks

Our collaborators want to study cell growth and propagation and the factors that influence it. In the following, we list representative domain tasks they are interested in.

- **DA-Propagation**: When cells divide, they do not always split their cell material evenly between the two daughter cells. When this happens, how does it affect later generations? For instance, if one cell receives a small amount of a particular protein, will later generations continue to be deficient in this protein, or will they bounce back and return to normal levels?
- **DA-Cell Cycle**: Cells go through “phases” in their lifetime, which is commonly referred to as the cell cycle. For example, shortly before cells divide, they get rounder and more compact. However, there are also cell attributes where the pattern before division is unknown and of interest.
- **DA-Changes**: Generally, cells grow and change gradually. Thus, deviation from this behavior, such as drastic changes in cell morphology, are of interest to our collaborators. For instance, a large drop in mass can indicate that a cell is dying.
- **DA-Comparison**: Identifying differences between cells is a common goal. For instance, analysts might ask how cancer cells respond to different drugs or how different branches of a cell lineage develop differently to the same drug. To answer this ques-

tion, domain scientists may want to identify divergences in any of the available data types across conditions, even between branches of a tree.

- **DA-Synchrony**: Healthy cells generally have roughly the same lifespan: two daughter cells born at the same time should undergo phases of the cell cycle synchronously. In contrast, unhealthy cells or cells exposed to drugs may exhibit more heterogeneity. Therefore, determining the synchrony of cells is of interest.

4.4.3 Communication Tasks

Our collaborators also need to disseminate their findings to their peers. Communicating findings based on complex data, as described in this chapter, can be equally challenging to quality control and discovery tasks. We specifically designed Aardvark with communication in mind and list specific communication tasks below.

- **COM-Explain**: Scientific work often involves complex experiments, ideas, and data. In particular, the data for cell microscopy includes multiple data types that each tell one part of the story. Distilling these data types into figures for use in papers and presentations is critical to conveying the main concepts.
- **COM-Trust**: Simply understanding the message that authors are communicating is not enough. Readers must also have enough information to make a judgment on whether they trust the findings. Here, trust means that the audience can see evidence that supports their belief that the findings in the work are correct. Here we focus only on trust in the data, not the myriad of other factors that can affect trust.

4.4.4 Task Abstraction

These domain goals can be mapped to abstract visualization tasks. The most fundamental task for any of these is the **synthesis** of the three data types. Relating data elements between the three types and visualizing them together is the primary visualization challenge we address in this chapter. In combination with this synthesis, there are three tasks: analyzing the **topological** structure of the cell lineage, viewing the **trends** of the cell attributes, and understanding the **spatial relationships** of cells. These abstract tasks can address the different domain goals. For instance, errors in the lineage data can be identified by examining the *topology* and cross-checking it with the cells *spatial relationships*. Similarly, understanding how cell attributes change over generations requires viewing *trends* in those attributes and investigating how those trends relate to the *topology*. Consequently, these visualization tasks are the primary drivers for our design principles (Section 4.5) and Aardvark designs (Section 4.6). We illustrate how these choices tie back to the lower-level tasks in the case studies (Section 4.8)

4.5 Design Principles for Visualizing Trees, Time-Series and Images

In this section, we describe design principles for integrating three types of cell microscopy data (trees, time-series, and images). Understanding the full picture of cell development requires analyzing all of these data types together. Consequently, our visualization designs must also represent the different data types together. In our design process, we adopt Javed and Elmqvist's design space for composite visualizations. According to them, composite visualizations "combine multiple visualizations in the same visual space" [145]. We use this design space and describe considerations on how to choose visual encodings. We then introduce three designs (tree-first, time-series-first, and image-first) that implement these design principles for our data and tasks.

As part of our design process, we **first identify a primary data type** and select a visual encoding suitable for that data type. This primary encoding serves as the *host* visualization. **Next, the secondary data types are embedded into this chart as *client* visualizations**, and **visual encodings are selected.** We use a combination of *superimposed views* and *nested views* (Figure 5.1a) for our client visualizations, depending on the affordances of the host view. Superimposed views place the client visualization within the same space as the host view. Nested views nest the client visualization within the marks of the host visualization. In addition, when it is necessary to minimize overplotting in the host visualization, we show detailed charts on interaction, which we consider *superimposed detail on demand views*..

The choice of primary data type for a visualization depends on the analysis task. Although all three types are required for most analyses, some questions still prioritize certain features of the data. For instance, an understanding of the cell lineage topology will benefit from an explicit rendering of a tree diagram showing cell division. Alternatively, comparing the growth curves of multiple cells benefits from showing them in a line chart.

Finally, spatial relationships of cells and their movements are best shown in a view that prioritizes the images of cells.

In practice, the choice of visual encoding and composition are not strictly sequential but **iterative and interleaved**. For example, when choosing a visual encoding for the host visualization, we ask the question of whether the encoding can accommodate desired composite views. The layout and appearance of a node-link tree visualization, for example, can be modified to make space for nested or superimposed composite views without occlusion. Similarly, the choice of a visual encoding may limit the options available for client visualizations. For example, if the primary data type and host visualization is a space-filling image, any composition will lead to overlap. In this case, small and unobtrusive client visualizations or on-demand superimposition are good choices. Finally, client visual encodings should be chosen to work within the constraints of the clients, which typically have reduced space. For example, it might be prudent to choose compact visualizations that are still useful if just a few pixels are available.

Since our collaborators are interested in all of these questions and more, we designed three composite visualizations, each encoding a different primary data type. Furthermore, each of these visualizations can be *juxtaposed* and linked to form a visualization system. The rest of this section discusses the high-level design decisions for each of these composite views, while the next section gives details for their instantiation in Aardvark.

4.5.1 Tree-First Visualization Design

The tree-first visualization (see Figure 4.3) uses the tree data type as the primary encoding. The nodes of the tree not only represent topology but also the lifetime of cells. Time-series charts are then nested inside the nodes of the tree diagram. Finally, images are superimposed along the time-series chart as space allows, and user-selected images are shown on demand. The composition of this view is illustrated in Figure 4.1b.

We use an explicit node-link representation for our *tree data*. The horizontal alignment and width of the nodes are determined by the start and end times of the cell tracks. Assuming correct data, these times correspond to the birth and division or death of the cell. The height of the node, as well as the spacing between nodes, can be adjusted based on analysis needs.

4.5.1.1 Nesting Time-Series

Within the node, we nest charts visualizing time-series data. As nodes are positioned and sized according to their birth and death time, the embedding and comparability of time-series data is straightforward. While there are various choices available to encode the time-series data, such as line charts or color maps, we use horizon charts [146] to encode the data. Horizon charts scale to vertically compact spaces, which is important for visualizing large trees and showing multiple time-series simultaneously (Figure 4.10). The redundant encoding of discrete bands using color further increases scalability for cases when the chart is only a few pixels high, at which point color is the dominant visual encoding. A second advantage of using redundant color in the tree is that color-coding along tree nodes is often familiar to domain scientists, unlike horizon charts.

4.5.1.2 Superimposed Images

To provide context for the time-series and topology data, we include images of individual cells superimposed along the horizon charts. The cell images are clipped from the *image data* based on segmentation information. Since the images are a secondary data type in this view, their size and position are constrained by the host visualization. Thus, it is generally impossible to show every cell, and a selection strategy to choose exemplars must be employed [7]. We considered various sampling methods, including sampling frames evenly throughout the cell's lifetime. This approach is useful if cells change gradually

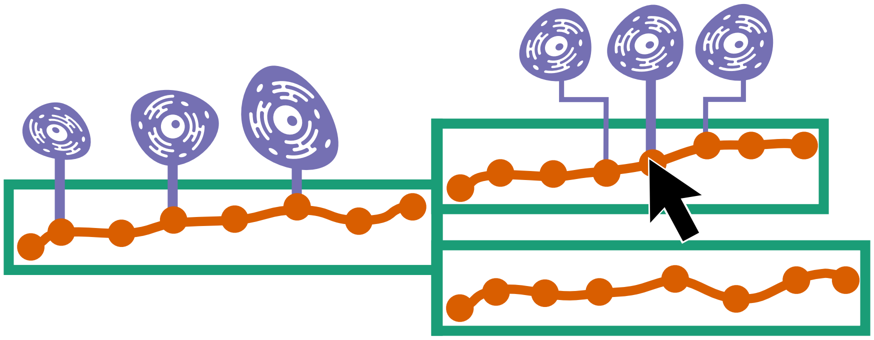


Figure 4.3: Schematic of the tree-first visualization design. The primary data is the lineage, i.e., the tree capturing the relationship between the parent and the daughter cells (green). The horizontal sizes of the nodes are scaled to correspond to the cells' lifetime. A time-series dataset (orange) is nested, and cell images (violet) are superimposed, either using an automatic selection algorithm (left) or on demand (right).

throughout their life. However, regular sampling may miss a critical development that occurs in the span of a few frames. Therefore, we developed an alternative approach to select exemplars based on importance metrics computed from the other two data types. Different metrics are conceivable, but we employ two for our use case. The first is based on highlighting topologically relevant frames: we include frames right before or after cell division (at the beginning and end of the node). The second metric is based on changes of features of cell attributes, so that cells undergoing rapid change are included, or errors can be identified as seen in Figure 4.4.

4.5.1.3 Superimposed Images on Demand

While a data-driven selection strategy for exemplars provides a good starting point for showing relevant cells, some scenarios benefit from showing additional cells on the fly. Hence, we provide two methods to show superimposed cells on demand. First, free-form selection along the timeline enables an analyst to show any frame for a cell. Second, we provide keyframes on demand before and after selected cells. The latter is useful to observe the different stages of the cell cycle around an important point, such as a division. If the main frame shows a cell in the process of division, the prior frame will show the cell just

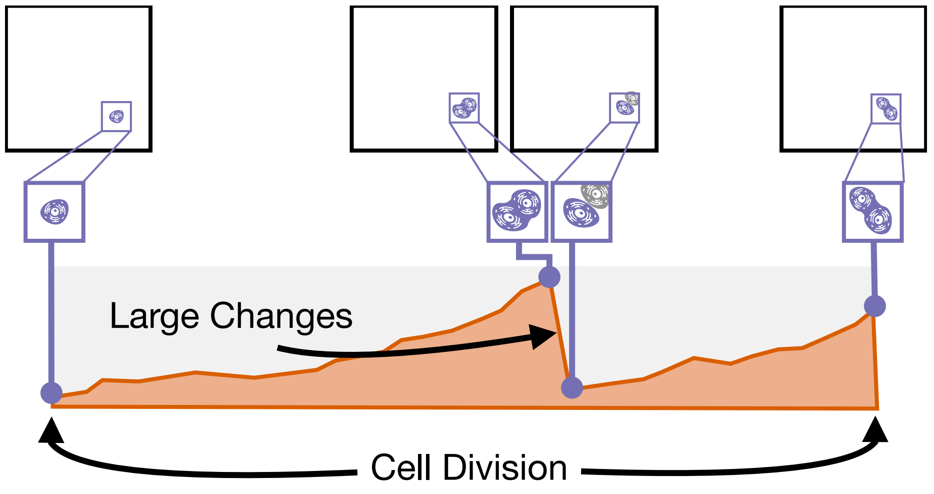


Figure 4.4: Illustration of cell snippet selection. Cells snippets are extracted and shown at the beginning and the end (far left and far right) of the life of a cell. Additional snippets are shown when a large change in an attribute is detected. In this example, the attribute experiences a sudden drop. The associated cell image indicates that the reason is a cell division that was missed by the algorithm.

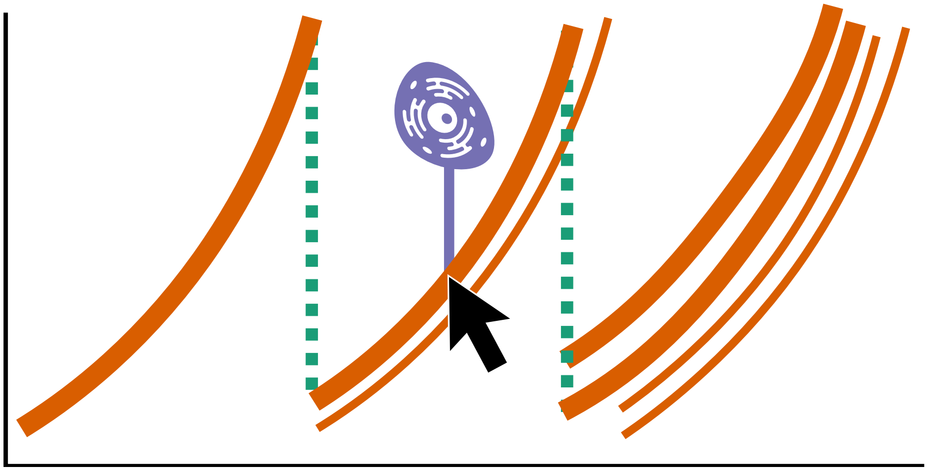


Figure 4.5: Time-series-first visualization design showing attributes of individual cells as line charts (orange). As cells divide, the original cells' line ends, and new lines representing the daughter cells begin. Topological information about lineages is shown using superimposed on-demand composition for selected cells (green): the selected cell is connected to its parent with a dashed line. Lines corresponding to ancestors and descendants of a selected cell are also shown in bold. Cell images are shown using superimposed on-demand composition by rendering a cell image at a selected time-point (violet).

before it divides, and the subsequent frame will show the newly divided cells.

4.5.2 Time-Series-First Visualization Design

The time-series-first visualization prioritizes the derived cell attribute data. Attributes of interest commonly vary between studies (e.g., mass, fluorescence, shape), yet the analysis tasks for these time-series are similar. The most fundamental task is understanding the change of an attribute for a single cell over time. For instance, an analyst may ask, *is a cell growing or shrinking?* Next, they may want to understand the behavior of a group of cells. *Are all cells growing, shrinking, or is there diverging behavior?* Line charts are well suited for this detailed comparison of time-series data and thus serve as our host visualization, as illustrated in Figure 4.5. The questions about cell attributes usually extend across generations. For example, analysts might ask whether a parent's characteristics are predictive of the daughter cells' attributes, e.g., *For cells that grow quickly, do their progeny also grow quickly?*

4.5.2.1 Superimposed Tree on Demand

A node in the line chart is represented by a line, and hence, similar to the image-first view the node positions of the tree are determined by the attributes. Therefore, plotting connections between nodes results in significant visual clutter. To address this issue, we do not attempt to show full topological information in the time-series-first View but instead show ancestry and descendants through superimposition on demand by connecting the end of one cell's line with the start of their daughter cell's line in the chart (see Figure 4.5 in green). To emphasize this relationship, we also display ancestors and descendants of a selected cell in bold.

4.5.2.2 Superimposed Images on Demand

Since line positions are driven by the data it represents, they can not be arbitrarily repositioned. This leads to challenges embedding images within the chart without obscuring the primary data type. Therefore, we chose to show cell images only when users select a specific feature in the data and then superimpose the image on demand in the form of a tool-tip visualization (see Figure 4.5 in purple).

4.5.3 Image-First Visualization Design

The image-first visualization, illustrated in Figure 4.6, uses the image data as the primary encoding. Images are especially useful for reviewing the spatial relationships of multiple cells and how cells move through space. For experiments that track cell lineages, combining all three data types is necessary to understand the relationship between cell movement and cell divisions. As an image view is a space-filling visualization, adding composite views will lead to occlusions. Hence, we choose to superimpose attributes that fit into the same coordinate space over embedding nested visualization thumbnails.

4.5.3.1 Superimposed Time-Series

Cell movement can be reduced to a trajectory of the cells' center of mass, resulting in a time-series of locations. Since this time-series data shares the same spatial coordinates as the image data, the trajectories can be superimposed on the image. This superimposition hence follows the "eyes over memory" guideline of visualization design [147] of explicitly

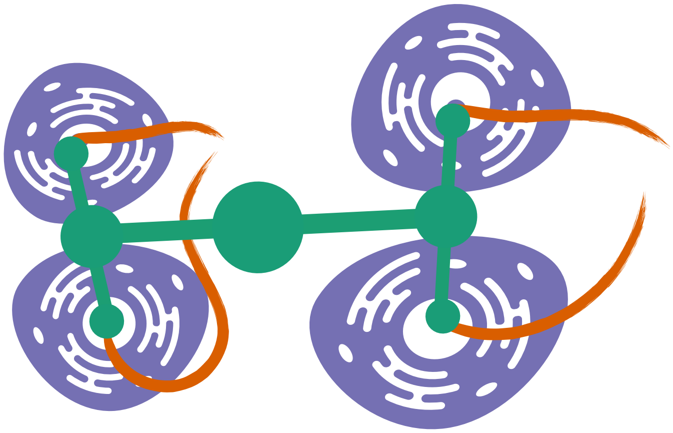


Figure 4.6: Image-first visualization showing a full image with multiple cells (violet). Time-series data, in the form of cell location over time (orange), is superimposed, enabling analysts to understand movement over time in a static image. Lineage trees are also superimposed (green), showing relationships between cells.

showing a temporal relationship in a static image, over-relying on a temporal animation, where viewers would have to memorize prior locations.

4.5.3.2 Superimposed Tree

Cells in an image commonly are closely related, and understanding that relationship is important for many analysis questions. To show these relationships directly in the image data, we superimpose the lineage tree on the images (see green marks in Figure 4.6). We use a node-link representation where the node positions are determined by the cells. Note that only the "leaves" of the tree can exist in an image; the parent cells have divided and do not exist anymore and are instead represented as circular marks. We also considered alternative layouts, such as implicit tree layouts [117], but found that the node-link representation best shows the relevant topology. Note that the tree uses a partially fixed layout, since the positions of the leaves are given by the cell location (parent cells can be freely placed). This restriction will make the topology less evident in most cases, illustrating the tradeoffs present when embedding these secondary visualizations in their client view.

4.5.4 Juxtaposition

Each of these composite visualizations has its strengths and weaknesses. Juxtaposing all three views together creates a system that can tackle a wide range of analyses. Linked and brushing ties the data elements together across these views. Since each view shows every data type, there are many opportunities to link data elements together. For instance, if a secondary data type is superimposed on demand in multiple views, triggering one view to show the superimposed chart should trigger all of them. In our designs, images are superimposed on demand in the tree-first view and the time-series-first view. Thus when a snippet is shown in the tree view, the same snippet is shown in the time-series view. However, the same principle for linking applies even if the data types, visualization encodings, or composition techniques differ.

4.6 Aardvark Design

In the previous section, we described our design process and the high-level design decisions of choosing visual encodings and choosing view compositions to integrate the different data types. These designs are intended to be transferable to and instructive for other similar scenarios. In this section, we provide details on how we instantiated these designs in our prototype, Aardvark, and provide sufficient details to ensure the reproducibility of our implementation. Figure 4.7 shows examples of how these designs are implemented in the tool.

4.6.1 Tree-First Visualization Details

4.6.1.1 Tree Layout

There are several user-configurable options that determine the tree layout, allowing analysts to balance priorities between data types. A generously spaced layout will ensure that there is enough room above each horizon chart to display the images for that cell. Alternatively, a dense layout will produce a compact view of the tree structure but will display fewer image snippets. Similarly, adjusting the height of horizon charts is a tradeoff between a larger tree with more detailed time-series data and a more compact tree showing less detail. The choice between a detailed and dense view is not binary but rather a spectrum that balances priorities between the three data types.

4.6.1.2 Cell Snippet Extraction

Cells are extracted from the source image by centering the cell and copying a constant number of pixels around the center. The number of pixels is the same for all cells so that

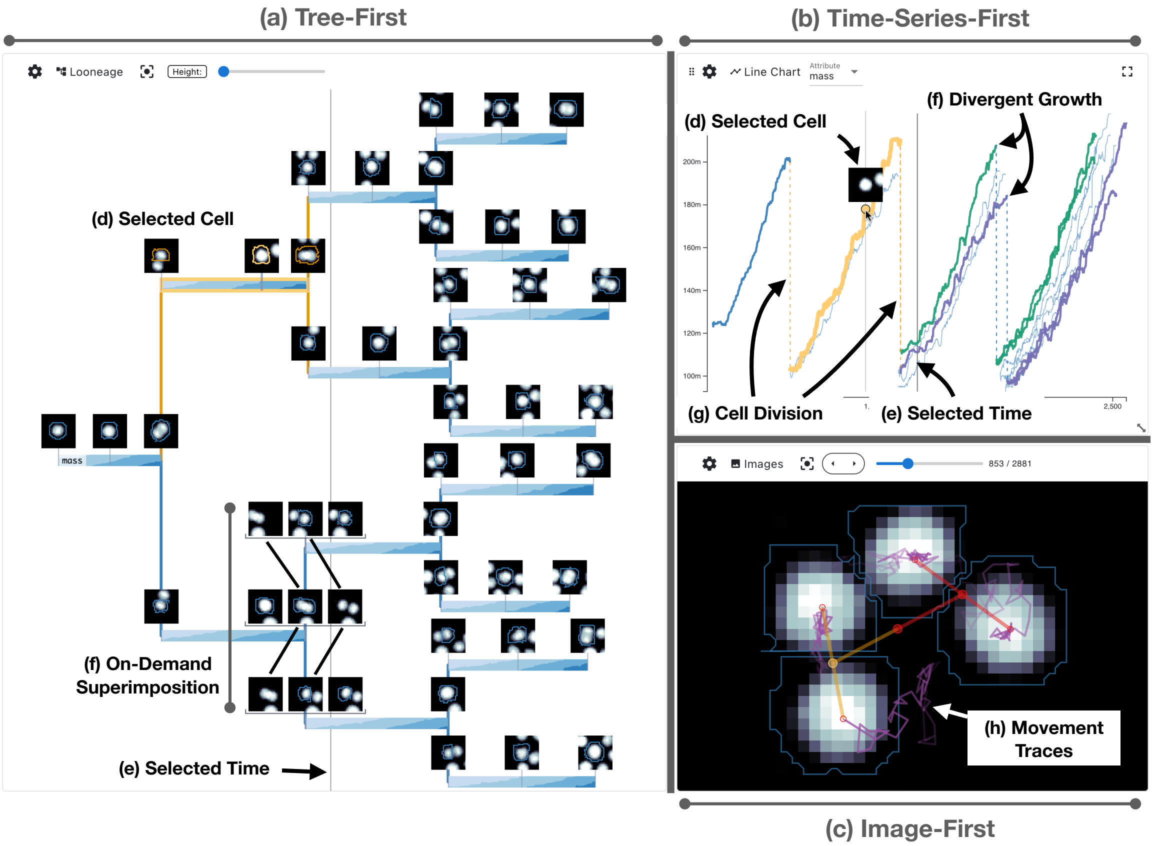


Figure 4.7: The three visualizations as implemented in Aardvark. (a) The tree-first view shows cell growth and cell image snippets. The node at level two at the top is highlighted in orange. (b) The image-first view shows the four leaf cells in the tree. The exact frame is highlighted by a vertical line in both the tree and time-series view. The lineage and the spatial movement are also shown. (c) The time-series-first view shows the highlighted cell in orange. The daughters of the selected cell show different growth behavior as evident from both the line chart and the horizon charts.

the display size and scaling are consistent across snippets. One side-effect of this approach is that the cell boundary may not fit completely inside the region that is extracted, which frequently happens if cells have elongated shapes. In this case, we indicate where the cell is cut off with red lines (Figure 4.8). These snippets are then placed above or below the horizon chart, depending on available space.

4.6.1.3 Interaction

Aardvark supports superimposed charts on demand in a few ways. Hovering on the horizon chart will show the snippet at that time point, and the cell boundary clipping is disabled, so the full segmentation outline can be seen. Selecting a time point will pin the

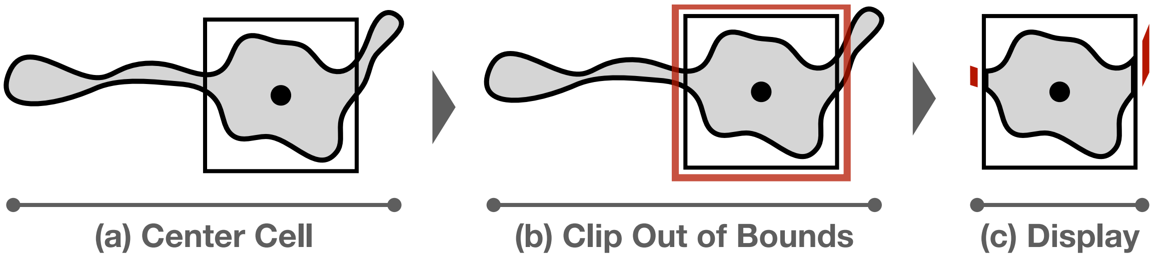


Figure 4.8: Clipping of cells with large peripheral features. Some types of cells have features that extend far beyond the cell core. Rendering the whole cell within a small embedded view would result in barely visible features. To address this, we (a) first determine the center of the cell, (b) then clip the features outside of the bounds of the core, and (c) display the clipped cell with indicators that features have been clipped (red).

snippet in place. Hovering on existing snippets will show the previous and next snippets, even beyond the cell track (Figure 4.7f).

4.6.2 Time-Series-First Visualization Details

The line chart view within Aardvark supports different modes of aggregation—from visualizing individual cell attributes to population-wide aggregation. In the time-series-first visualization (Figure 4.7b), time is always mapped to the x-axis, and attributes are mapped to the y-axis. Selecting a line will show a snippet of that cell above the line (Figure 4.7d). If a line is selected, lineage information is displayed for cells with direct relationships to the selected cell through explicit rendering with dashed lines or emphasis and color highlighting, as described in Section 4.5.2. To help distinguish between different branches of progeny, the two subtrees of the selected cell are assigned a different color.

4.6.3 Image-First Visualization Details

Aardvark supports four different imaging layers. The base microscopy **images** record pixel intensities that can have different meanings (mass, fluorencense, etc). A colormap is applied with an adjustable range so the signal in the data can be highlighted while reducing noise. **Cell boundaries** are displayed with an outline. **Cell trajectories** are shown as a line that fades into the past, as shown in Figure 4.7h. **Cell lineages** are displayed with a node-link diagram where internal nodes represent cell ancestors. Figure 4.9 illustrates this process across four generations of a single lineage.

4.6.4 Across View Interaction

There are many interactions across views in Aardvark, most importantly based on selecting a cell. Selecting a cell also selects a cell track, selects a time or image frame, and selects a cell lineage. In Aardvark, each view has special logic for how to display a selected cell, track, lineage, and time. All of these selections are highlighted. In the *tree-first visualization*, the selected cell snippet is displayed (Figure 4.7d). All cell snippets in the selected track have their cell boundary highlighted. The horizon chart of the selected track and its connecting lines are highlighted. Finally, a vertical line indicates the selected time (Figure 4.7e). In the *image-first visualization*, the cell boundary is highlighted if the selected cell is in the current frame. Otherwise, if the selected cell's progeny are in the current frame, the node and connecting edges in the tree representation are highlighted (Figure 4.7c). In the *time-series-first visualization*, the selected cell is shown as a snippet (Figure 4.7d). The line corresponding to the selected cell track is highlighted, as are the edges directly connected to the selected cell. Finally, the selected time is indicated with a vertical line (Figure 4.7e).

4.6.5 Other Views

Aardvark includes other useful views. **Tables** provide access to raw values for cells, tracks, and lineages. Sorting data columns provides a means to select lineages with specific characteristics. Displaying **dataset metadata** provides basic context and sanity checking for a dataset. Finally, the **visualization interaction state** is tracked and displayed for state recovery and provenance [148].

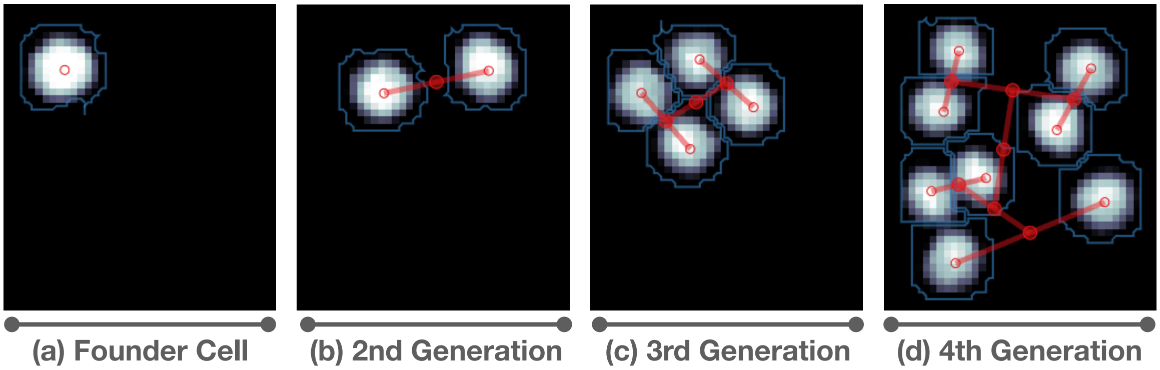


Figure 4.9: Image views illustrating cell division across four generations and the overlaid cell lineages.

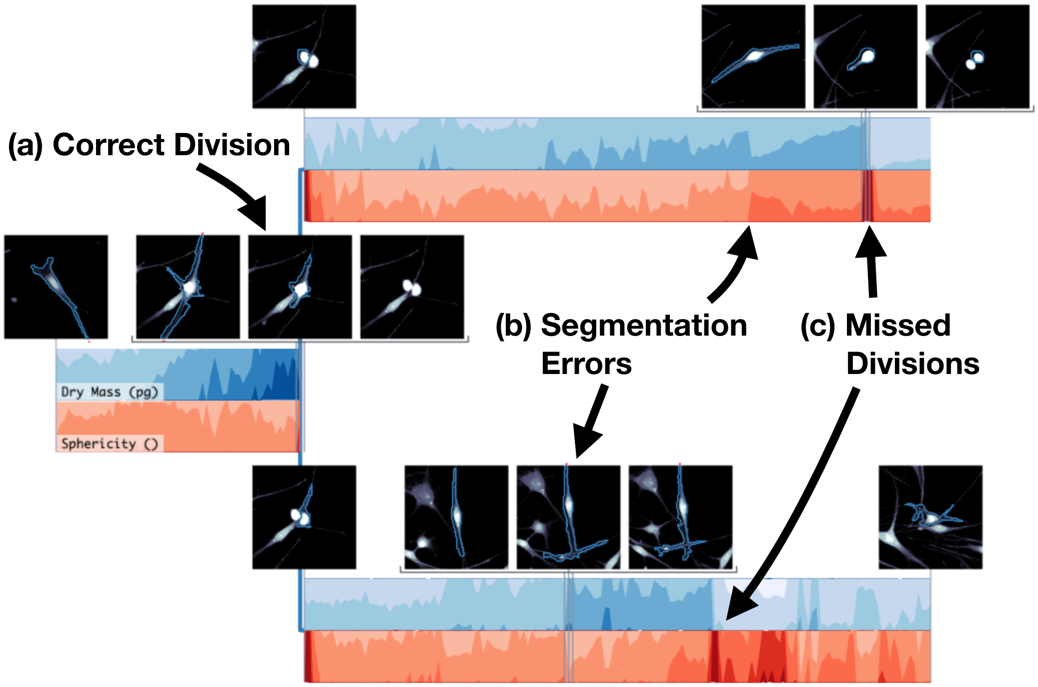


Figure 4.10: Shows on example of (a) correct division, (b) multiple segmentation errors resulting in rapid changes in the attributes, and (c) two missed divisions. Immediately before the missed division, mass and sphericity increase, but then, one of the children is incorrectly tracked as its parent.

4.7 Implementation

Aardvark is implemented as an open-source front-end application available with demo datasets at `https://aardvark.sci.utah.edu/`. Imaging and metadata are fetched from a file server that can be specified to support various setups. For example, the demo datasets associated with this chapter are stored on an AWS S3 bucket, but the tool can also be configured to access files stored locally.

Aardvark uses various libraries and web technologies: TypeScript, Vue 3, Pinia, and Quasar comprise the base framework and UI library; deck.gl serves as the base WebGL framework for the image and lineage charts. Components from Viv are used to load and render standard microscopy image formats as layers in deck.gl [149]. These are combined with custom deck.gl layers developed for this project.

We use utility functions from the D3 library [108] and the D3 Flextree plugin, which extends the tree layout module of D3 to produce a compact layout with variable width nodes [150]. Finally, the Trrack library is used to record and display interaction provenance

[148]. For a complete list of libraries and the full source code, refer to `https://github.com/visdesignlab/aardvark`.

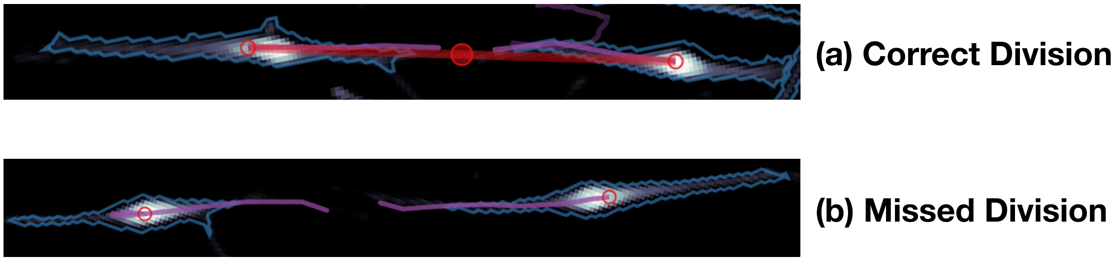
4.8 Case Studies

Aardvark was designed with our long-term collaborators (who are also co-authors), who have real data and real scientific questions [104] related to cancer cell development. Since our collaborators are the only researchers we have access to who can evaluate our tool, we do not record subjective numerical evaluations of the tool due to the influence of demand characteristics [109]. Instead, we provide rich descriptions for how Aardvark was used to ensure rigor [151]. The following case studies are selected examples collected over the course of the collaboration intended to illustrate the utility of our design with real data.

4.8.1 Quality Control: Cancer Triggering Microenvironments

This dataset is part of a study that examines the difference between the development of benign moles and melanoma [152]. Understanding the steps that initiate malignant disease could reveal potential chemopreventative strategies for skin cancer. The study exposed healthy human skin cells to a chemical that simulates the environment skin cells experience when exposed to the sun (UV radiation).

As a first step, we browse lineages that exhibit a large drop in mass and show it in the tree-first view. Since our image selection prioritizes data points with large changes (**DA-Changes**), image snippets at points of change are automatically shown in the tree-first view, which quickly reveals why the drop in mass occurs. Some instances occur because the cell moves out of the imaging view (**QC-Tracking**). Some occur because part or all of a nearby cell is incorrectly included in the cell (**QC-Segmentation**). Some occur because a cell divides, but tracking is incorrectly connected to one of the daughter cells (**QC-Lineage**,**QC-Tracking**). Understanding which of these errors occurs requires a combination of all the different data types. The tree-first view is well suited for quickly making these identifications. In Figure 4.10, several errors can be seen in one lineage. This figure shows two cell attributes in the horizon charts (mass and sphericity), which are informative for quality control because their behavior within the cell cycle is well un-



(a) Correct Division

(b) Missed Division

Figure 4.11: Reviewing tracking of cell divisions in the image view. (a) A correctly tracked division can be identified by the visible tree indicating that the cells are siblings and the location traces showing that they originated at the same place spatially. (b) The lack of a tree indicates that no division was recorded, yet the location tracks show that they did indeed originate at the same place.

derstood (**QC-Attributes**). The first cell division recorded in this example appears correct (Figure 4.10a). The image segmentations are in alignment (**QC-Segmentation**), and the cell attributes show the expected response — cell mass increases throughout the cell's life, and cells are spherical just before and after division (**DA-Cell Cycle**). In the second generation, there are similar patterns in the attributes, but closer inspection reveals different reasons and types of errors. Figure 4.10b shows a sudden change in mass. Inspection of the cell images reveals that this is a segmentation artifact from a cell "limb" being excluded and included in different frames. Figure 4.10c highlights two missed divisions. In both cases, mass and sphericity increase and then suddenly drop. Inspecting three frames at this point reveals that the cell divided, and one of the daughter cells is tracked incorrectly as its parent cell.

QC issues can also be identified in the image view. Segmentation errors can be spotted by matching the segmentation outline to the image. Divisions can be validated using the superimposed tree and location data. Figure 4.11a shows a clipped example of a correctly identified division. The two cells are connected, indicating that they share a parent, and the location traces show that they both came from the same origin. Figure 4.11b shows an example with similar image and location trace data, but the tree connection is missing. Navigating to an earlier frame reveals that these cells are from a common parent.

These examples illustrate the variety of issues that can be quickly identified in Aardvark as a first step toward addressing them.

4.8.2 Data Discovery: Automated Cell Lineage Tracking

Producing cell lineage data is challenging. Automated systems can be error-prone, and manual reconstruction can be time-intensive. This dataset is from work that demonstrates a technique for producing high-quality lineages automatically [153]. In short, this technique separates individual cells into small microwells where they can grow and divide in isolation. These microwells ensure that cells remain within the microscope's view and that all cells within a microwell originate from a single founder cell. The data shown in this chapter is focused on a single microwell containing a mouse leukemia cell lineage.

Before interpreting the results of the data collection and processing, a critical first step is to verify that the lineage data is correct. For this dataset, it is expected that the lineage tracking will eventually fail once the number of cells in the microwell leads to cells overlapping in 3D which makes them impossible to reliably separate. A question our collaborators are asking is hence: how many generations can be accurately tracked? We can approach this question in Aardvark by iteratively exploring the dataset for an individual lineage. The tree-first view provides an initial overview of the data (Figure 4.7a). The default view shows that every cell starts at roughly the same mass and has a similar growth rate, which is expected for this dataset (**QC-Attributes**). Image snippets of the cells are automatically shown for each division point. Selecting a snippet just before cell division reveals the next frame (Figure 4.7f), which makes it easy to verify that the three cells (parent and two daughter cells) are recorded correctly (**QC-Lineage**). Interactively expanding the width of the tree allows more space for additional snippets to be shown. These can be quickly scanned to verify that the segmentation and tracking of the cell is consistent (**QC-Segmentation**,**QC-Tracking**).

After verifying the quality of our dataset, our collaborators start investigating whether there are biologically interesting patterns. They select a cell in the second generation (Figure 4.7d), which updates the time-series-first view to show part of the lineage tree (Figure 4.7b). The distinct coloring of the two different branches reveals a difference between them (Figure 4.7f, **DA-Propagation**). Our collaborators conclude that data such as these could be used to assess drivers of asymmetric division and heredity by, for example, using genetic mutants of key growth regulation pathways.

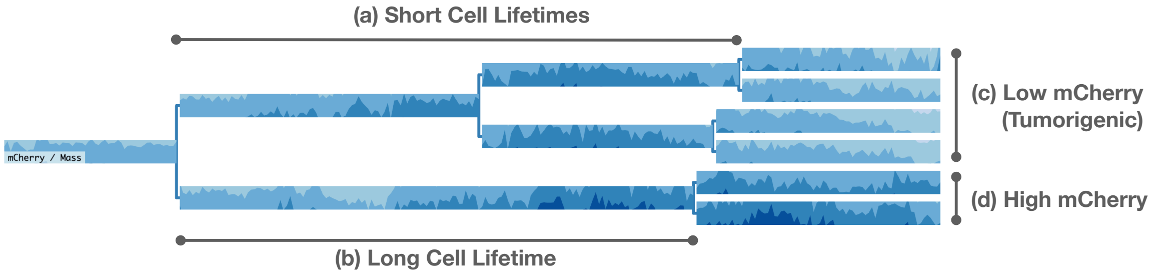


Figure 4.12: Example of the emergence of tumorigenic melanoma cells in a single lineage. Notice the distinct asymmetry of the tree: cells in the top branch (a) live about half the time before they subdivide compared to the cells in the bottom branch (b) — cancerous cells tend to grow faster than benign cells. The embedded horizon charts show mCherry, a fluorescent marker of tumorigenesis normalized by cell mass. Low mCherry indicates a tumorigenic cell. We observe a distinct differentiation between the cells in (c) the top branch (low mCherry) and (d) the bottom branch (high mCherry).

4.8.3 Communication: Tumorigenic Cell States

Our collaborators study how skin cancer cells (melanoma) leave the primary tumor and form new tumors in other organs (metastatic dissemination) [144]. Cells that can form new tumors are referred to as tumorigenic cells. However, cells are not predestined to be tumorigenic from their genetic material. Instead, the tumorigenic state of a melanoma cell can change within a cell's lifetime or across generations. This state can be measured by engineering the cells to express a fluorescent marker, mCherry, under the control of a specific promoter. Low levels of mCherry indicate that the cell is in a tumorigenic state. In this dataset, a combination of automated and manual processing was used to construct several lineages. These lineages illustrate how this tumorigenic state can change across generations.

Figure 4.12 shows an example of this effect: A distinct asymmetry in the tree topology is apparent in the tree-first view (compare the long lifetime of the cell in Figure 4.12b with the short lifetime of its sister cell and its descendants (Figure 4.12a). The cells in the branch at the top are subdividing in about half the time compared to those in the bottom branch (**DA-Synchrony**, **DA-Comparison**). Within the tree-first view, the normalized abundance of the fluorescent marker is shown, *mCherry / Mass*. This marker shows the relative amount of mCherry in the cell independent of its total mass, the latter of which varies throughout the cell lifetime (**DA-Cell Cycle**). Since the horizon charts are nested within the tree structure, it can be seen that the later generations of the top branch end the experiment with a lower amount of *mCherry / mass* (Figure 4.12c). This short division time and low mCherry values indicate that this top branch of cells have transitioned into a tumorigenic state. This illustrates how within a single lineage, some, but not all, cells transition into a tumorigenic state. A similar figure from an earlier version of Aardvark is included in a preprint describing the study [144] (**COM-Explain**, **COM-Trust**), illustrating the value of Aardvark for scientific communication.

4.9 Discussion and Limitations

In this work, we have focused on visualizing data that combines trees, time-series, and images, where all the data is derived from the primary data type: time-series microscopy images. A major theme throughout this work is that understanding this data wholistically requires understanding how these three pieces of information fit together.

4.9.1 Scalability

Our work has limitations, particularly when scaling to large trees with over a few hundred nodes. While it is technically difficult to construct trees of this size for the domain of cell lineage tracing, it is conceivable for similar datasets from other domains. In such cases, displaying the entire tree with all cell images may not be feasible. To mitigate this, we enable interaction to adjust tree size and spacing. We also show only one tree simultaneously. Showing multiple trees would be useful for comparing or exploring a collection of lineages. We leave dedicated approaches for multiple lineage comparisons to future work.

4.9.2 Generalizability

While our work is directly relevant to those interested in cell microscopy data, other domains with similar data combinations could benefit from elements of our approach. Satellite imaging of Earth can have derived attributes computed on the images or be associated with data collected on Earth in similar regions. A subdivision of regions could produce a tree relationship. Astronomy imaging tracks and analyzes the change of celestial bodies. Analyzing data from smart cars could combine imaging and sensor data. Finally, physics simulations that track attributes, positions, and shapes of objects under different conditions could also use elements of our designs.

More broadly, although this combination of data types is specific, the trend of combining data types is not. In particular, deriving secondary data from some imaging modality is a widely employed approach across disciplines. In biology, imaging is often combined with other approaches to reveal the complex mechanisms that comprise organisms. Our work would be a useful reference for problems that have a similar combination of data types. Yet, we argue that our design principles could still be applied to other data type combinations. For example, the process of selecting primary and secondary data types, their encodings, and their composition could be leveraged for any combination of data types. This framework for reasoning about the design options helps navigate the complex space of these multivariate datasets.

4.9.3 Evaluation

We considered several strategies for evaluating Aardvark. Since it is a specialized tool with a limited number of expert users, a quantitative evaluation is difficult. A quantitative study with a broader audience could potentially be performed for certain isolated components of our system. However, evaluating the full system in this way lacks ecological validity. Alternatively, we considered evaluating the tool with our current users. Since they have codesigned the tool with us and are authors of this paper, a study is susceptible to demand characteristics [109], i.e., introducing bias to give positive scores or feedback. Therefore, we decided that, in order to ensure the rigor [151] of our design study, we report factually on different scenarios where the tool has provided utility for our collaborators.

4.10 Conclusion

In conclusion, this work examines three distinct data types (trees, time-series, and images) that are interwoven to create a complex multivariate dataset. We describe our design principles for combining these complex, disparate data types into intuitive composite visualizations. We use these principles to implement an open-source visualization tool, Aardvark. We demonstrate the utility of Aardvark to perform quality control, data analysis, and communication tasks with three case studies.

In particular, for quality control, the synthesis of these three data types provides a more complete picture for understanding the data (*Trees-and-Foreset*). Additionally, since some data types, like images, cannot be easily compressed visually, it is necessary to sample them for compact views. In this work, for the tree-first view, image samples are selected automatically, but any image can still be selected by users for review (*Flexible-Structure*).

CHAPTER 5

FERRET: REVIEWING TABULAR DATASETS FOR MANIPULATION

This chapter is based on our previous published work [9] that investigates the manipulation of tabular scientific datasets. Reviewing data for intentional manipulation is a less common practice than reviewing for unintentional errors, thus it may not be obvious what to look for. In this work we collect common patterns found in presumably intentionally manipulated datasets so that the visualization tool we developed, Ferret, can guide users through an investigation.

5.1 Introduction

Data manipulation is an unfortunate reality of the scientific publication process. Like plagiarism, it is an unethical attempt to game the system, usually to further academic careers. The effects of falsified data in research vary. Manipulated data and the resulting incorrect claims can mislead scientists who want to build on the incorrect knowledge or lead to actions not based on evidence. Manipulated data can even lay a faulty foundation for a whole area of research, leading to years of wasted effort by researchers. At worst, incorrect and dishonest findings can result in the inappropriate application of knowledge in society, with potentially severe consequences, such as the harmful treatment of patients. In a far-ranging Alzheimer's scandal [154], image and numerical data were suspected to have been manipulated in what was considered one of the most important publications on the topic. Based on this — now considered false — knowledge, drugs were developed and even FDA-approved, exposing patients to potentially useless medication while foregoing alternative treatments and causing side effects. Pharmaceuticals have also invested "millions of dollars, or even billions" [154] based on the manipulated findings. Unlike plagiarism — which can be discovered by checking articles against other published sources — falsified data is difficult to detect. Plagiarism checks are now part of the editorial process

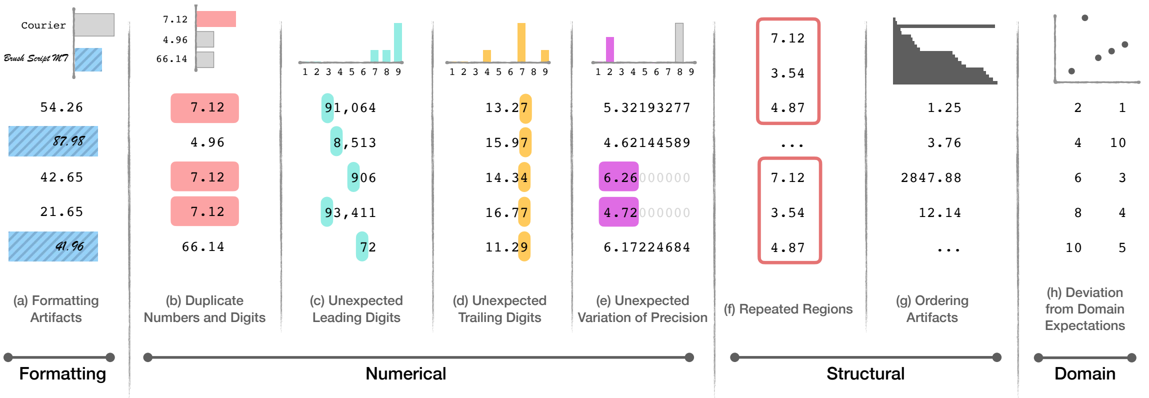


Figure 5.1: Artifacts of manipulation we discovered when analyzing manipulated datasets. The artifacts range from (a) unexpected formatting; to numerical issues such as (b) duplicates, unusual distributions of (c) leading or (d) trailing digits, and (e) variations in precision; to structural issues such as (f) repeated regions and (g) artifacts associated with sorting and ordering items; and to (h) the unrealistic relationships in the data.

of many conferences and journals. However, in several high-profile data manipulation cases, scientists have had seemingly productive careers, and only after a single case of misconduct surfaced did the community critically scrutinize their whole academic record to find many instances of wrongdoing [155].

Besides urging individuals to refrain from such activity, how can we prevent or at least mitigate this problem? To address this issue, we look to peer review, a cornerstone of the scientific process. Although peer review has known flaws, the premise of peer review is that experts can verify the soundness of the research and increase the quality of published works. So why is fabricated data not caught in this step of the publishing pipeline? There are many factors: reviewers may assume a good-faith effort by their peers and are not looking for falsified data. In addition, combing through data to find signs of malpractice is difficult and time-consuming, especially when reviewers are not educated on what to look for and have no tools that can help at their disposal. Also, checking data requires that the data is made available to the reviewers and, subsequently, the readers, a practice gaining momentum with the open science movement but still far from universally adopted [156].

Existing tools that help find cases of data fabrication tend to focus on finding duplicated regions in images. The goal of our work is to equip editors, reviewers, and scientists with the knowledge and tools to make the investigation of fabricated tabular data feasible. In particular, the tools we provide are designed to aid and enhance human judgment, as scientific data can be noisy and extremely varied, and alternatives leveraging automated statistical analysis can potentially encourage false accusations.

Our work has two primary contributions: first, we identify common artifacts of data manipulation using a combination of analyzing datasets known to be manipulated and interviews with researchers investigating fraudulent datasets. Second, we propose an array of design principles and visualization methods to saliently surface these artifacts, thereby enabling experts to easily and confidently identify fraudulent datasets.

As an additional contribution, we have developed Ferret, a prototype in which we have implemented these visualizations. In addition to these different visualization designs, we also include guidance on interpreting the results directly in the tool. Since artifacts can arise both from falsified and truthful data, it is important for users of the tool to have guidance on interpreting the results without being prescriptive in how they use the tool.

We evaluate our methods and our tool using case studies from a series of known fraudulent datasets, demonstrating that these patterns become evident by leveraging Ferret. Finally, we discuss the ethics and the potential for abuse of our approaches.

5.2 Methods

To further understand how datasets can be manipulated, we collected datasets with known issues predominantly associated with retracted publications. To identify fraudulent datasets, we leveraged a database collected by the Retraction Watch Project, a website that tracks retractions in their database and disseminates them through blog-style articles [157], through community feedback on social media, and through interviews with two researchers who have investigated and reported evidence of data falsification. In total, we identified 10 datasets, with strong evidence that some manipulation occurred on them, summarized in Table 5.1.

We obtained a complete version of the Retraction Watch Database [158] through a special request to the database curator. Since these papers were retracted for various reasons, we filtered to papers that included "Falsification/Fabrication of Data" as one of the reasons for retraction, resulting in 1161 candidate papers. Next, we manually examined the papers to find fraudulent datasets by reading the official reason for retraction and checking the retracted publication for any references to public data. After checking 103 papers, we found only a single tabular dataset with signs of manipulation. We hypothesize that this low success rate is due to a focus on manipulated images in the database and because authors who manipulate data are incentivized to not publish it.

We then elicited help through social media. Using this approach, we identified four datasets associated with retracted papers. All four datasets also have an associated blog post where the evidence for manipulation and process of investigation has been posted. Search for the paper titles in the Retraction Watch Database revealed that these papers are in the dataset but were not flagged with "Falsification/Fabrication of Data". Three of the four include a flag related to data, such as "Error in Data," "Unreliable Data," and "Concerns/Issues About Data." The fourth paper, which is a preprint, has only a flag of "Notice - Limited or No Information." We suspect that these less serious classes may have been used due to an abundance of caution by editors.

We also interviewed two researchers who have investigated and reported evidence of data falsification. These interviews provided us with two additional datasets. One of these was in the Retraction Watch Database, again without the "Falsification/Fabrication of Data" flag but with other flags related to data. The other paper was not in the Retraction Watch Database as of Oct 28, 2022. The interviews also introduced us to patterns of manipulations and approaches for data fabrication that these experts had encountered. For example, we had not considered checking the plausibility of the data in a larger, domain-specific context. The interviews also provided additional context for how analysts search for anomalies.

To find common patterns of artifacts across datasets, we performed a primary analysis of the data in Excel and Ferret. We also reviewed existing discussions of anomalies in the data in published works, blog posts, and online forums such as PubPeer.

5.3 Datasets Overview

As described in the previous section, we collected datasets that contain data manipulations associated with retracted papers. All datasets are listed in Table 5.1. Here we briefly introduce a subset of these datasets, and how they were likely manipulated so that it is easier to understand the artifacts present in the datasets.

- **DS-Driving**: This dataset comes from a retracted study on honesty in the field of psychology. One experiment asked participants to report the odometer mileage of their car both before and after some period of time. It appears that the "after" column was generated by adding a random number between 0 and $50,000$ to the "before" number. In addition, half of the rows also appear to be generated by adding a small amount of noise to the original values.
- **DS-Gaming**: In this study, a survey was sent over email asking about video gaming habits, demographic information, and sleeping habits. The paper contains a table with summary statistics that include duplicate regions.
- **DS-Spider-E**: This study measured the "boldness" of spiders by recording how long it will take spiders to reemerge from their enclosure after a simulated predator attack. The dataset includes a large number of duplicates, as well as repeated regions.
- **DS-Fly**: In this study, the sizes of flies were measured, as well as the distance they traveled. Both measurements include values that have a high degree of precision, with roughly 16 digits after the decimal point, as well as values with a precision of two.

5.4 Artifacts of Manipulation

The act of manipulating or completely fabricating a dataset can leave behind signs: We call these signs artifacts of manipulation. As shown in Figure 5.1, we have organized these artifacts into four common categories: **formatting**, relating to how the data appears in the data files; **numerical**, relating to patterns of numbers and digits in and across columns; **structural**, relating to patterns that appear when analyzing multiple rows or columns together; and **domain**, relating to patterns that show impossible or implausible effects in the data given the meaning of the data. The types of artifacts we found in our ten datasets are summarized in Figure 5.2. It is important to note that since these artifacts are derived from a limited collection of manipulated datasets, the list is only a starting point. To track artifacts that may be discovered in the future, we have created a living document of artifacts and invite others to suggest changes. Also, the presence of artifacts is not always an indication of wrongdoing — they can be produced by a valid data processing

		DS-PRIMING	DS-DRIVING	DS-COVID	DS-GAMING	DS-SPIDER-P	DS-SPIDER-E	DS-SPIDER-I	DS-GLIOMA	DS-FLY	DS-FISH
Formatting	UNEXPECTED FORMATTING		●	●					●	●	
Numerical	DUPLICATE NUMBERS AND DIGITS					●	●	●	●		
	UNEXPECTED LEADING DIGITS										
	UNEXPECTED TRAILING DIGITS		●	●							
	UNEXPECTED VARIATION OF PRECISION						●			●	
Structural	REPEATED REGIONS	●			●	●	●	●	●		●
	ORDERING ARTIFACTS					●		●		●	
Domain	DEVIATION FROM DOMAIN EXPECTATIONS	●	●								

Figure 5.2: An overview of which datasets exhibit which artifacts.

step or be an artifact of the data collection methodology. Finally, it is sometimes difficult to distinguish intentional wrongdoing from honest mistakes while working with data. We discuss the implications of our visualization design in Section 5.5 and the necessary care in our section on broader impacts (Section 5.9).

5.4.1 AR-Formatting: Formatting Artifacts

Spreadsheet tools like Excel or Google Sheets allow users to format the appearance of the data, including choosing a *font*; *font size*; methods of text emphasis such as *bold*, *italics*, and *underlining*; and background *colors*. In addition, users can select a *data format*. For example, changing a cell to a date format will alter how the cell is displayed without changing the underlying information. These formats can be flexibly chosen for cells, columns, or rows, and combinations thereof. Formatting is typically consistent and logical in inconspicuous data. However, the occurrence of odd patterns of formats can hint at manipulation, as illustrated in Figure 5.1a. For instance, in the study on honesty (**DS-Driving**), it appears that data was copied to a temporary file, where the values were modified, and later copied back into the master spreadsheet. Notably, these two files seem to have used different fonts, so in the final dataset, exactly half of the rows contained text using the font *Calibri* and the other half using *Cambria*. These rows were interspersed, likely due to shuffling or sorting the table after augmenting it. In total, we found evidence of strange formatting in three of the datasets we collected (**DS-Driving**, **DS-Covid**, **DS-Fly**).

This type of artifact, however, could also appear in authentic data, for example, when assembling a dataset from multiple data sources. Whether or not such a pattern is a sign of manipulation will depend on details, such as whether a whole column has a different format (likely not suspicious), or whether individual cells are formatted differently (possibly suspicious).

5.4.2 Numerical

The variety of possible numerical artifacts left behind by bad actors is considerable. Here we describe common types that we have observed. All these artifacts occur both in individual columns and across columns.

5.4.2.1 AR-Duplicate: Duplicate Numbers and Digits

This artifact describes cases when (whole) numbers or sequences of digits (parts of numbers) are repeated more frequently than expected (Figure 5.1b). Encountering duplicate numbers or digits can suggest that data was copied and pasted or manually entered. When measuring a natural phenomenon, there is typically variation in the data, either from differences in the signal being measured or from noise introduced by the tools used to measure the signal. For a specific number of values sampled from a distribution at a specified precision, a certain number of duplicate values can be expected. With more duplicates than expected, a few possible explanations can be suggested. First, the underlying distribution could be different than expected. For instance, a narrow Gaussian distribution would result in more duplicates than a wider one. Next, low precision generally would make duplicates more likely.

A common cause of duplicate numbers and sequences of digits that may seem suspicious at first, but is typically innocent, is high-precision duplicates caused by converting measurements. For example, converting fractions to decimals could introduce duplicates with seemingly high precision. If an experiment recorded the length of an animal in inches as integers, but in a subsequent step, the data was converted to feet using decimals, we would expect that the resulting decimals have values with high precision, such as 0.33333333 and 0.41666667. In this case, the number 0.33333333 may appear more often than naively expected and an n-gram of digits, such as 3s, or 6s may appear frequently.

Another common cause of duplicates that is likely innocent is thresholding or reaching a maximum value. In many scientific experiments, there is a terminating condition, such as a maximum time of the experiment or a score corresponding to a maximum achievable value. For example, whereas the spider datasets (**DS-Spider-E**) are definitely manipulated, they also measured only a time period of ten minutes, recorded as 600 seconds, and that maximum threshold was reached often. Hence, the frequent occurrence of 600 in such a dataset is likely inconspicuous.

Duplicate numbers can also appear when a dataset is manipulated by copying items or by manually inventing numbers. Humans are bad at generating random numbers [177]–[179] and random sequences of digits. When humans simulate the process of sampling from a distribution by repeatedly typing numbers, they tend to produce patterns (duplications) that often can be distinguished from collected data. In addition, sequences of digits appear more frequently in fabricated sets of numbers. For example, 54.23 and 23.54 are not duplicate numbers, but they do contain duplicate digit sequences, 54 and 23. We observed a suspicious amount of duplicated numbers and digits in four of our datasets (**DS-Spider-E**, **DS-Spider-P**, **DS-Spider-I**, and **DS-Glioma**).

5.4.2.2 AR-Leading: Unexpected Leading Digits

Benford's law [12], [25] is an expected pattern of the first digits of numbers in a dataset (Figure 5.1c). In short, it states that in datasets that span multiple orders of magnitude, the most frequent first digit should be a one, followed by a two, then a three, and so on. For example, in a dataset of the number of people living in cities and villages, we would expect more cities with 100,000–199,999 inhabitants (leading digit 1) than cities with 900,000–999,999 inhabitants (leading digit 9). We have included this artifact in our collection since checking for violations of Benford's law is a known technique for unearthing fabricated data. However, none of the scientific datasets in our collection spreads densely over such multiple orders of magnitude; hence, we did not identify this pattern.

5.4.2.3 AR-Trailing: Unexpected Trailing Digits

We have also found it useful to examine the last digit of numbers (Figure 5.1d). In some situations, the last digit of a collection of measurements might represent a randomly sampled uniform distribution. In other situations, different patterns would be expected. For example, in a list of prices for grocery-store products, an increased frequency in the digit nine would be expected since prices ending with 99 are strategically selected to make a product appear cheaper. On the other hand, if people are asked to provide an estimate for a value, we expect a final digit of zero to be more frequent than other digits. For example, if participants at a large event were asked how many people attended, we would expect an answer of $15,000$ to be much more common than $14,872$. We consider a trailing digits artifact to be a mismatch between the expected pattern of the last digit and the pattern observed in the data or an unexplained inconsistency of trailing digits between parts of a dataset.

In **DS-Driving**, two columns represent values where drivers are asked to give the mileage of their car. In one column, this rounding effect — showing a large amount of numbers ending with zeros — is present. In the other column, the trailing digits follow a uniform distribution. The retracted manuscript does not describe any difference in data collection between these two columns that might explain the difference in pattern between the final digits.

Although this example could be identified by shifting the decimal place and performing an analysis on precision, this is not always the case. The frequency of numbers ending in nine would not be noticeable in a precision analysis. Furthermore, a precision analysis of **DS-Covid** would not catch a strange pattern where even trailing digits occur more frequently than odd digits. Vice versa, not all precision artifacts are noticeable through a trailing digit analysis. For instance, the variance of precision of the stopwatch example is independent of the frequency of different trailing digits.

5.4.2.4 AR-Precision: Unexpected Variation of Precision

Data formatting in spreadsheet programs can also obfuscate data, leading to numerical artifacts that may not be evident in the source spreadsheet. In particular, this obfuscation can occur with the precision of numbers. Numerical data may record a varying number of digits after the decimal places. However, if the data is formatted as a *number*, the default in Excel is to show two digits after the decimal place. We assume inconspicuous data has similar precision for similar observations. Time measured with a stopwatch, for example, would typically have a precision of up to $1/100$ of a second. Most numbers should have two digits after the decimal points, a few with one digit (e.g., exactly 3.1 seconds), and even fewer with no digits (3 seconds). Manipulated data may have extremely varied precision

Figure 5.1e. This variation could happen if data is recorded (or generated) with a high degree of precision, then manually manipulated to change some values. Such a difference may not be apparent in a spreadsheet program when two digits are displayed. Alternatively, some authentic data could be collected with limited precision, and a function with high precision could be used to generate the rest. However, such a phenomenon could also arise innocently, e.g., when converting between fractions and decimals, as explained earlier. We have observed unexplained varied precision in **DS-Fly**.

5.4.3 Structural

Beyond the frequency of data or attributes of data, the structure of data can also play a role in detecting manipulation. Structural patterns are concerned with both the value of measurements and the order of the observations in the data file.

5.4.3.1 AR-Regions: Repeated Regions

Whereas six duplicate numbers may be considered a weak signal of manipulation, two identical sequences of six numbers are a much stronger one. We consider a region to consist of multiple cell values that have a spatial relationship in a spreadsheet, as illustrated in Figure 5.1f. Regions include nearby cells, vertically or horizontally, and may include gaps. Repeated regions can be artifacts of manipulation. Although some repeated regions could be caused by how the data is collected, such an innocuous structure is likely obvious. For manipulated data, regions are likely copied and pasted multiple times, either accidentally or as a convenient way to augment a dataset. In addition to simply copying and pasting regions, parts of the region are sometimes modified manually, resulting in similar regions with gaps. We saw this type of artifact in seven of our ten datasets, making it the most common artifact (**DS-Priming**, **DS-Gaming**, **DS-Spider-E**, **DS-Spider-I**, **DS-Spider-P**, **DS-Fish**).

5.4.3.2 AR-Ordering: Ordering Artifacts

It is natural for ordering artifacts to exist in authentic datasets. For instance, if multiple observations are recorded over time, we would expect that time increases throughout the dataset. Our interviews revealed that some experts consider it a good practice to avoid changing the order of a dataset. However, re-sorted data is not uncommon or automatically suspicious.

The ordering of the data can still indicate manipulation, as illustrated in Figure 5.1g. For example, if a bad actor wants to show that an experimental condition has an effect on the weight of animals, they might sort the data based on weight. Then, they might modify values at the distribution's tails — altering the data to match their hypothesis. This approach is economical since changing the extreme values will have the largest effect on aggregate measurements. However, this approach can leave behind ordering artifacts. This kind of dataset where a column is nearly sorted is one example of an ordering artifact. If the order is reset after modifications, such a pattern might be difficult to detect. However, if the data is reset by sorting on a column with duplicates (e.g., by a categorical value), then the effects of sorting on weight before the reset will still be seen within the groups. This kind of ghost sorting is another variation of an ordering artifact. A different order artifact exists in **DS-Fly**. Here one column has a mixture of high and low precision (**AR-Precision**). Additionally, the cells with low precision do not appear to be randomly interspersed throughout the rows, but rather appear in a repeated structured way.

5.4.4 AR-Domain: Deviation from Domain Expectations

The artifacts discussed so far are visible in the data's formatting, structure, or values. However, authors may use more sophisticated techniques, such as scripts that randomly sample from a distribution, to generate fabricated data. Manipulation in these datasets may be difficult or even impossible to detect. In these situations, more sophisticated techniques are required to find the artifacts.

5.4.4.1 Single-Dimensional

For single-dimension data, we often have prior knowledge about how that data should look, at least in the aggregate. For instance, many natural measurements, such as the height of humans, will exhibit a normal distribution. We consider drastic variations from these expectations, such as a uniform distribution occurring when a normal distribution is expected or a normal distribution with an obviously clipped tail, to be a single-dimensional domain artifact.

5.4.4.2 Relational

With prior knowledge of how data should look and some scripting ability, it may be possible to fabricate a column of data that is indistinguishable from authentic data. However, such fabrication becomes increasingly difficult as more columns are included. For example, if an experiment records the height, weight, age, sex, lab values, etc, it would be more difficult to generate the data while ensuring all relationships remain plausible. In other words, scripts that generate data would have to ensure that height correlates with weight, age, sex, etc. — it is insufficient for each column to pull from its underlying distribution independently.

Relational artifacts can be more nuanced than a missing correlation. In **DS-Driving**, for example, it appears that some data has been programmatically generated by adding a random value between 0 and $50,000$ to create a new column for $13,488$ rows. When comparing the relationship of the cars' mileages before and after a period of time, the miles driven in this period are uniformly distributed between 0 and $50,000$, with many drivers close to the $50,000$, violating an assumption of a smooth distribution.

5.5 Visualization Design Principles

Analyzing datasets for manipulation is a difficult and potentially fraught endeavor. A claim of manipulation, even during the review process, is a serious accusation and should be levied with caution. Hence, we believe domain experts must maintain agency while analyzing a dataset. Analysis tools should support experts by providing guidance without being prescriptive. To realize this sentiment, we developed design principles to guide our development of Ferret, a visualization tool for reviewing tabular datasets for manipulation. In this section, we introduce these design principles, and we describe the particulars of Ferret in the next section. Some of our guidelines are related to general visualization guidelines, such as Shneiderman's Mantra (overview first, zoom and filter, details on demand) [4], yet we provide more specific guidance for the use case of detecting manipulations in datasets.

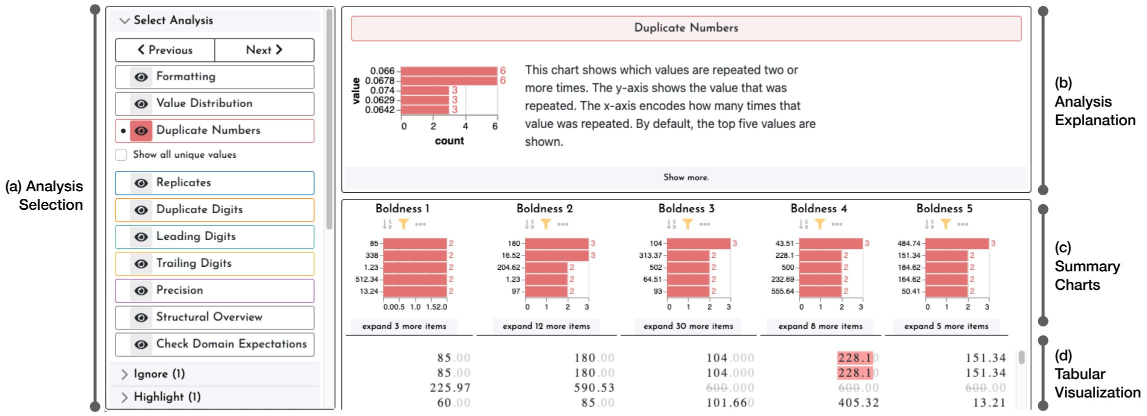
5.5.1 Framing the Exploration

One early observation we made is that the breadth of artifacts of manipulation is significant and that domain knowledge about the data is often necessary to make accurate judgments. Hence, we argue that a human, ideally with domain expertise, is needed to discern whether an artifact is the result of manipulation. However, we also learned that typical exploration of data through visualizations and statistical tests might be insufficient without knowledge of what to look for. In other words, "until you know what to look for, the patterns are not obvious" [169]. As a result, our first design principle is to provide guidance by describing artifacts of manipulation and why they may exist (including benign explanations), and providing salient visualizations of artifacts. At the same time, we avoid being prescriptive, for example, by describing why a dataset is manipulated or recommending a particular analysis or statistical test.

This principle is manifested in Ferret in several ways: First, Ferret lists and explains the different types of artifacts (Figure 5.3a and b). The introductory text gives advice on how to spot an artifact but also lays out common benign causes of those artifacts. Second, Ferret provides visualizations to identify and confirm suspicious patterns (Figure 5.3c and d). Finally, Ferret refrains from using statistical tests to identify issues. Our argument for not using tests is that most tests would be valid only under narrow circumstances (such as a specific type of distribution), and that the danger of inappropriately using a test outweighs the benefits.

5.5.2 Make Artifacts Salient

Our next design principle is to make artifacts of manipulation salient. Since our first design guideline necessitates an expert human to investigate the data, our goal is to make that investigation more efficient by quickly exposing artifacts of manipulation. This principle is best illustrated with an example: It can be difficult to notice things like the difference between Calibri and Cambria fonts or between 11.9 and 12-point font sizes in traditional spreadsheet software. In Ferret, cells with deviating formatting are highlighted with a distinct background color and pattern (Figure 5.7a) so that differences are salient. Similarly, spreadsheet tools will often round decimals in their display to two digits. Ferret will display all the digits recorded and aligned at the decimal point (Figure 5.5a).



This chart shows which values are repeated two or more times. The y-axis shows the value that was repeated. The x-axis encodes how many times that value was repeated. By default, the top five values are shown.

Boldness 1	Boldness 2	Boldness 3	Boldness 4	Boldness 5
85.00	180.00	104.000	228.10	151.34
85.00	180.00	104.000	228.10	151.34
225.97	590.53	~~600~~.000	~~600~~.00	~~600~~.00
60.00	85.00	101.660	405.32	13.21

Figure 5.3: Overview of the Ferret visualization tool. (a) The *Analysis Selection* panel gives quick access to the available analysis modes. (b) An explanation is provided for each analysis mode to help users understand artifacts of manipulation and guard against false positives. (c) The *Summary Charts* display aggregate information for each column in the dataset. (d) The *Tabular Visualization* gives access to the raw tabular data with relevant data surfaced through highlighting and rearranging. Values can be highlighted (such as 228.1) and ignored (600).

5.5.3 Use Overview and Details

A well-designed visual overview handles large datasets and helps analysts quickly spot suspicious patterns. At the same time, making the raw tabular data a first-class citizen within the visualization is essential. Only access to the raw data enables an analyst to confirm their suspicion or identify a benign explanation. In other words, any overview visualizations should be tightly integrated with a visualization of the details. If an interesting feature is noticed in an overview, it should be possible to query for details and easily see the rows generating that feature, as shown in Figure 5.3d. Conversely, if an interesting pattern is found by inspecting the raw tabular data, it should be easy to switch to the overview visualization and observe that pattern from a higher vantage point (Figure 5.5b and Figure 5.7b).

5.5.4 Leverage Interactivity

While investigating artifacts of manipulation, interactive sorting and filtering is essential. **Sorting** by different columns provides many ways to view the data, and combining this ability with different visual encodings can reveal interesting patterns, such as alternating fonts Figure 5.7b. **Filtering** is useful for focusing/excluding specific items. However, unlike most systems, ignoring only the values of specific cells (in contrast to filters that remove a row from a dataset) is more useful for detecting manipulations. For instance, in the case where values are clamped to an upper bound (**DS-Spider-E**), there may be many duplicates. Such duplicates will affect the analysis of several artifacts. Excluding those frequent values from the analysis is a convenient way of running the visualizations on the remaining data without excluding entire rows (Figure 5.3d).

5.6 Visualizations in Ferret

Ferret is based on the design principles described to surface artifacts of manipulation. Ferret provides various visualizations for different aspects, yet some visual encodings can be used for multiple patterns. At the heart of Ferret is a tabular visualization technique [51],[52] combining spreadsheet-like raw values with graphical marks, with a series of custom visual encodings, enriched by a set of supplementary views.

Formatting Ferret uses dedicated visual encodings for formatting artifacts within the tabular visualization and considers font styling and emphasis as well as the *data format*. Ferret does not use the styling of the source, since the exact formatting is usually immaterial for detecting manipulations. Instead, our encoding emphasizes the differences in formatting: The most frequent combination is assigned the default white background. All other unique combinations of formats are assigned a background color and texture/pattern (see Figure 5.7a). We chose to use five patterns and seven colors (35 combinations) because the number of unique combinations can exceed the number of reasonably distinguishable colors. When a cell is selected, the exact formatting parameters and a count for the number of cells that share the same formatting are listed.

Summary Charts: Counts, Proportions, and Distributions We use histograms and bar charts to visualize distributions (how values in a column are distributed), counts (how often a number is duplicated), and proportions of values (what percentage of numbers has a precision of 2). To view the **counts of values**, we use horizontal bar graphs (Figure 5.4a), which is useful for visualizing the count of duplicates and duplicate digits **AR-Duplicate**. These graphs can contain long labels, which is well suited for a horizontal layout. In Figure 5.3c, the duplicate numbers of one of the spider datasets (**DS-Spider-E**) are shown at the top of the five numerical columns. The duplicate digits chart works analogously; instead of visualizing duplicated whole numbers, it shows duplicated sequences of digits (2- or 3-

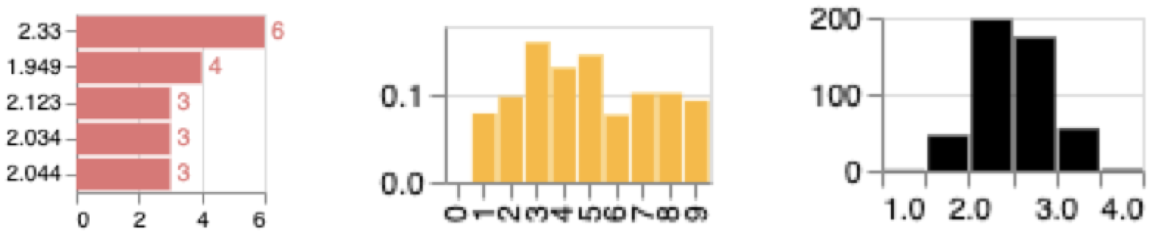
grams). To view the **proportion of values** with certain properties, Ferret shows vertical bar charts, where each bar shows a percentage of the property on the overall column (Figure 5.4b). We use proportion bar charts to show the frequency of trailing and leading digits (**AR-Leading** and **AR-Trailing**), as well as the frequency of precisions (**AR-Precision**). Finally, we use a histogram to show the **distribution of values** (Figure 5.4c), which is useful for sanity checks and alignment with domain expectations (**AR-Domain**).

Tabular Visualization These summary visualizations are tightly integrated with the tabular visualization. Using the summary charts, values can be selected or filtered. In Figure 5.3 the number 600 has been filtered out, which removes it from the bar chart, and strikes it out in the tabular view. The value 228.1 has been selected, which highlights it in red.

Figure 5.5a shows another example of tight integration between the summary visualization on top and the tabular visualization below. The bar chart shows the proportions of different levels of precision, and the tabular visualization below shows data, highlighting the precision through alignment.

Structural Visualization For large tables, scrolling through the full dataset can be cumbersome, and raw numbers do not show structural effects well (**AR-Structural**). The table overview mode [52] in Ferret solves this problem by reducing the cell height to at least one pixel, maximizing the number of rows visible on the screen (see Figures 5.5b and 5.7b). In overview mode, exact values are elided, and graphical representations are shown.

Domain Visualizations Finally, Ferret includes a suite of domain visualizations to help reviewers test the data for deviations from their domain expectations. Ferret supports faceted strip plots (Figure 5.6), scatterplots (Figure 5.9), violin charts, bar charts, and parallel coordinate plots. Although these visualizations share similarities with general visualization exploration tools, we believe their inclusion is useful due to convenience and

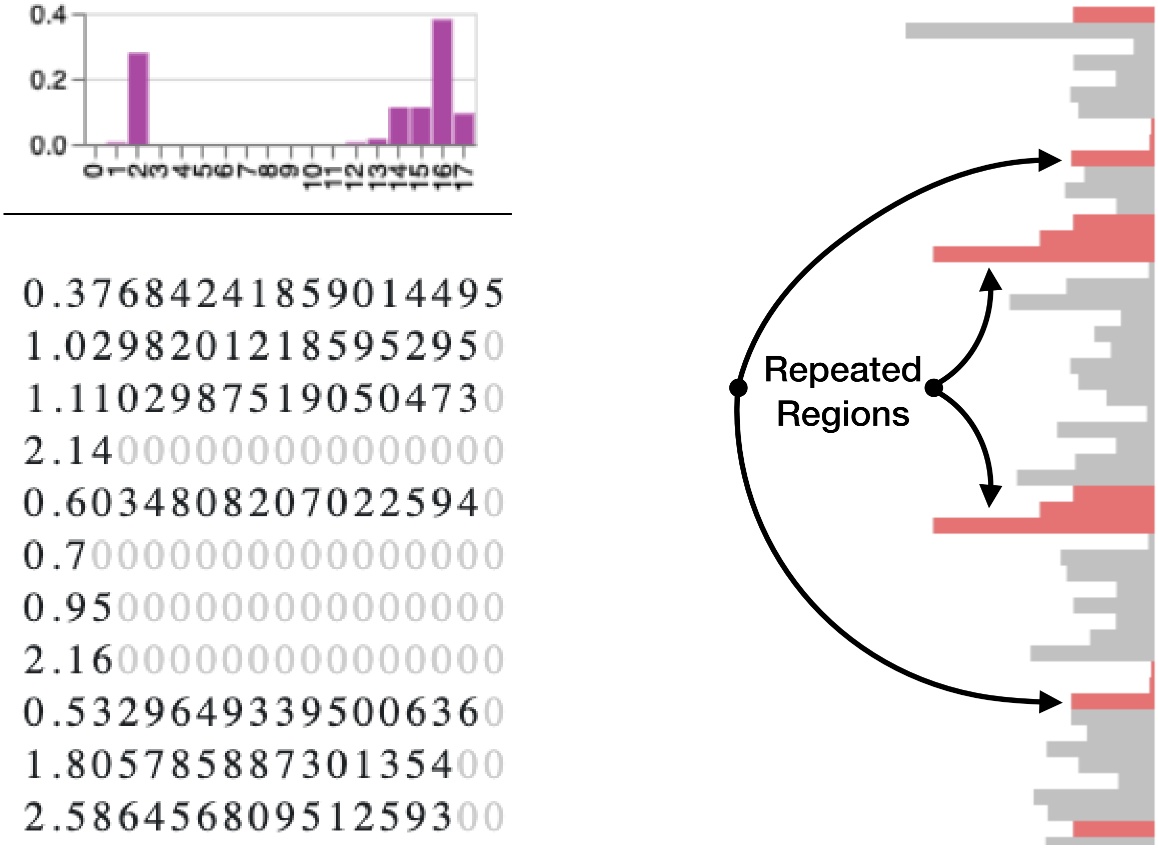


(a) Count

(b) Proportion

(c) Distribution

Figure 5.4: Different summary visualizations available in Ferret.



(a) Precision

(b) Structural

Figure 5.5: Visualizations for precision and structural artifacts. (a) A precision artifact (**AR-Precision**) is visible in **DS-Fly** through the proportion chart and the tabular visualization. (b) Repeated regions (**AR-Regions**) are visible for **DS-Gaming** using the overview.

the guidance the tool provides.

5.7 Implementation

Ferret is open source and implemented as a front-end web application. The code is available at `https://github.com/visdesignlab/ferret`, and a demo of the tool is available at `https://ferret.sci.utah.edu/`. The summary charts are built with Vega-Lite [180]. The table is built on top of LineUp [181] and Taggle [52] and uses custom code for

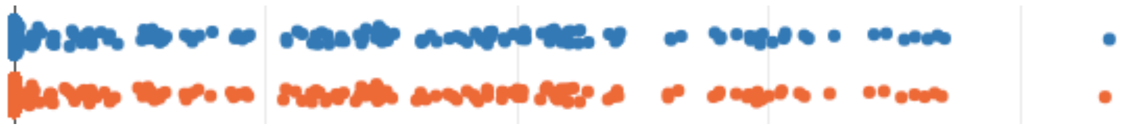


Figure 5.6: Strip-plot for miles driven in **DS-Driving** faceted by font. The data rendered appears to be duplicated with minor noise added.

cell rendering. The general visualizations are implemented with React and Plot.ly [182]. Ferret uses excel.js [183] to load and process Excel files, which can be uploaded by users.

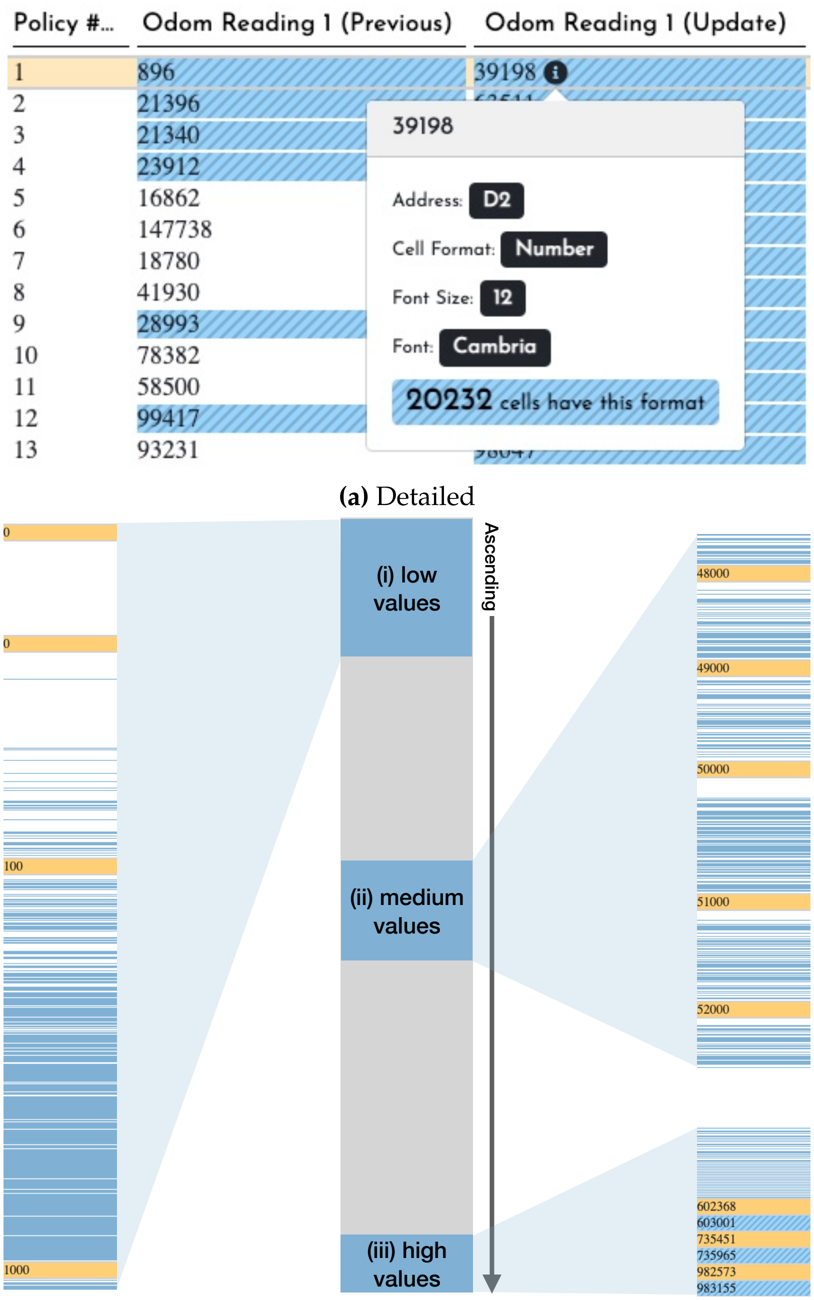
5.8 Case Study

In this section, we demonstrate the utility of the classification of artifacts, our design guidelines, and the Ferret prototype. We include case studies for all ten datasets in the supplementary material.

As our primary case study, we analyze the driving dataset (**DS-Driving**) and recreate and expand upon the analysis in a blog post that led to the paper's retraction [162]. The post discusses four anomalies and provides two hypotheses to explain them. A similar analysis with Ferret unearths the same and some additional anomalies, sometimes using different kinds of visualizations, that support the claims of the blog post. Upon loading a dataset, Ferret displays the Formatting visualization (**AR-Formatting**), also showing instructions on when to use it and how to read it. For the driving dataset (**DS-Driving**), suspicious formatting in the second (*Odom Reading 1 (Previous)*) and the third (*Odom Reading 1 (Update)*) columns is immediately obvious. Mixed formatting within a single column, as seen in Figure 5.7a, is unusual and suspicious. The second column has a mixture of Calibri and Cambria fonts. If this formatting was the only artifact found in a dataset, an editor could ask the authors for an explanation. However, further investigation reveals additional irregularities. Switching to the overview mode allows a faster review of the table's 13, 488 rows. The pattern of seemingly random mixes of fonts continues throughout the column. Yet, sorting the data reveals several patterns, shown in Fig 5.7b.

First, most values less than 100 in this column are in Calibri font, and all rows with a value of zero are in Calibri . Conversely, values between 100 and 1000 are predominately Cambria . For the remaining data, the two fonts are interspersed, except for certain regions, where Calibri dominates. Inspecting the values reveals they are duplicate round numbers, such as 75, 000. Since these values represent self-reported car mileage, the data makes sense if people estimate the mileage of their car. Suspiciously, these rounding effects are not visible for the values in Cambria font, suggesting that the data collection method for the two fonts diverges.

Finally, the high values (Figure 5.7b) alternate perfectly between Calibri and Cambria .



Policy #...	Odom Reading 1 (Previous)	Odom Reading 1 (Update)
1	896	39198
2	21396	
3	21340	
4	23912	
5	16862	
6	147738	
7	18780	
8	41930	
9	28993	
10	78382	
11	58500	
12	99417	
13	93231	

39198

Address: D2

Cell Format: Number

Font Size: 12

Font: Cambria

20232 cells have this format

(a) Detailed

(b) Overview

Figure 5.7: Visualizing formatting artifacts with color and patterns for **DS-Driving**. (a) The most frequent type of formatting is not highlighted (white background). All other formatting combinations are assigned a unique pattern/color combination. A tool-tip shows the formatting details on demand. (b) Showing structural patterns related to formatting. The pull-outs i-iii are taken from a large column, illustrated schematically in the center. Low values (i) are formatted in Cambria (white), and (ii) medium values alternate between Calibri and Cambria (blue), with Cambria clusters of round numbers. High values (iii) alternate between the fonts.

Closer inspection reveals that every value styled in Calibri font has a corresponding Cambria value that is within 1000 miles. This pattern suggests data was copied and a random number between 1 and 1000 was added. Visualizing this column (Figure 5.6), reveals that the two datasets are extremely similar.

A different approach to analyzing this dataset is to look at rounding effects. Fig 5.8 reveals tell-tale signs of rounding using the duplicate numbers, duplicate digits, and trailing digit frequency charts for the first column (the initial odometer reading). However, the second column (the follow-up reading after some time has passed) does not show any rounding effects.

To further explore this difference, we visualize the relationship between these two variables with a scatterplot. Figure 5.9 shows that the miles driven never exceed 50, 000, and the distribution of miles driven is uniform between zero and 50, 000, an unlikely distribution for this dataset, supporting a hypothesis made in the blog that the odometer readings in the updated column were generated by adding a random number between 0 and 50, 000.

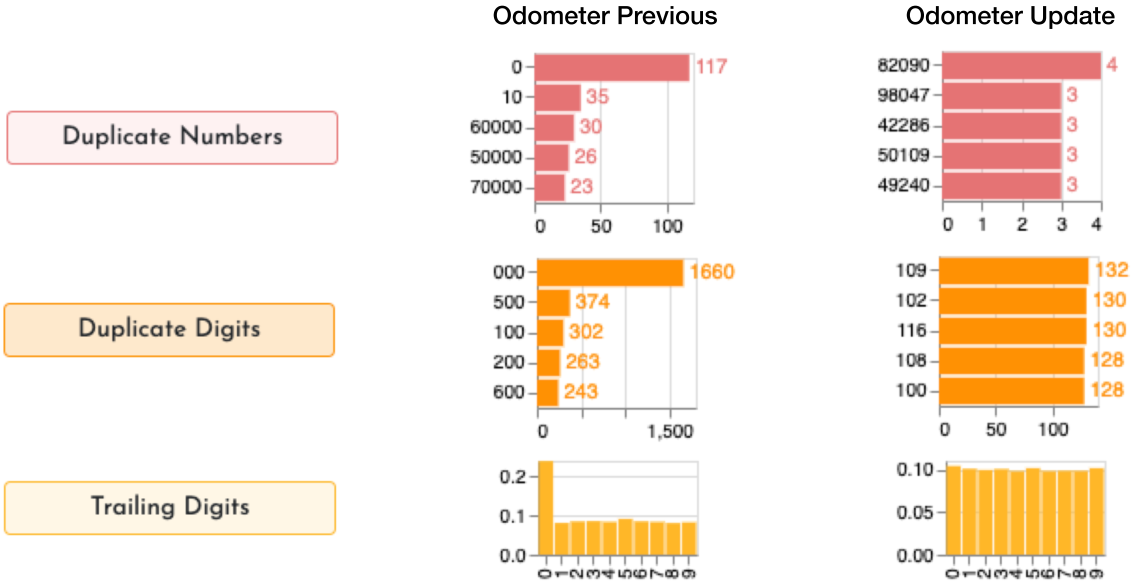


Figure 5.8: Rounding effects are clearly present in the duplicate numbers, duplicate digits, and leading digit frequency charts for the initial odometer reading (left column) but suspiciously absent for the follow-up reading (right column) in **DS-Driving**.

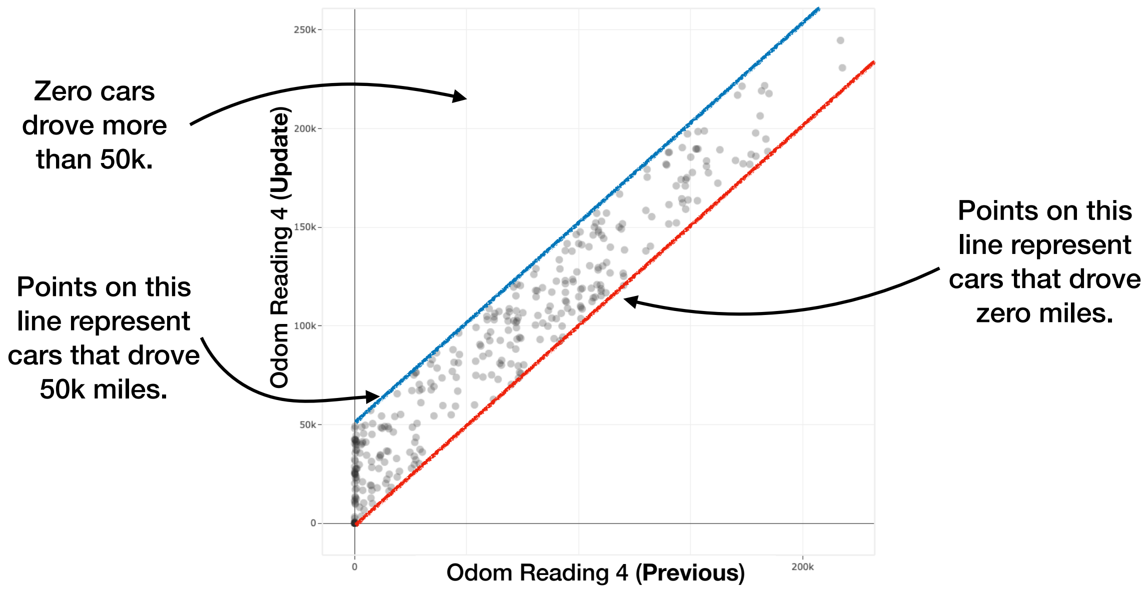


Figure 5.9: Scatterplot used to analyze a deviation from the domain expectation (**AR-Domain**). The x-axis corresponds to odometer readings taken at the beginning of the study, and the y-axis corresponds to readings taken at a later time, as recorded in **DS-Driving**. Note that the distribution of miles driven seems uniform, up to a hard cut-off after $50,000$ miles, indicated by the blue line.

5.9 Discussion and Broader Impacts

With the goal of increasing the trustworthiness of scientific research, our work collects manipulated datasets, categorizes artifacts of manipulation, designs visualization methods to explore them, and prototypes a tool to make those artifacts salient. We observe that manipulated datasets tend to present multiple artifacts simultaneously, which can be spotted with different techniques offered by Ferret. Hence, we believe that our approach of providing multiple visualizations that are easy to step through and interpret is a robust method for spotting artifacts and minimizing risks. That said, several potential unintended consequences from our work could affect researchers and society in general.

5.9.1 False Positives

One concern is the possibility that our methods indicate that data has been manipulated when, in reality, it has not. To address this concern, we suggest that when artifacts are identified, they should be used as a means of discussion with the authors, not as indisputable evidence of wrongdoing. This concern is also one of the reasons we believe statistical tests or summary reports would be treacherous, as they might reduce the nuance and complexity of the topic to simplistic answers. Still, a tool such as Ferret can make it easier to levy accusations against authors. An overly zealous individual could cause harm if they place too much confidence in individual artifacts of manipulation and do not give authors opportunities to respond. In the worst case, bad actors could use a tool like Ferret to maliciously target individuals. To remedy this problem, we suggest that Ferret should predominantly be deployed for general checks as part of the review process or when there are reasons to suspect wrongdoing with a paper.

5.9.2 Shaming

Our hope in collecting references to the manipulated datasets in this chapter is that it will be a resource for others interested in investigating data manipulation. However, our work may lead to additional unwanted attention for the authors of these datasets. To minimize the potential impact of our actions, we have published only datasets that come with an official retraction or an expression of concern from the publishing journal.

5.9.3 Security Theater

Reviewers and editors are often volunteers; hence, limiting their workload is an important consideration, especially if the additional work would be akin to useless “security theater”. Similar concerns can be raised about plagiarism checkers, yet they have detected numerous cases of plagiarism. We also attempt to make Ferret easy to use to avoid unnecessary burdens. However, conducting a cost-benefit analysis in a trial run with a selected journal is a logical next step.

5.9.4 Abuse

Knowledge about artifacts of manipulation and the existence of tools to identify them may help bad actors avoid detection of their misconduct. Experience with plagiarism detection tools shows that they continue to catch manipulation. Although we cannot ensure that abuse will not happen, we hope that the burden of “engineering” a dataset that does not raise suspicion is so high that bad actors may conclude that manipulation is not worth the risk.

5.9.5 Data Sharing

Using tools like Ferret may disincentivize authors to submit data with their manuscripts for fear of being unjustly accused of manipulation. Many journals and conferences already require the publication of data. Some researchers may choose to publish with journals that do not. We hope that the scientific community can meet this challenge by (a) carefully using tools like Ferret and (b) more broadly endorsing open science practices.

5.10 Conclusion

We believe our work will help future reviewers "ferret out" manipulations in tabular datasets. Knowing what artifacts of manipulation to look for will help analysts focus their search. Our design guidelines will aid in the development of tools for performing data forensics. Finally, Ferret is a first step toward instantiating this knowledge in a tool. Due to the adversarial nature of catching data manipulation, designing a single static tool is likely impossible. However, we believe our approach, which emphasizes the importance of the human-in-the-loop, is robust to changes in future manipulation techniques.

The approach of guiding users through a set of visual analyses can be adapted over time by adding additional analyses. Furthermore, this approach is not prescriptive and allows users to perform other arbitrary explorations (*Flexible-Structure*). Finally, including the raw tabular data alongside the various visualizations of transformed data allows the reviewer of data to perform detailed and broad analysis (*Trees-and-Foreset*).

Table 5.1: Table of datasets associated with retracted or withdrawn papers. Clicking on the dataset name will open Ferret with the dataset loaded. The *Status* column indicates whether a paper was retracted (R), withdrawn (W), or has earned an expression of concern (C). References in the *Statement* column link to the retraction statement. References in the *Blog* column link to blog posts that discuss how the data was manipulated.

Name	Status	Statement	Domain	Blog
DS-Priming	R	[159]	Mrkt.	[160] [161]
DS-Driving	R	[162]	Psy.	[163]
DS-Covid	W	[164]	Med.	[165]
DS-Gaming	R	[166]	Med.	[167]
DS-Spider-P	R	[168]	Bio.	
DS-Spider-E	R	[170]	Bio.	[169]
DS-Spider-I	R	[171]	Bio.	
DS-Glioma	R	[172]	Med.	
DS-Fly	C	[173]	Bio.	[174]
DS-Fish	R	[175]	Bio.	[176]

CHAPTER 6

DISCUSSION

Quality control of scientific data can be a daunting task. In Chapter 1, we discussed two *tensions* that contribute to the difficulty. The first tension, *Trees-and-Forest*, is the synthesis of detailed information with broad understanding. The second tension, *Flexible-Structure*, is between a desire for a flexible and structured analysis. In this thesis, **we claim that data visualizations are well suited for easing these tensions and, thus are a useful tool for scientific quality control**. This section will provide arguments for how visualizations are useful for easing these two tensions, backed by specifics from the three projects included in this dissertation

6.1 Easing Tensions with Visualizations

6.1.1 Trees-and-Forest

First, we will discuss the tension between detailed and broad analysis of data (Trees-and-Forest). The most straightforward way that visualizations are helpful here is to connect data points to the larger distributions. For instance, in Chapter 3, exemplar cells are selected to be displayed in Loon. These exemplar data points are useful to display directly because they contain rich information. Specifically, the images of the cells show the behavior of the cells in a way that cannot be fully captured by the derived attributes of the images. Furthermore, showing the images of cells with the computed cell boundary overlaid allows an inspection of the segmentation algorithm results. These are all detail-focused investigations of individual cells. However, in a given experiment, there can be millions of cells. It is impossible to review each of these in detail, and reviewing a few in detail may give a biased perspective of the dataset. To account for this, the detailed data points can be selected automatically and situated within the broader distribution. In the exemplar view of Loon, this happens by drawing a mark between the exemplar data point and where that point lies within a distribution of values.

6.1.1.1 Multimodal Data

To fully understand a cell's behavior, it is useful to investigate all of the data available and synthesize them together. For Loon, the exemplar view also facilitates this by showing line charts next to exemplar cell images. This proximity lets the researcher view the different aspects of the data together to create a more holistic view of the cell behavior. In Chapter 4, this approach is taken even further by creating composite visualization views. These views combine three different modalities of data together: images, time series, and trees. Each of these modalities conveys some aspects of the data effectively. The images show spatial relationships between cells, the time-series data shows trends of cell changes, the tree shows the cell relationships. Each piece can be useful for building an understanding of the data, but some tasks require combining all pieces together. For instance, checking the concordance of modalities is one way to evaluate the correctness of this data. In other words, do the different data modalities tell the same story? To do this, the different data modalities need to be connected. Finding the corresponding data elements across data modalities and matching them together is a detailed analytical task, but considering them together requires deep reflection of the data. The composite visualizations developed in Aardvark help relate [4] the data elements together. If we extend the trees and forest metaphor to understand even a single tree, you might consider the tree's height, width, species, age, leaf type, root distribution, etc. Visualizations that can connect these disparate pieces of information into an easy-to-interpret view allow a more holistic interpretation of the underlying phenomenon.

6.1.1.2 Data Transformations

Data transformations are used broadly in quality control and data visualizations. For instance, a common transformation performed on a collection of numbers is to count the frequency of numbers in binned ranges and then display those counts as a histogram. This representation of the distribution of values is useful for analysis and quality control. However, unique situations benefit from more specific data transformations. In Chapter 5 we show that investigating potentially manipulated data necessitates nonstandard data transformations. For instance, understanding the distribution of leading digits, trailing digits, or degree of precision is not likely to be helpful for data-driven discovery, but they can highlight signs that data has been manipulated. Visualizations can effectively show the transformed data and can be integrated with the raw data. The synthesis of various data transformations and individual data points is necessary for judging whether a dataset has been manipulated. This tension between understanding detailed information and broadly thinking about experimental procedures and data processing is facilitated by an interface that can sensibly integrate the different pieces of data. Similar to the previous section where the visualizations helped relate data elements across modalities, the ability to relate raw data to derived data helps users interpret the detailed information together.

6.1.1.3 Generalization

These three projects demonstrate how visualizations have been helpful for scientific data quality control. Although these projects have a relatively narrow scope, we argue that the reasons that visualizations are helpful generalize more broadly. The common strength across these projects is that visualizations can be composed to effectively integrate different aspects of data into a more cohesive whole. This visual linking can occur across data modalities, levels of detail, and data transformations. The projects in this thesis have resulted in various visualizations specific to those domains that link data in this way. Some of the specifics of those visualizations are unique to those domains, but the need to link across these dimensions more generally is not. It is common in science to record the same phenomenon with different instruments, resulting in multiple data modalities. It is common to collect a large amount of detailed data, necessitating investigating the data at multiple levels of detail. It is common to perform data transformations to highlight patterns in data. All of these challenges are common in scientific data quality control, and visualizations can ease the mental burden of the researcher by integrating these data dimensions for them.

6.1.2 Flexible-Structure

6.1.2.1 Sampling

The second tension (*Flexible-Structure*) is between a desire for both a structured and flexible approach to quality control. Here, we review how we created visualizations that can account for both of these desires. One scenario that leads to this tension is a large dataset with detailed data elements. To assess the quality of such a dataset, it is necessary to inspect individual data points. However, selecting which data points to inspect can lead to issues. A completely user-driven selection of data points can lead to a biased interpretation of the larger dataset. Alternatively, a strict sampling of data points prevents experts from investigating data points of interest. The solution is to allow both approaches; visualizations can do this effectively. In Chapter 3, the cell exemplar view automatically selects a sample of cells with varied attributes across experimental conditions. Additionally, this interface allows users to select any cell from the experiment to be displayed in this view alongside the automatically selected ones. Similarly, in Chapter 4, the tree-first view shows image snippets of cells across a cell's lifetime. Since this can include hundreds of images, the image frames are sampled. Here, images are automatically selected based on when the cell exhibits large changes in attributes, but the user can also select any point in the cell's lifetime to view that snippet. These both illustrate how visualizations can provide structure in data point selection while still allowing experts the flexibility to investigate any part of their data.

6.1.2.2 Guidance

Scientific experiments are extremely varied and result in commensurately varied datasets. Despite this, there are still some commonalities that exist when reviewing the quality of these datasets. Recognizing the common review steps and performing them is useful, but it is insufficient for a comprehensive review of a dataset. Thus, there is a tension between a structured review that includes commonly useful analyses and a flexible approach that can adapt to each unique situation. In Chapter 5, we developed a visualization system, Ferret, that can account for both of these desires. Ferret was designed specifically to review datasets for signs that they have been manipulated. A review like this requires a deep understanding of the domain, the methods for collecting the data, and a detailed analysis of the data itself. As a result, reviews of datasets will often require unique investigations. However, we also found that manipulated datasets shared several common patterns that we call artifacts of manipulation. In our tool, we provide a way to quickly step through various visualizations that highlight these artifacts. This *directed* [90] analysis structure for performing the review. The analysis is not *prescriptive* but rather embedded into a flexible visualization system that includes a sortable table view and a generic visualization panel for more varied ways to interact with the data. Altogether, Ferret illustrates that useful knowledge can be embedded into visualization tools to provide guidance while still supporting flexible investigations.

6.1.2.3 Provenance

Exploratory data analysis is a useful technique for identifying errors in datasets. The ability to quickly and iteratively explore different aspects of the data, identify patterns, and ask questions can help identify the root cause of data quality issues. However, it is also useful to be able to review the actions taken in this process to help *orient* [90] the analysis or to quickly back up to a previous state of exploration and refine the approach [4]. In short, there is a desire for fast, flexible exploration of data with a structured way to navigate past exploration. In Chapter 4, we account for this tension in the Aardvark visualization system. In this system, different data points can be selected, and various visualization configurations can be changed. Together, these allow a flexible way to explore the dataset. Additionally, the Trrack library [148] is used to record the interaction provenance of the system. With this, every action taken in Aaardvark is recorded and displayed in a provenance graph view. This view can help the researcher reflect on the different actions they have taken and can navigate back to a previous state in their exploration. Together, provenance tracking in Aardvark provides a flexible data exploration tool that provides structure to the exploration.

6.1.2.4 Generalization

Again, these three projects are relatively narrow examples of how visualizations can balance structure and flexibility. However, the ideas here generalize more broadly. In essence, visualization systems can provide structure and flexibility by carefully designing the affordances of the interface. A visualization that *directs* or *orients* a users provides structure without losing flexibility. For any visualization system that shows a section of exemplar data points, those points can be chosen automatically while still allowing a free-form selection. This idea of automatically selecting reasonable defaults, but allowing arbitrary edits is widely applicable: from matrix visualizations that order rows and columns [184] to visualization design systems that recommend visualization encodings [185]. A guided analysis can be integrated into a flexible system, resulting in a *guidance-first* or *directed* interface for review. Alternatively, a flexible system can record and display actions taken by the user to provide an *exploration-first* or *oriented* design that still includes structure. In our examples we do not have any guidance systems that are *prescriptive*. There may be some scenarios where a *prescriptive* system can help help scientists review the quality of their data, but such a system is likely to be less generally applicable than a less prescriptive approach.

6.2 Design Guidelines

The lessons learned throughout this process can be distilled into a few visualization design recommendations when constructing visualizations for scientific data quality control.

6.2.1 Show Data Relationships

Scientific data can be large and complex. Understanding that requires: spanning the scale of data, from individual data points to the entire dataset, considering how different modalities of data combine, and determining the relationships of raw data and transformed data. In essence, all of these require understanding data relationships, and visualizations have the ability to show these relationships.

Whether the relationship is from a part of the data to the whole, across modalities, or from original data to transformed data, visualizations can make the relationship clearer by drawing marks connecting them, using color to implicitly link them, or placing them in close proximity. This is an important strength of visualization systems that should be utilized.

6.2.2 Automate Sampling but Allow Arbitrary Selection

For large datasets, it is sometimes necessary to sample individual data points. This sampling is useful, for instance, if individual data points have rich aspects that cannot be summarized without losing important context [7]. In these situations, the sampling of data points should be automated. Automated data point selection can occur quickly, speeding up the analysis process. Furthermore, automated selection strategies can bring forward data points based on some metric of interest or provide a representative subsample. This can help structure the exploration and guide the user towards some objective in the data. However, such a system can and should also let the user make arbitrary selections of data points. This gives the user the flexibility to investigate as needed.

6.2.3 Provide Guidance

To tackle large, complex datasets, visualization designers sometimes make powerful but complicated interfaces. Guidance has been proposed to help users navigate these interfaces [90]. There does not appear to be a downside to providing some kind of guidance, though the degree of guidance may vary. For instance, an expert analyst could embed their knowledge into a system that directs others toward useful steps. Alternatively, if a visualization designer prefers to provide a general-purpose collection of tools for an analyst and let the expert analyst decide how they want to use them, they can still provide a lower degree of orienting guidance. One way to do this that works generally is to record and show the visualization state and support navigating that state (undo/redo).

CHAPTER 7

FUTURE WORK AND CONCLUSION

7.1 Future Work

There are multiple avenues for expanding on this work. In this section, we describe two specific directions in detail. The first is focused on conducting studies to evaluate visualization designs included in Ferret (Chapter 5), and the second is on how we can build additional features into Loon and Aardvark (Chapters 3 4).

7.1.1 Evaluating Visualization Designs

The goal of Ferret (Chapter 5) is to create a visualization interface that can guide users while they review a tabular dataset for manipulation. In Ferret [9] we justify our design choice with case studies. However, it would also be informative to run a formal study to understand how more people reason using Ferret and to more broadly understand how they reason about data manipulation. There are several factors to the design of such a study: the **participants** of the study, the **data** the participants will interact with, the textual **prompts** given to the participants, the **tools** available to the participants, the participant **responses** requested, and the study **results**. We will describe options for each of these factors.

7.1.1.1 Participants

The goal with Ferret is to have reviewers or editors of academic papers use it in the review process. Therefore, the most representative pool of participants is people who have reviewed papers that also include tabular datasets. A larger pool of participants that would still be informative are people who work with tabular datasets. For this study, participants do not need to be data analysts with scripting knowledge but should demonstrate some data analysis proficiency with spreadsheet tools like Excel. Tools like Prolific [186] could be used to recruit online participants. Prolific includes many prescreening options to limit participants, and additional criteria can be included as a part of running the study. To recruit researchers who review papers with associated tabular datasets, it may be more practical to advertise to our academic communities, though this could bias the participants towards certain domains like visualization.

7.1.1.2 Data

There are several options for what tabular dataset to present to participants. One option is to include existing datasets that have led to papers being retracted. One advantage to this approach is that it has higher ecological validity — this is an actual dataset that a reviewer could have investigated. However, there are several challenges with this approach. First, most of these datasets are included in high-profile retractions. Some have even led to stories in the New York Times, Wall Street Journal, New Yorker, Forbes, NPR, and other popular news outlets. The problem with this is that participants may already be familiar with these stories and may even know details about the issues in the datasets. Another challenge is that it is impossible to say with certainty how these datasets were manipulated. Based on the patterns in the data, theories can proposed, but it is impossible to confirm these theories. By extension, this makes it impossible to know with certainty that the participant's theories about the data match how they were manipulated. There are a few alternatives to this approach.

The first alternative is a relatively small change that only addresses the first issue of people recognizing datasets. For this approach, the data could still be drawn from existing datasets, but the column headers are changed, and the prompt describing the data is also changed. This change would reduce the likelihood that people associate the dataset with any prior knowledge of these high-profile retractions. Although this approach maintains the advantages of using an existing dataset, it also carries the disadvantage of not knowing with certainty exactly how the data was manipulated.

Another approach would be to create dedicated datasets, introducing controlled manipulations. This could be done by taking a real dataset that has not been manipulated and then falsifying it with different techniques, such as augmenting it by duplicating regions of data or duplicating all the data and adding a random value to it. The data could also be entirely fabricated by typing all the values manually or using a script to generate them. The advantage of this approach is that we can directly compare any theories participants propose for how the data was manipulated to exactly how we manipulated it.

In addition to showing datasets that have been manipulated, we could also show datasets that have not been manipulated. In this case, we could show existing unmanipulated datasets without any changes. We could also augment these datasets by adding reasonable derived columns, but do it in a way that we know introduces similar patterns as the manipulated dataset. For instance, we could add a column that changes the units to introduce varied precision in the values (e.g., inches to feet). We could also copy part of the data into a new worksheet that has different formatting applied (e.g., a changed font), perform a valid operation on that data, and then copy the new data, with the different font, back into the original spreadsheet, resulting in varied formatting in the cells.

7.1.1.3 Prompt

When reviewing a dataset, there is important related information. In the situation we are considering, the associated manuscript contains much of this contextual information. Including an entire paper in a study is likely too much information and could distract the review from the data to the manuscript, however the dataset should still be contextualized with the prompt. That is, it should briefly explain what the data is, how it was collected, and what results the data indicates.

Additionally, different prompts could direct participants to investigate the data in different ways. For instance, the prompt could tell the participants they are supposed to review the quality of the data generally, or it could tell them to review the data for possible manipulation, or it could state that manipulation exists (even if it doesn't) and they are tasked with identifying it.

7.1.1.4 Tools

During the study, we could provide the full interactive Ferret interface for the participant to review the data with. The inclusion of interactive visualizations in remote studies has recently been made easier with the reVISit [187] framework. Including the full functionality of Ferret would provide the most ecological validity of how using Ferret affects the judgment of participants. It would also be interesting to provide a more basic spreadsheet tool to some participants to compare the different results, or even more targeted tools for specific types of manipulation, such as statistical tests for Benford's law.

7.1.1.5 Responses

After the participant has had the opportunity to investigate the data, we can require several responses. In a real situation where a reviewer has found something suspicious in a dataset, a common next step may be to communicate with the author of that dataset and ask about the suspicious pattern. Therefore, it could be interesting to require participants to write a mock email with any follow-up questions about the dataset. Next, we can require that the participant makes a decision on whether to accept or reject the paper based on the quality of the data. In addition to this binary decision, we can require a textual explanation for the decision that would be shared with a primary reviewer, as well as a confidence score for the decision.

7.1.1.6 Results

These responses would provide several ways to analyze the results. The binary decision would show the liklihood that manipulated papers were rejected (true positives), but also would show how many unmanipulated datasets were rejected (false positives). The textual responses could provide additional nuance for the decisions. These could indicate if the participants are correctly identifying not only if but also how the data was manipulated or are basing their choice on spurious relationships even if the decision was correct. These could also reveal additional ways to identify signs that a dataset has been manipulated. Finally, if interactions are recorded within the tools, then we could also analyze how the tools were used to interact with data.

7.1.2 Visualization Development

These projects and discussions have focused on how visualizations have been useful for identifying errors in data, but we have not explored the logical next step for fixing these errors. Visualizations have already been incorporated into data cleansing workflows [188], [189] and have even been used in cell microscopy pipelines to refine data [68]. Furthermore, visualizations can be integrated with AI models to help humans correct datasets [190]. One similar direction we could explore with our work is the integration of visualizations into a system for collecting ground truth datasets for training AI systems and using visualizations to evaluate those systems.

The creation of ground truth datasets for training AI models can be incredibly time-intensive. For example, in order to collect a dataset with individual cell labels, our collaborators first look through a dataset and identify individual cells that clearly belong to one of the categories of interest. Then, they record the unique ID for that cell. Once enough cell have been identified a model can be trained based ont his data to classify cells generally. The current visualization tools, Loon and Aardvark, are useful for this task in that they provide an interface that shows all the relevant information in order for the researchers to identify which class a cell belongs to, but it does not provide guidance for stepping through cells of interest, or recording labels. We could build on the existing visualizations and provide this framework to build better datasets more quickly.

The way users would use this system can be broken into three major steps. First, users can label an initial set of data with the assistance of unsupervised learning techniques. Specifically, clustering algorithms could be applied to cell metadata to produce groups of similar cells. These groups of cells could then be presented with techniques from Loon and Aardvark that compose all the relevant information for these cells in compact visualizations. The user could then review this group and reject cells that do not belong to the same category and then label all remaining cells simultaneously. The goal here is to label many cells more quickly instead of one at a time.

Second, once an initial dataset has been collected, a classifier could be trained on this data. This could be done primarily outside the visualization tool so that any type of classifier could be used. Finally, to evaluate this model, test data that was not included in the training data could be passed through the model to get its predicted labels. Then, those cells plus their predicted label could be presented in the visualization interface. Again, different visualizations would provide the user with enough information to determine the correct classification, and they could then accept the model's classification as correct or provide the correct label. This process would serve to evaluate the current model since, in essence, the user is labeling the testing portion of the dataset on the fly to compare with the model. At the same time, this would generate additional expert-labeled data. In the case where the model did not perform well enough, this new data could be fed back into the training data, and the process of training, evaluating, and labeling could continue iteratively.

Beyond this particular interface for cell labeling is a larger challenge. The integration of visualization tools with programming environments. This could be accomplished by bringing visualizations into programming environments, such as computational notebooks [191]–[193], or by providing command line tools for generating useful visualizations similar to UpSetR [194]. Alternatively, scripting capability could be embedded into the visualization system. This type of technical integration would be helpful for the visualization classification system, but would also be useful for a wide range of tasks for interacting with data, like data wrangling, data cleaning, and discovery.

7.1.3 Conclusion

Evaluating the quality of scientific data is an essential and challenging task. It requires a careful review by experts who are well-versed in data analysis and their scientific domain. In particular, two tensions in this review process make this process difficult. In this thesis, we argue that well-designed visualizations can ease these tensions and improve the review process. We support this claim by referencing specific cases from three projects where we have designed visualizations for quality control. Furthermore, we generalize why visualizations more broadly are well suited for this task and propose design guidelines to help others produce well-designed visualizations.

REFERENCES

[1] H. Lin, D. Akbaba, M. Meyer, and A. Lex, "Data Hunches: Incorporating Personal Knowledge into Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 504–514, 2022.

[2] S. Alspaugh, N. Zokaei, A. Liu, C. Jin, and M. A. Hearst, "Futzing and Moseying: Interviews with Professional Data Analysts on Exploration Practices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 22–31, 2019.

[3] T. Munzner, *Visualization Analysis and Design*. CRC Press, Taylor & Francis Group, 2014.

[4] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *Proceedings of the IEEE Symposium on Visual Languages (VL '96)*, pp. 336–343, 1996.

[5] A. Perer and B. Shneiderman, "Balancing Systematic and Flexible Exploration of Social Networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 693–700, Sept. 2006.

[6] A. Perer and B. Shneiderman, "Systematic yet flexible discovery: Guiding domain experts through exploratory data analysis," in *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*, pp. 109–118, ACM, 2008.

[7] D. Lange, E. Polanco, R. Judson-Torres, T. Zangle, and A. Lex, "Loon: Using Exemplars to Visualize Large-Scale Microscopy Data," *IEEE Transactions on Visualization and Computer Graphics (VIS)*, vol. 28, no. 1, pp. 548–258, 2022.

[8] D. Lange, R. Judson-Torres, T. Zangle, and A. Lex, "Aardvark: Composite Visualizations of Trees, Time-Series, and Images," Apr. 2024.

[9] D. Lange, S. Sahai, J. M. Phillips, and A. Lex, "Ferret: Reviewing Tabular Datasets for Manipulation," *Computer Graphics Forum*, vol. 42, no. 3, pp. 187–198, 2023.

[10] N. Boukhelifa, M.-E. Perrin, S. Huron, and J. Eagan, "How Data Workers Cope with Uncertainty: A Task Characterisation Study," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 3645–3656, Association for Computing Machinery, 2017.

[11] N. Mccurdy, J. Gerdes, and M. Meyer, "A Framework for Externalizing Implicit Error Using Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 925–935, Jan. 2019.

[12] S. J. Miller, *Benford's Law*. Princeton University Press, 2015.

[13] M. J. Nigrini and S. J. Miller, "Data diagnostics using second-order tests of benford's law," *Auditing: A Journal of Practice & Theory*, vol. 28, no. 2, pp. 305–324, 2009.

[14] L. M. K. Padilla, M. Powell, M. Kay, and J. Hullman, "Uncertain About Uncertainty: How Qualitative Expressions of Forecaster Confidence Impact Decision-Making With Uncertainty Visualizations," *Frontiers in Psychology*, vol. 11, p. 579267, Jan. 2021.

[15] S. Powell, K. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," *Decision Support Systems*, vol. 46, pp. 128–138, Dec. 2008.

[16] A. A. Bock, "A Literature Review of Spreadsheet Technology," *IT University of Copenhagen Technical Report Series*, p. 35, 2016.

[17] D. Sukhobok, N. Nikolov, and D. Roman, "Tabular Data Anomaly Patterns," in *2017 International Conference on Big Data Innovations and Applications (Innovate-Data)*, pp. 25–34, Aug. 2017.

[18] D. W. Barowy, E. D. Berger, and B. Zorn, "ExceLint: Automatically Finding Spreadsheet Formula Errors," *Proceedings of the ACM on Programming Languages*, vol. 2, pp. 1–26, Oct. 2018.

[19] S.-C. Cheung, W. Chen, Y. Liu, and Changxu, "CUSTODES: Automatic Spreadsheet Cell Clustering and Smell Detection Using Strong and Weak Features," in *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, May 2016.

[20] D. Jannach, T. Schmitz, B. Hofer, and F. Wotawa, "Avoiding, finding and fixing spreadsheet errors – A survey of automated approaches for spreadsheet QA," *Journal of Systems and Software*, vol. 94, pp. 129–150, Aug. 2014.

[21] P. Koch, K. Schekotihin, D. Jannach, B. Hofer, and F. Wotawa, "Metric-Based Fault Prediction for Spreadsheets," *IEEE Transactions on Software Engineering*, vol. 47, pp. 2195–2207, Oct. 2021.

[22] D. Li, H. Wang, C. Xu, R. Zhang, S.-C. Cheung, and X. Ma, "SGUARD: A Feature-Based Clustering Tool for Effective Spreadsheet Defect Detection," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 1142–1145, Nov. 2019.

[23] D. Li, H. Wang, C. Xu, F. Shi, X. Ma, and J. Lu, "WARDER: Refining Cell Clustering for Effective Spreadsheet Defect Detection via Validity Properties," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*, pp. 139–150, July 2019.

[24] Y. Huang, C. Xu, Y. Jiang, H. Wang, and D. Li, "WARDER: Towards effective spreadsheet defect detection by validity-based cell cluster refinements," *Journal of Systems and Software*, vol. 167, p. 110615, Sept. 2020.

[25] M. J. Nigrini, *Benford's Law: Applications for forensic accounting, auditing, and fraud detection*, vol. 586. John Wiley & Sons, 2012.

[26] C. Durtschi, W. Hillison, and C. Pacini, "The effective use of benford's law to assist in detecting fraud in accounting data," *Journal of forensic accounting*, vol. 5, no. 1, pp. 17–34, 2004.

[27] SC. Rife, MB. Nuijten, and S. Epskamp, "Statcheck: Extract statistics from articles and recompute p-values [web application]," 2016.

[28] D. Chawla, "Controversial software is proving surprisingly accurate at spotting errors in psychology papers." https://www.science.org/content/article/controversial-software-proving-surprisingly-accurate-spotting-errors-psychology-papers, Nov. 2017.

[29] T. Park, H. Song, and S. J. Lee, "Detecting and Recovering Integer Data Manipulated by Multiplication With a Nonintegral Real Number and a Rounding Operation," *IEEE Access*, vol. 9, pp. 57149–57164, 2021.

[30] T. Foltýnek, N. Meuschke, and B. Gipp, "Academic Plagiarism Detection: A Systematic Literature Review," *ACM Computing Surveys*, vol. 52, pp. 112:1–112:42, Oct. 2019.

[31] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, (New York, NY, USA), pp. 76–85, Association for Computing Machinery, June 2003.

[32] S. Bellon, R. Koschke, G. Antoniol, J. Krinke, and E. Merlo, "Comparison and Evaluation of Clone Detection Tools," *IEEE Transactions on Software Engineering*, vol. 33, pp. 577–591, Sept. 2007.

[33] F. Hermans, B. Sedee, M. Pinzger, and A. van Deursen, "Data clone detection and visualization in spreadsheets," in *2013 35th International Conference on Software Engineering (ICSE)*, (San Francisco, CA, USA), pp. 292–301, IEEE, May 2013.

[34] Y. Zhang, W. Dou, J. Zhu, L. Xu, Z. Zhou, J. Wei, D. Ye, and B. Yang, "Learning to detect table clones in spreadsheets," in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2020, (New York, NY, USA), pp. 528–540, Association for Computing Machinery, July 2020.

[35] E. Bik, "Science Has a Nasty Photoshopping Problem," *The New York Times*, Oct. 2022.

[36] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. Efros, "Detecting Photoshopped Faces by Scripting Photoshop," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10071–10080, IEEE Computer Society, Oct. 2019.

[37] H. Li and J. Huang, "Localization of Deep Inpainting Using High-Pass Fully Convolutional Network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Seoul, Korea (South)), pp. 8300–8309, IEEE, Oct. 2019.

[38] M. Barni, E. Nowroozi, B. Tondi, and B. Zhang, "Effectiveness of Random Deep Feature Selection for Securing Image Manipulation Detectors Against Adversarial Examples," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2977–2981, May 2020.

[39] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, "Constrained R-Cnn: A General Image Manipulation Detection Model," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE Computer Society, July 2020.

[40] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5012–5019, IEEE Computer Society, Jan. 2021.

[41] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, "MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 3539–3553, Mar. 2023.

[42] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, "Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization," *International Journal of Computer Vision*, vol. 130, pp. 1875–1895, Aug. 2022.

[43] J. Tukey, "The Future of Data Analysis," vol. 33, p. 13, Annals of Mathematical Statistics, 1 ed., 1962.

[44] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer, "Visplause: Visual Data Quality Assessment of Many Time Series Using Plausibility Checks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 641–650, Jan. 2017.

[45] Z. Wang, X. Yuan, T. Ye, Y. Hao, S. Chen, J. Liangk, Q. Li, H. Wang, and Y. Wu, "Visual data quality analysis for taxi GPS data," in *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 223–224, Oct. 2015.

[46] D. Teng, H. Yang, C. Ma, and H. Wang, "VDQAM: A toolkit for database quality evaluation based on visual morphology," in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 245–246, Oct. 2012.

[47] M. Correll, M. Li, G. Kindlmann, and C. Scheidegger, "Looks Good To Me: Visualizations As Sanity Checks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 830–839, Jan. 2019.

[48] A. McNutt, G. Kindlmann, and M. Correll, "Surfacing Visualization Mirages," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, (Honolulu HI USA), pp. 1–16, ACM, 2020.

[49] M. Correll, "Teru Teru Bōzu: Defensive Raincloud Plots," *Computer Graphics Forum*, vol. 42, no. 3, pp. 235–246, 2023.

[50] Y. Zhang, K. Chanana, and C. Dunne, "IDMVis: Temporal Event Sequence Visualization for Type 1 Diabetes Treatment Decision Support," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 512–522, Jan. 2019.

[51] R. Rao and S. K. Card, "The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, (New York, NY, USA), pp. 318–322, Association for Computing Machinery, Apr. 1994.

[52] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, A. Lex, and M. Streit, "Taggle: Combining overview and details in tabular data visualizations," *Information Visualization*, vol. 19, no. 2, pp. 114–136, 2020.

[53] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit, "Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets," *IEEE Transactions on Visualization and Computer Graphics (InfoVis'14)*, vol. 20, no. 12, pp. 2023–2032, 2014.

[54] M. Blumenschein, M. Behrisch, S. Schmid, S. Butscher, D. R. Wahl, K. Villinger, B. Renner, H. Reiterer, and D. A. Keim, "SMARTexplore: Simplifying High-Dimensional Data Analysis through a Table-Based Visual Analytics Approach," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST '18)*, pp. 36–47, IEEE, 2018.

[55] C. Niederer, H. Stitz, R. Hourieh, F. Grassinger, W. Aigner, and M. Streit, "TACO: Visualizing Changes in Tables Over Time," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 677–686, Jan. 2018.

[56] A. Pandey, S. L'Yi, Q. Wang, M. A. Borkin, and N. Gehlenborg, "GenoREC: A Recommendation System for Interactive Genomics Data Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 570–580, 2023.

[57] S. L'Yi, Q. Wang, F. Lekschas, and N. Gehlenborg, "Gosling: A Grammar-based Toolkit for Scalable and Interactive Genomics Data Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 140–150, 2022.

[58] S. L'Yi and N. Gehlenborg, "Multi-View Design Patterns and Responsive Visualization for Genomics Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 559–569, 2023.

[59] A. K. Ai-Awami, J. Beyer, D. Haehn, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger, "NeuroBlocks – Visual Tracking of Segmentation and Proofreading for Large Connectomics Projects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 738–746, 2016.

[60] J. Beyer, J. Troidl, S. Boorboor, M. Hadwiger, A. Kaufman, and H. Pfister, "A Survey of Visualization and Analysis in High-Resolution Connectomics," *Computer Graphics F.*, vol. 41, no. 3, pp. 573–607, 2022.

[61] J. Troidl, S. Warchol, J. Choi, J. Matelsky, N. Dhanyasi, X. Wang, B. Wester, D. Wei, J. W. Lichtman, H. Pfister, and J. Beyer, "ViMO - Visual Analysis of Neuronal Connectivity Motifs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 748–758, 2024.

[62] R. Krueger, J. Beyer, W.-D. Jang, N. W. Kim, A. Sokolov, P. K. Sorger, and H. Pfister, "Facetto: Combining Unsupervised and Supervised Learning for Hierarchical Phenotype Analysis in Multi-Channel Image Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 227–237, 2020.

[63] A. Somarakis, M. E. Ijsselsteijn, S. J. Luk, B. Kenkhuis, N. F. de Miranda, B. P. Lelieveldt, and T. Höllt, "Visual cohort comparison for spatial single-cell omics-data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 733–743, 2021.

[64] S. Warchol, R. Krueger, A. J. Nirmal, G. Gaglia, J. Jessup, C. C. Ritch, J. Hoffer, J. Muhlich, M. L. Burger, T. Jacks, S. Santagata, P. K. Sorger, and H. Pfister, “Visinity: Visual Spatial Neighborhood Analysis for Multiplexed Tissue Imaging Data,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2022.

[65] J. Jessup, R. Krueger, S. Warchol, J. Hoffer, J. Muhlich, C. C. Ritch, G. Gaglia, S. Coy, Y.-A. Chen, J.-R. Lin, S. Santagata, P. K. Sorger, and H. Pfister, “Scope2Screen: Focus+Context Techniques for Pathology Tumor Assessment in Multivariate Image Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 259–269, 2022.

[66] A. Kerren and F. Schreiber, “Why Integrate InfoVis and SciVis?: An Example from Systems Biology,” *IEEE Computer Graphics and Applications*, vol. 34, pp. 69–73, Nov. 2014.

[67] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J.-K. Hériché, “Visualization of image data from cells to organisms,” *Nature Methods*, vol. 7, no. 3, pp. S26–S41, 2010.

[68] J. Choi, H.-J. Oh, H. Lee, S. Kim, S.-K. Kwon, and W.-K. Jeong, “MitoVis: A Unified Visual Analytics System for End-to-End Neuronal Mitochondria Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2022.

[69] J. Troidl, C. Cali, E. Gröller, H. Pfister, M. Hadwiger, and J. Beyer, “Barrio: Customizable Spatial Neighborhood Analysis and Comparison for Nanoscale Brain Structures,” *Computer Graphics Forum*, vol. 41, no. 3, pp. 183–194, 2022.

[70] F. Cheng, M. S. Keller, H. Qu, N. Gehlenborg, and Q. Wang, “Polyphony: An Interactive Transfer Learning Framework for Single-Cell Data Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 591–601, 2023.

[71] K. Dinkla, H. Strobelt, B. Genest, S. Reiling, M. Borowsky, and H. Pfister, “Screenit: Visual Analysis of Cellular Screens,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 591–600, 2017.

[72] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, “CellProfiler: Image analysis software for identifying and quantifying cell phenotypes,” *Genome Biology*, vol. 7, no. 10, p. R100, 2006.

[73] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter, “CellProfiler Analyst: Data exploration and analysis software for complex image-based screens,” *BMC Bioinformatics*, vol. 9, no. 1, p. 482, 2008.

[74] L. Marconato, G. Palla, K. A. Yamauchi, I. Virshup, E. Heidari, T. Treis, W.-M. Vierdag, M. Toth, S. Stockhaus, R. B. Shrestha, B. Rombaut, L. Pollaris, L. Lehner, H. Vöhringer, I. Kats, Y. Saeys, S. K. Saka, W. Huber, M. Gerstung, J. Moore, F. J. Theis, and O. Stegle, “SpatialData: An open and universal data framework for spatial omics,” *Nature Methods*, pp. 1–5, 2024.

[75] M. S. Keller, I. Gold, C. McCallum, T. Manz, P. V. Kharchenko, and N. Gehlenborg, "Vitessce: Integrative visualization of multimodal and spatially-resolved single-cell data," *OSF Preprints*, 2021.

[76] W. De Leeuw, R. Van Liere, P. Verschure, A. Visser, E. Manders, and R. Van Drielf, "Visualization of time dependent confocal microscopy data," in *Proceedings Visualization (VIS)*, pp. 473–476, 2000.

[77] A. Merouane, N. Rey-Villamizar, Y. Lu, I. Liadi, G. Romain, J. Lu, H. Singh, L. J. Cooper, N. Varadarajan, and B. Roysam, "Automated profiling of individual cell–cell interactions from high-throughput time-lapse imaging microscopy in nanowell grids (TIMING)," *Bioinformatics*, vol. 31, no. 19, pp. 3189–3197, 2015.

[78] G. Hattab, V. Wiesmann, A. Becker, T. Munzner, and T. W. Nattkemper, "A Novel Methodology for Characterizing Cell Subpopulations in Automated Time-lapse Microscopy," *Frontiers in Bioengineering and Biotechnology*, vol. 6, 2018.

[79] G. Anselmi and M. Humphry, "Livecyte: Creating a comprehensive cell profile," *Microscopie*, vol. 30, no. 2, 2019.

[80] M. Sebesta, P. J. Egelberg, A. Langberg, J.-H. Lindskov, K. Alm, and B. Janicke, "HoloMonitor M4: Holographic imaging cytometer for real-time kinetic label-free live-cell analysis of adherent cells," in *Quantitative Phase Imaging II*, vol. 9718, p. 971813, International Society for Optics and Photonics, 2016.

[81] T. L. Nguyen, S. Pradeep, R. L. Judson-Torres, J. Reed, M. A. Teitell, and T. A. Zangle, "Quantitative Phase Imaging: Recent Advances and Expanding Potential in Biomedicine," *ACS nano*, vol. 16, pp. 11516–11544, Aug. 2022.

[82] A. J. Pretorius, I. A. Khan, and R. J. Errington, "A Survey of Visualization for Live Cell Imaging," *Computer Graphics Forum*, vol. 36, no. 1, pp. 46–63, 2017.

[83] S. H. Choi, E. J. Ku, Y. A. Choi, and J. W. Oh, "Grave-to-cradle: Human embryonic lineage tracing from the postmortem body," *Experimental & Molecular Medicine*, vol. 55, no. 1, pp. 13–21, 2023.

[84] J. S. Packer, Q. Zhu, C. Huynh, P. Sivaramakrishnan, E. Preston, H. Dueck, D. Stefanik, K. Tan, C. Trapnell, J. Kim, R. H. Waterston, and J. I. Murray, "A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution," *Science*, vol. 365, no. 6459, p. eaax1971, 2019.

[85] M. Meyer, T. Munzner, A. DePace, and H. Pfister, "MulteeSum: A Tool for Comparative Spatial and Temporal Gene Expression Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 908–917, 2010.

[86] A. Cedilnik, J. Baumes, L. Ibanez, S. Megason, and B. Wylie, "Integration of information and volume visualization for analysis of cell lineage and gene expression during embryogenesis," in *Visualization and Data Analysis 2008*, vol. 6809, pp. 193–203, SPIE, 2008.

[87] I. Salvador-Martínez, M. Grillo, M. Averof, and M. J. Telford, "CeLaVi: An interactive cell lineage visualization tool," *Nucleic Acids Research*, vol. 49, no. W1, pp. W80–W85, 2021.

[88] J. Hong, A. Trubuil, and T. Isenberg, “LineageD: An Interactive Visual System for Plant Cell Lineage Assignments based on Correctable Machine Learning,” *Computer Graphics Forum*, vol. 41, no. 3, pp. 195–207, 2022.

[89] A. J. Pretorius, I. A. Khan, and R. J. Errington, “Cell lineage visualisation,” *Computer Graphics Forum*, vol. 34, no. 3, pp. 21–30, 2015.

[90] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, “Characterizing Guidance in Visual Analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 111–120, 2017.

[91] C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer, “Guidance in the human–machine analytics process,” *Visual Informatics*, vol. 2, pp. 166–180, Sept. 2018.

[92] D. Ceneda, T. Gschwandtner, and S. Miksch, “A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective,” *Computer Graphics Forum*, vol. 38, no. 3, pp. 861–879, 2019.

[93] F. Sperrle, D. Ceneda, and M. El-Assady, “Lotse: A Practical Framework for Guidance in Visual Analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, pp. 1124–1134, Jan. 2023.

[94] D. Ceneda, C. Collins, M. El-Assady, S. Miksch, C. Tominski, and A. Arleo, “A Heuristic Approach for Dual Expert/End-User Evaluation of Guidance in Visual Analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, pp. 997–1007, Jan. 2024.

[95] E. Kerzner, A. Lex, C. L. Sigulinsky, T. Urness, B. W. Jones, R. E. Marc, and M. Meyer, “Graffinity: Visualizing Connectivity in Large Graphs,” *Computer Graphics Forum (EuroVis)*, vol. 36, no. 3, pp. 251–260, 2017.

[96] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski, “Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks,” *Computer Graphics Forum*, vol. 34, pp. 31–40, June 2015.

[97] F. Lekschas, B. Bach, P. Kerpedjiev, N. Gehlenborg, and H. Pfister, “HiPiler: Visual Exploration of Large Genome Interaction Matrices with Interactive Small Multiples,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 522–531, 2018.

[98] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, “Activation Atlas,” *Distill*, vol. 4, p. e15, Mar. 2019.

[99] A. A. Friedman, A. Letai, D. E. Fisher, and K. T. Flaherty, “Precision medicine for cancer with next-generation functional diagnostics,” *Nature Reviews Cancer*, vol. 15, pp. 747–756, Dec. 2015.

[100] T. A. Zangle and M. A. Teitell, “Live-cell mass profiling: An emerging approach in quantitative biophysics,” *Nature Methods*, vol. 11, pp. 1221–1228, Dec. 2014.

[101] J. Chun, T. A. Zangle, T. Kolarova, R. S. Finn, M. A. Teitell, and J. Reed, "Rapidly quantifying drug sensitivity of dispersed and clumped breast cancer cells by mass profiling," *Analyst*, vol. 137, no. 23, pp. 5495–5498, 2012.

[102] H. Zeng, A. Jorapur, A. H. Shain, U. E. Lang, R. Torres, Y. Zhang, A. S. McNeal, T. Botton, J. Lin, M. Donne, I. N. Bastian, R. Yu, J. P. North, L. Pincus, B. S. Ruben, N. M. Joseph, I. Yeh, B. C. Bastian, and R. L. Judson, "Bi-allelic Loss of CDKN2A Initiates Melanoma Invasion via BRN2 Activation," *Cancer Cell*, vol. 34, pp. 56–68.e9, July 2018.

[103] M. Hejna, A. Jorapur, Y. Zhang, J. S. Song, and R. L. Judson, "Quantification of mammalian tumor cell state plasticity with digital holographic cytometry," in *Quantitative Phase Imaging IV*, vol. 10503, p. 1050312, International Society for Optics and Photonics, Feb. 2018.

[104] M. Sedlmair, M. Meyer, and T. Munzner, "Design Study Methodology: Reflections from the Trenches and the Stacks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[105] K. Holtzblatt and H. Beyer, "Contextual Design: Evolved," *Synthesis Lectures on Human-Centered Informatics*, vol. 7, pp. 1–91, Oct. 2014.

[106] E. Kerzner, S. Goodwin, J. Dykes, S. Jones, and M. Meyer, "A Framework for Creative Visualization-Opportunities Workshops," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 748–758, Jan. 2019.

[107] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister, "Pathline: A Tool For Comparative Functional Genomics," *Computer Graphics Forum (EuroVis '10)*, vol. 29, no. 3, pp. 1043–1052, 2010.

[108] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[109] B. Brown, S. Reeves, and S. Sherwood, "Into the wild: Challenges and opportunities for field trial methods," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York), pp. 1657–1666, ACM, May 2011.

[110] P. A. Ascierto, L. Borgognoni, G. Botti, M. Guida, P. Marchetti, S. Mocellin, P. Muto, G. Palmieri, R. Patuzzo, P. Quaglino, I. Stanganelli, and C. Caracò, "New paradigm for stage III melanoma: From surgery to adjuvant treatment," *Journal of Translational Medicine*, vol. 17, p. 266, Aug. 2019.

[111] R. Lopez-Hernandez, D. Guilmaine, M. J. McGuffin, and L. Barford, "A layer-oriented interface for visualizing time-series data from oscilloscopes," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 41–48, Mar. 2010.

[112] T. Manz, I. Gold, N. H. Patterson, C. McCallum, M. S. Keller, B. W. H. Ii, K. Börner, J. M. Spraggins, and N. Gehlenborg, "Viv: Multiscale Visualization of High-Resolution Multiplexed Bioimaging Data on the Web," 2020.

[113] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 2713–2722, Dec. 2013.

[114] A. Cutler and L. Breiman, "Archetypal Analysis," *Technometrics*, vol. 36, pp. 338–347, Nov. 1994.

[115] C. Nobre, D. Wootton, L. Harrison, and A. Lex, "Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, ACM, 2020.

[116] H.-J. Schulz, "Treevis.net: A Tree Visualization Reference," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 11–15, 2011.

[117] H.-J. Schulz, S. Hadlak, and H. Schumann, "The Design Space of Implicit Hierarchy Visualization: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 4, pp. 393–411, 2011.

[118] T. Harbig, M. W. Paz, and K. Nieselt, "GO-Compass: Visual Navigation of Multiple Lists of GO terms," *Computer Graphics Forum*, vol. 42, no. 3, pp. 271–281, 2023.

[119] C. Han, J. Jo, A. Li, B. Lee, O. Deussen, and Y. Wang, "SizePairs: Achieving Stable and Balanced Temporal Treemaps using Hierarchical Size-based Pairing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 193–202, 2023.

[120] C. Nobre, M. Meyer, M. Streit, and A. Lex, "The State of the Art in Visualizing Multivariate Networks," *Computer Graphics Forum (EuroVis)*, vol. 38, no. 3, pp. 807–832, 2019.

[121] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer, "Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1693–1702, 2014.

[122] C. Nobre, N. Gehlenborg, H. Coon, and A. Lex, "Lineage: Visualizing Multivariate Clinical Data in Genealogy Graphs," *Transaction on Visualization and Computer Graphics*, vol. 25, no. 3, pp. 1543–1558, 2019.

[123] C. Nobre, M. Streit, and A. Lex, "Juniper: A Tree+Table Approach to Multivariate Graph Visualization," *Transaction on Visualization and Computer Graphics (InfoVis)*, vol. 25, no. 1, pp. 544–554, 2019.

[124] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.

[125] J. Seo and B. Shneiderman, "Interactively exploring hierarchical clustering results [gene identification]," *Computer*, vol. 35, no. 7, pp. 80–86, 2002.

[126] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg, "Comparative Analysis of Multidimensional, Quantitative Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1027–1035, 2010.

[127] B. Lee, L. Nachmanson, G. Robertson, J. M. Carlson, and D. Heckerman, "PhyloDet: A scalable visualization tool for mapping multiple traits to large evolutionary trees," *Bioinformatics*, vol. 25, no. 19, pp. 2611–2612, 2009.

[128] B. R. Briggs, J. W. Pohlman, M. Torres, M. Riedel, E. L. Brodie, and F. S. Colwell, "Macroscopic Biofilms in Fracture-Dominated Sediment That Anaerobically Oxidize Methane," *Applied and Environmental Microbiology*, vol. 77, no. 19, pp. 6780–6787, 2011.

[129] Ł. Kreft, A. Botzki, F. Coppens, K. Vandepoele, and M. Van Bel, "PhyD3: A phylogenetic tree viewer with extended phyloXML support for functional genomics data visualization," *Bioinformatics*, vol. 33, no. 18, pp. 2946–2947, 2017.

[130] S. K. Card, B. Suh, B. A. Pendleton, J. Heer, and J. W. Bodnar, "Time Tree: Exploring Time Changing Hierarchies," in *2006 IEEE Symposium On Visual Analytics Science And Technology*, pp. 3–10, 2006.

[131] T. Tekusova and T. Schreck, "Visualizing Time-Dependent Data in Multivariate Hierarchic Plots - Design and Evaluation of an Economic Application," in *12th International Conference Information Visualisation*, pp. 143–150, 2008.

[132] W. Köpp and T. Weinkauf, "Temporal Treemaps: Static Visualization of Evolving Trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 534–543, 2019.

[133] J. B. Kruskal and J. M. Landwehr, "Icicle Plots: Better Displays for Hierarchical Clustering," *The American Statistician*, vol. 37, no. 2, p. 162, 1983.

[134] T. Schreck, D. Keim, and F. Mansmann, "Regular TreeMap layouts for visual analysis of hierarchical data," in *Proceedings of the 22nd Spring Conference on Computer Graphics*, SCCG '06, (New York), pp. 183–190, ACM, 2006.

[135] M. Burch, F. Beck, and S. Diehl, "Timeline trees: Visualizing sequences of transactions in information hierarchies," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI, (New York), pp. 75–82, ACM, 2008.

[136] M. Burch and D. Weiskopf, "VISUALIZING DYNAMIC QUANTITATIVE DATA IN HIERARCHIES - TimeEdgeTrees: Attaching Dynamic Weights to Tree Edges," in *International Conference on Information Visualization Theory and Applications*, pp. 177–186, 2023.

[137] C. Nobre and A. Lex, "OceanPaths: Visualizing Multivariate Oceanography Data," *Computer Graphics Forum*, 2015.

[138] F. Lekschas, M. Behrisch, B. Bach, P. Kerpedjiev, N. Gehlenborg, and H. Pfister, "Pattern-Driven Navigation in 2D Multiscale Visualizations with Scalable Insets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 611–621, 2020.

[139] S. Ghani, N. H. Riche, and N. Elmqvist, "Dynamic Insets for Context-Aware Graph Navigation," *Computer Graphics Forum*, vol. 30, no. 3, pp. 861–870, 2011.

[140] D. Coffey, Chi-Lun Lin, A. G. Erdman, and D. F. Keefe, "Design by Dragging: An Interface for Creative Forward and Inverse Design with Simulation Ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2783–2791, 2013.

[141] F. Lekschas, X. Zhou, W. Chen, N. Gehlenborg, B. Bach, and H. Pfister, "A Generic Framework and Library for Exploration of Small Multiples through Interactive Piling," *IEEE Transactions on Visualization and Computer G.*, vol. 27, no. 2, pp. 358–368, 2021.

[142] H. Luo, J. Fan, J. Yang, W. Ribarsky, and S. Satoh, "Exploring Large-Scale Video News via Interactive Visualization," in *2006 IEEE Symposium On Visual Analytics Science And Technology*, pp. 75–82, 2006.

[143] Y. Yang, F. Dadgostar, C. Sanderson, and B. C. Lovell, "Summarisation of surveillance videos by key-frame selection," in *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, (Ghent, Belgium), pp. 1–6, IEEE, 2011.

[144] Y. Zhang, R. L. Belote, M. A. Urquijo, M. M. K. Hansen, M. Hejna, T. E. Moustafa, T. Liu, D. Lange, F. Vand-Rajabpour, M. Chang, B. K. Lohman, C. Stubben, X. Zhang, L. S. Weinberger, M. W. VanBrocklin, D. Grossman, A. Lex, R. Kulkarni, T. Zangle, and R. L. Judson-Torres, "Bidirectional interconversion between mutually exclusive tumorigenic and drug-tolerant melanoma cell phenotypes," 2023.

[145] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *Proceedings of the IEEE Pacific Visualization Symposium*, pp. 1–8, IEEE, 2012.

[146] T. Saito, H. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda, "Two-tone pseudo coloring: Compact visualization for one-dimensional data," in *IEEE Symposium on Information Visualization (InfoVis)*, (Minneapolis), pp. 173–180, IEEE, 2005.

[147] T. Munzner, "Rules of Thumb," in *Visualization Analysis and Design*, 2014.

[148] Z. T. Cutler, K. Gadhave, and A. Lex, "Trrack: A Library for Provenance Tracking in Web-Based Visualizations," in *IEEE Visualization Conference (VIS)*, pp. 116–120, 2020.

[149] T. Manz, I. Gold, N. H. Patterson, C. McCallum, M. S. Keller, B. W. Herr, K. Börner, J. M. Spraggins, and N. Gehlenborg, "Viv: Multiscale visualization of high-resolution multiplexed bioimaging data on the web," *Nature Methods*, vol. 19, no. 5, pp. 515–516, 2022.

[150] A. Van Der Ploeg, "Drawing non-layered tidy trees in linear time: DRAWING NON-LAYERED TIDY TREES IN LINEAR TIME," *Software: Practice and Experience*, vol. 44, no. 12, pp. 1467–1484, 2014.

[151] M. Meyer and J. Dykes, "Criteria for Rigor in Visualization Design Study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, pp. 87–97, Jan. 2020.

[152] A. S. McNeal, R. L. Belote, H. Zeng, M. Urquijo, K. Barker, R. Torres, M. Curtin, A. H. Shain, R. H. Andtbacka, S. Holmen, D. H. Lum, T. H. McCalmont, M. W. VanBrocklin, D. Grossman, M. L. Wei, U. E. Lang, and R. L. Judson-Torres, "BRAFV600E induces reversible mitotic arrest in human melanocytes via microRNA-mediated suppression of AURKB," *eLife*, vol. 10, p. e70385, 2021.

[153] J. Zhang, J. Griffin, K. Roy, A. Hoffmann, and T. A. Zangle, "Tracking of Lineage Mass via Quantitative Phase Imaging and Confinement in Low Refractive Index Microwells," 2024.

[154] C. Piller, “Potential fabrication in research images threatens key theory of Alzheimer’s disease,” *Science*, vol. 377, no. 6604, pp. 358–363, 2022.

[155] G. Viglione, “‘Avalanche’ of spider-paper retractions shakes behavioural-ecology community,” *Nature*, vol. 578, pp. 199–200, Feb. 2020.

[156] S. Haroz, “Open Practices in Visualization Research : Opinion Paper,” in *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 46–52, Oct. 2018.

[157] A. I. Oransky, “Why write a blog about retractions?,” Aug. 2010.

[158] The Center for Scientific Integrity, “The Retraction Watch Database,” 2018.

[159] J. Editors, “Retraction Note to: Why money meanings matter in decisions to donate time and money,” *Marketing Letters*, vol. 27, pp. 409–409, June 2016.

[160] A. Charlton, “RETRACTED ARTICLE: Why money meanings matter in decisions to donate time and money.” https://openmkt.org/blog/2021/retracted-article-why-money-meanings-matter-in-decisions-to-donate-time-and-money/, July 2021.

[161] H. Pashler, D. Rohrer, I. Abramson, T. Wolfson, and C. R. Harris, “A Social Priming Data Set With Troubling Oddities,” *Basic and Applied Social Psychology*, vol. 38, pp. 3–18, Jan. 2016.

[162] J. Editors, “Retraction for Shu et al., Signing at the beginning makes ethics salient and decreases dishonest self-reports in comparison to signing at the end,” *Proceedings of the National Academy of Sciences*, vol. 118, p. e2115397118, Sept. 2021.

[163] U. Simonsohn, J. Simmons, and L. Nelson, “[98] Evidence of Fraud in an Influential Field Experiment About Dishonesty.” https://datacolada.org/98, Aug. 2021.

[164] J. Lawrence, “Why Was a Major Study on Ivermectin for COVID-19 Just Retracted?,” July 2021.

[165] N. Brown, “Some problems in the dataset of a large study of Ivermectin for the treatment of Covid-19,” July 2021.

[166] F. A. E. Sosso, D. J. Kuss, C. Vandelanotte, J. L. Jasso-Medrano, M. E. Husain, G. Curcio, D. Papadopoulos, A. Aseem, P. Bhati, F. Lopez-Rosales, J. R. Becerra, G. D’Aurizio, H. Mansouri, T. Khoury, M. Campbell, and A. J. Toth, “Retraction Note: Insomnia, sleepiness, anxiety and depression among different types of gamers in African countries,” *Scientific Reports*, vol. 10, p. 9256, June 2020.

[167] N. Brown, “Some issues in a recent gaming research article: Etindele Sosso et al. (2020),” Apr. 2020.

[168] K. L. Laskowski, A. P. Modlmeier, A. E. DeMarco, A. Coleman, K. Zhao, H. A. Brittingham, D. R. McDermott, and J. N. Pruitt, “Retraction: Persistent social interactions beget more pronounced personalities in a desert-dwelling social spider,” *Biology Letters*, vol. 16, p. 20200062, Feb. 2020.

[169] K. L. Laskowski, “What to do when you don’t trust your data anymore – Laskowski Lab at UC Davis,” Jan. 2020.

[170] K. L. Laskowski and J. N. Pruitt, “Retraction: Evidence of social niche construction: Persistent and repeated social interactions generate stronger personalities in a social spider,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 287, p. 20200077, Jan. 2020.

[171] K. L. Laskowski, P.-O. Montiglio, and J. N. Pruitt, “Retraction: Individual and Group Performance Suffers from Social Niche Disruption,” *The American Naturalist*, vol. 195, pp. 393–393, Feb. 2020.

[172] S. Wang, “Retraction: Glioma Gene Therapy Using Induced Pluripotent Stem Cell Derived Neural Stem Cells,” *Molecular Pharmaceutics*, vol. 16, p. 4088, Sept. 2019.

[173] P. B. Editorial Board, “Editor's Note on: Host–parasitoid evolution in a metacommunity,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 288, p. 20210505, Mar. 2021.

[174] A. Author, “PubPeer discussion of ”Host-parasitoid evolution in a metacommunity”,” Aug. 2020.

[175] H. H. Thorp, “Editorial Retraction,” *Science*, vol. 377, pp. 826–826, Aug. 2022.

[176] M. Enserink, “Sea of doubts,” *Science*, vol. 372, pp. 560–565, May 2021.

[177] J. N. Towse, T. Loetscher, and P. Brugger, “Not all numbers are equal: Preferences and biases among children and adults when generating random sequences,” *Frontiers in Psychology*, vol. 5, p. 19, Jan. 2014.

[178] M.-A. Schulz, B. Schmalbach, P. Brugger, and K. Witt, “Analysing Humanly Generated Random Number Sequences: A Pattern-Based Approach,” *PLOS ONE*, vol. 7, p. e41531, July 2012.

[179] M. Figurska, M. Stańczyk, and K. Kulesza, “Humans cannot consciously generate random numbers sequences: Polemic study,” *Medical Hypotheses*, vol. 70, pp. 182–185, Jan. 2008.

[180] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-Lite: A Grammar of Interactive Graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2017.

[181] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, “LineUp: Visual Analysis of Multi-Attribute Rankings,” *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, vol. 19, no. 12, pp. 2277–2286, 2013.

[182] P. T. Inc., “Collaborative data science.” Plotly Technologies Inc., 2015.

[183] T. excel.js developers, “ExcelJS.” exceljs, Dec. 2022.

[184] C. Perin, P. Dragicevic, and J.-D. Fekete, “Revisiting Bertin Matrices: New Interactions for Crafting Tabular Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 2082–2091, Dec. 2014.

[185] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, "Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 438–448, Jan. 2019.

[186] S. Palan and C. Schitter, "Prolific.ac A subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, Mar. 2018.

[187] Y. Ding, J. Wilburn, H. Shrestha, A. Ndlovu, K. Gadhave, C. Nobre, A. Lex, and L. Harrison, "reVISit: Supporting Scalable Evaluation of Interactive Visualizations," in *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 31–35, Oct. 2023.

[188] T. Gschwandtner, W. Aigner, S. Miksch, J. Gärtner, S. Kriglstein, M. Pohl, and N. Suchy, "TimeCleanser: A visual analytics approach for data cleansing of time-oriented data," in *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*, I-KNOW '14, (New York, NY, USA), pp. 1–8, Association for Computing Machinery, Sept. 2014.

[189] T. Gschwandtner and O. Erhart, "Know Your Enemy: Identifying Quality Problems of Time Series Data," in *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 205–214, IEEE Computer Society, Apr. 2018.

[190] A. Bäuerle, H. Neumann, and T. Ropinski, "Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks," *Computer Graphics Forum*, vol. 39, no. 3, pp. 195–205, 2020.

[191] M. B. Kery, D. Ren, F. Hohman, D. Moritz, K. Wongsuphasawat, and K. Patel, "Mage: Fluid Moves Between Code and Graphical Work in Computational Notebooks," *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pp. 140–151, Oct. 2020.

[192] K. Gadhave, Z. Cutler, and A. Lex, "Persist: Persistent and reusable interactions in computational notebooks," *Computer Graphics Forum (EuroVis)*, 2024.

[193] Y. Wu, J. M. Hellerstein, and A. Satyanarayan, "B2: Bridging Code and Interactive Visualization in Computational Notebooks," in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, (New York, NY, USA), pp. 152–165, Association for Computing Machinery, Oct. 2020.

[194] J. R. Conway, A. Lex, and N. Gehlenborg, "UpSetR: An R package for the visualization of intersecting sets and their properties," *Bioinformatics*, vol. 33, pp. 2938–2940, Sept. 2017.