Accelerating Scientific Data Exploration via Visual Query Systems

Doris Jung-Lin Lee, John Lee, Tarique Siddiqui,
Jaewoo Kim, Karrie Karahalios, Aditya Parameswaran

Abstract—The increasing availability of rich and complex data in a variety of scientific domains poses a pressing need for tools to enable scientists to rapidly make sense of and gather insights from data. One proposed solution is to design visual query systems (VQSs) that allow scientists to interactively search for desired patterns in their datasets. While many existing VQSs promise to accelerate exploratory data analysis by facilitating this search, they are unfortunately not widely used in practice. Through a year-long collaboration with scientists in three distinct domains—astronomy, genetics, and material science—we study the impact of various features within VQSs that can aid rapid visual data analysis, and how VQSs fit into a scientists' analysis workflow. Our findings offer design guidelines for improving the usability and adoption of next-generation VQSs, paving the way for VQSs to be applied to a variety of scientific domains.

Index Terms—Data visualization, exploratory data analysis, visual query, scientific data.

1 Introduction

From high-throughput genome sequencing, to multi-resolution astronomical imaging telescopes, to at-scale physical testing of battery candidates, many fields of science and engineering are facing an increasing availability of large volumes of complex data [6, 13], holding the key to some of the most pressing unanswered scientific questions of our time, such as: How does a treatment affect the expression of a gene in a breast cancer cell-line? Which battery components have sustainable levels of energy-efficiency and are safe and cheap to manufacture in production? While data analysis is central to a scientist's knowledge discovery process, scientists often lack the extensive experience to deal with data of this scale and complexity in a way that can facilitate rapid insight discovery [23].

To explore their data, many scientists currently create visualizations, either programmatically using tools (such as `ggplot` or `matplotlib`), or visualization construction interfaces (such as Excel or Tableau) [31, 41]. In either case, scientists are required to specify exactly what they want to visualize. From early discussions with analysts from twelve different application areas, we learned that analysts often need to search for some desired pattern or trend amongst large numbers of visualizations. For example, when trying to find celestial objects corresponding to supernovae, which have a specific pattern of brightness over time, scientists need to individually inspect the visualizations of each object (often numbering in the thousands) until they find ones that match the pattern. Similarly, when trying to infer relationships between two physical properties for different subsets of battery electrolytes, scientists need to individually visualize these properties for each subset (out of an unbounded number of such subsets) until they identify relationships that make sense to them. This process of manually exploring a large number of visualizations is not only error-prone, but also overwhelming for scientists who do not have extensive knowledge about their dataset.

One potential solution for this challenge of manually exploring a large collection of visualizations are systems that allow users to specify desired visual patterns, via a high-level specification language or interface, with the system automatically traversing all potential visualization candidates to find and return those that match the specification. We define such systems to be *Visual Query Systems*, or VQSs for short. There are a number of VQSs that have been introduced in the literature [20, 30, 42, 48, 50]. For example, Google Correlate [30] and QuerySketch [50] allow users to draw a trend of interest, with the system automating the search to find matching visualizations.

Not surprisingly, when asked about the potential viability of VQSs in aiding scientific data analysis, many scientists indicated that VQSs could be useful in mitigating the challenge of visualization exploration described earlier—since it automates the painful manual exploration of visualizations to find desired patterns. Yet, to the best of our knowledge, VQSs are not very commonly used in practice. *Our paper seeks to bridge this gap between current research on VQSs to understand why existing VQSs are not commonly used by analysts and how they can actually be used in practice, as a first step towards the broad adoption of VQSs in data analysis*. In this paper, we present findings from a series of interviews, cognitive walkthroughs, participatory design, and user studies with scientists from three different scientific domains—*astronomy, genetics,* and *material science*—through a year-long collaboration. These scientific use cases present a diverse set of goals and datasets that could benefit from a VQS. Our three main research questions are as follows:

RQ1: What are the challenges in existing scientific data analysis workflows that could be potentially addressed by a VQS?

Via cognitive walkthroughs and interviews, we gained an understanding of the data analysis workflows presently employed by the scientists, their needs, and the challenges they face. We identified opportunities where a VQS could help accelerate their analysis, by helping them discover insights, gain intuition, or provoke directions for exploration. Finally, we determined the types of research questions and dataset properties that would be most suitable for exploration on VQSs.

RQ2: What types of interface capabilities are necessary to develop VQSs into a useful component of data analysis?

Via participatory design, we distilled a set of key features that our scientist participants would need for VQSs to be useful and usable within their data analysis workflows. Based on our early interactions with scientists, we started to build a VQS [47, 48] that, similar to existing VQSs [50], allowed them to search for desired trends via drawing on a canvas. This early system served as a functional prototype for us to engage with scientists further in the participatory design process, understand how they envision themselves using a VQS, and gather feedback on feature designs that could make the VQS more useful. The features we developed address challenges shared across the three scientific domains, ranging from additional querying modalities, to features that support a more integrated workflow, to improving the interpretability of the system output, most of them missing in prior VQSs in the literature. Our collaborative design experience culminated in a full-fledged VQS, *zenvisage*, capable of facilitating rapid hypothesis

• *All authors are with University of Illinois, Urbana-Champaign. E-mail: jlee782, lee98, tsiddiq2, jkim475, kkarhal, adityagp@illinois.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

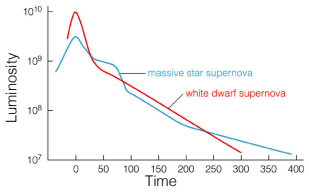
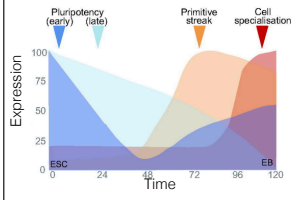
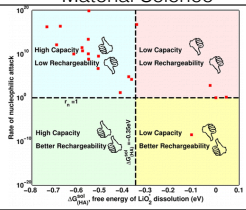
	Astronomy	Genetics	Material Science
Desired Pattern			
User	Data Engineers	Model Builders	Data Collectors
Challenge	Discovering ***rare*** astronomical objects in ***large, noisy, non-uniform*** dataset while making use of large numbers of additional attributes to support inference.	Clustering ***large*** numbers of genes into ***multiple characteristic profiles*** *that* can rise and peak at ***different time points***.	Identifying solvents from a ***large*** multidimensional dataset by ***comparing the relationships*** between ***multiple attributes*** *while considering* ***tradeoffs*** *between desirable characteristics.*
Problem	Find small number of visualizations for fixed x,y with **known** patterns	Find groups of visualizations for fixed x,y with **unknown** patterns	Find relationships between variable x,y with **unknown** patterns

Figure 1: Descriptions of the three scientific use cases discussed in this paper.

generation and insight discovery.

RQ3: How do VQSs accelerate scientific insights? and *RQ4: How can VQSs fit within the context of existing data analysis workflows?*

To evaluate our final system *zenvisage*, we conducted a user study with nine scientists (including those who had participated in the design process), all of whom had a vested interest in using a VQS to address their research questions on their datasets. In a 1.5-hour user study, our scientist participants were able to gain novel scientific insights, such as *identifying a star with a transient pattern that was known to harbor a Jupiter-sized planet, finding characteristic gene expression profiles that confirmed the results of a related publication, and learning that the dip in an astronomical light curve is caused by saturated imaging equipment overlooked by the existing error-detection pipeline*. Participants also gained additional insights about their datasets, including debugging mislabelled features and uncovering the erroneous data preprocessing procedure applied to a collaborator's dataset. We learned how VQSs could be contextualized within scientific data analysis workflows and discovered that VQSs can be used beyond the exploratory phase of analysis, for data verification, debugging preliminary datasets, and performing sanity-checks on downstream models.

As most of the existing VQSs are evaluated in a standalone fashion via artificial tasks and datasets, to the best of our knowledge, *our study is the first to holistically examine how VQSs can be used in practice and integrated into existing data analysis workflows*. From these experiences, we advocate common design guidelines and end-user considerations when building the next generation VQSs.

2 Visual Query Systems: Definition and Brief Survey

Visual analytics systems, such as Tableau [3], support powerful visualization construction interfaces that enable users to specify their desired visual encoding and data subset for generating a visualization. However, during data exploration, users might only have a vague, high-level idea of what they want to address, rather than specific instances of what they want to visualize.

To address this issue, recent studies have explored the use of visualization recommendations to accelerate data exploration. The techniques used include using statistical and perceptual measures [25, 51, 52], past user history [17], and visualizations that look "different" from the rest [49].

Instead of providing generic visualization recommendations, VQSs are a special class of visualization systems that enable users to more directly search for visualizations through an intuitive interface. We elaborate on our description of VQSs in the introduction and define VQSs as *systems that allow users to specify the desired pattern via some high-level specification language or interface, with the system returning recommendations of visualizations that match the specified pattern.* One instantiation of VQSs are sketch-to-query interfaces that allows users to sketch the desired "shape" of a visualization with the system returning visualizations that look similar [1, 30, 42, 50]. Other work has explored the types of shape features a user may be interested in for issuing more specific queries [11]. Additionally, some VQSs support specifying patterns through boxed constraints for range-queries [20], regular expressions [54], and natural language [15, 43]. While these systems have shown to be effective for dynamic querying in controlled lab studies, they have not been evaluated in-situ on real-world use cases. In this work, we additionally investigate how VQSs complement other common interactions in visual data analysis, and scientists' existing workflows.

3 Methods

In this section, we will first provide a brief overview of existing evaluation methodologies for visualization and visual analytics systems, then describe our chosen research methodology for this paper.

3.1 Evaluating Visual Analytics Systems

Visualization systems are often evaluated using controlled studies that measure the user's performance against an existing visualization baseline [39]. Cognitive measures such as insight time [35, 53] have been developed to capture how well users perform on a task against existing baselines. However, since the operations, hypotheses generated, and insights obtained through exploratory analysis are variable and subjective with respect to individual users and their analytic goals, it is impossible to define tasks beforehand and compare across control groups. Techniques such as artificially inserting "insights" or setting predefined tasks for example datasets work well for objective tasks, such as debugging data errors [21, 36], but these contrived methods are unsuitable for trying to learn about the types of real-world queries users may want to pose on VQSs. In order to make the user study more realistic, we opted for a qualitative evaluation where we allowed participants to bring datasets that they have vested interests in to address unanswered research questions.

Due to the unrealistic nature of controlled studies, many have proposed using a more multi-faceted, ethnographic approach to understand how analysts perform visual data analysis and reasoning [26, 28, 33, 39, 46]. For example, multi-dimensional, in-depth, long-term case studies (MILCs) advocate the use of interviews, surveys, logging and other empirical artifacts to create a holistic understanding of how a visualization system is used in its intended environment [46]. Some papers have explored designing visualization and collaborative tools for scientific workflows through individual case studies, e.g., [8, 40]. Similarly, in our work, real-world case studies help us situate how VQSs could be used in the context of an existing analysis workflow.

We adopt *participatory design* practices in this work: participatory design "allows potential users to participate in the design of a system that they will ultimately use" [18, 32]. Participatory design has been successfully used in the development of interactive visualization systems in the past [5, 9]. Sedlmair et al. [28] highlights the benefits and

pitfalls of design studies in visualization research. They advocate that design study methodology is suitable for use cases in which the data is available for prototyping, but the task is only partially known and the information is partially in the user's head. In that regard, our scientific use cases with VQS is well-suited for a design study methodology, as we learn about the scientist's data and analysis requirements and design interactions that helps users translate their "in-the-head" specifications into actionable visual queries.

3.2 Our Approach: Multiple Design Studies with Initial Prototype and Realistic Evaluation

We adopted a mixed methods research methodology that draws inspiration from ethnographic methods, iterative and participatory design, and controlled studies [29, 32, 46] to understand how VQSs can accelerate scientific data analysis. Our methodology was designed to address the research questions outlined in the introduction. Working with researchers from three different scientific research groups, we identified the needs and challenges of scientific data analysis and the potential opportunities for VQSs to fit in, via interviews and cognitive walkthroughs *(RQ1)*.

Given our early conversations with the participants, we built a basic VQS to serve as the functional prototype in the design study. As shown in Figure 2, this early version of *zenvisage* allowed users to sketch a pattern or drag-and-drop an existing visualization as a query, then the system would return visualizations that had the closest Euclidean distance from the queried pattern. *zenvisage* also displayed representative and outlier patterns to help provide an overview of typical trends. In addition, *zenvisage* supports a sophisticated query language, ZQL, that we did not employ for this study. The details of the system is described in our previous work [47, 48], which focused on the systems and scalability aspects of the VQSs.

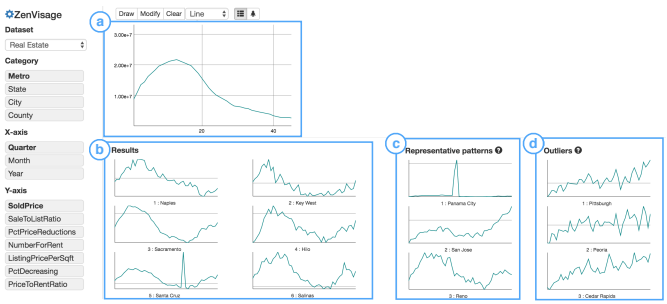


Figure 2: The *zenvisage* prototype allowed users to sketch a pattern in (a), which would then return (b) results that had the closest Euclidean distance from the sketched pattern. The system also displays (c) representative patterns obtained through K-Means clustering and (d) outlier patterns to help the users gain an overview of the dataset.

The use of functional prototypes is common in participatory design to provide a starting point for the participants. For example, Ciolfi et al. [10] studied two different alternatives to co-design (starting with open brief versus functional prototype) in the development of museum guidance systems and found that while both approaches were equally fruitful, functional prototypes can make addressing a specific challenge more immediate and focused (in our case the challenge is making comparison across large numbers of visualizations as we found through informal discussions with practitioners). Our motivation for providing a functional prototype at the beginning of the participatory design sessions is to showcase capabilities of VQSs. Especially since VQSs are not common in the existing workflows of these scientists, participants may not be able to imagine their use cases without a starting point. Through this process, we identified and incorporated more than 20 desired features into the VQS prototype over the period of a year during the participatory design process *(RQ2)*.

Finally, we conducted a realistic, qualitative evaluation to study how our improved VQS affected the way the users explore their data *(RQ3)*, as well as how these systems can be used within the data analysis workflow *(RQ4)*.

4 Understanding Scientific Data Analysis (RQ1)

Our initial inspiration for building a VQS came from informal discussions with many academic and industry analysts. Their current workflows required the analysts to manually examine a large number of visualizations to derive insights from their data. In this section, we address RQ1 by understanding the limitations and opportunities in existing scientific data analysis workflows in three research areas. We begin by describing the participants in these areas.

4.1 Participants, Datasets and Workflows

We recruited participants by reaching out to research groups who were interested in using VQSs for exploring their data via email. As we will describe in Section 9, we initially spoke to analysts from 12 different potential application areas and narrowed down to three use cases in astronomy, genetics, and material science for our participatory design study. We summarize the common properties of and differences between these three groups of researchers in Figure 1 and describe the desirable characteristics common to these datasets that make them suitable for VQSs in Section 9.

Six scientists from three research groups participated in the design of *zenvisage*. The evaluation study participants included these six scientists, along with three "blank-slate" participants who had never encountered *zenvisage* before. While participatory design subjects actively provided feedback on *zenvisage* with their data, they only saw us demonstrating their requested features and explaining the system to them, rather than actively using the system on their own. So the evaluation study was the first time that all nine of the participants used *zenvisage* to explore their datasets. On average, the participants had more than 8 years of research experience working in their respective fields. We list the participants in Table 1, and will refer to them by their anonymized ID as listed in the table throughout the paper.

	ID	Dataset	Participatory design participant	Position	Years of experience in subject area	Dataset Familiarity (1-5)
astro	**A1**	DES	Yes	Research scientist	10	3
	A2	Kepler	No	Postdoc	8	5
	A3	Kepler	No	Postdoc	8	5
genetics	**G1**	Mouse	Yes	Graduate student	4	4
	G2	Cancer	No	Graduate student	2	2
	G3	Mouse	Yes	Professor	10	2
matsci	**M1**	Solvent (8k)	Yes	Postdoc	4	5
	M2	Solvent (Full)	Yes	Professor	10	5
	M3	Solvent (Full)	Yes	Graduate student	3	5

Table 1: Participant information. The Likert scale used for dataset familiarity ranges from 1 (not at all familiar) to 5 (extremely familiar).

The research questions and objectives of the participants were diverse even among those in the same subject area and using the same dataset. Examples of research questions included:

- Understanding gene expression profiles of breast cancer cells that exhibit induced, transient, and repressed patterns after a particular treatment.
- Studying common patterns among stars that exhibit planetary transits versus stars that do not, from the Kepler space telescope[1].
- Identifying battery solvents with favorable properties and mass production potential through studying how changes in certain chemical properties correlate with changes in other chemical properties.

[1] `www.nasa.gov/mission_pages/kepler/main/index.html`

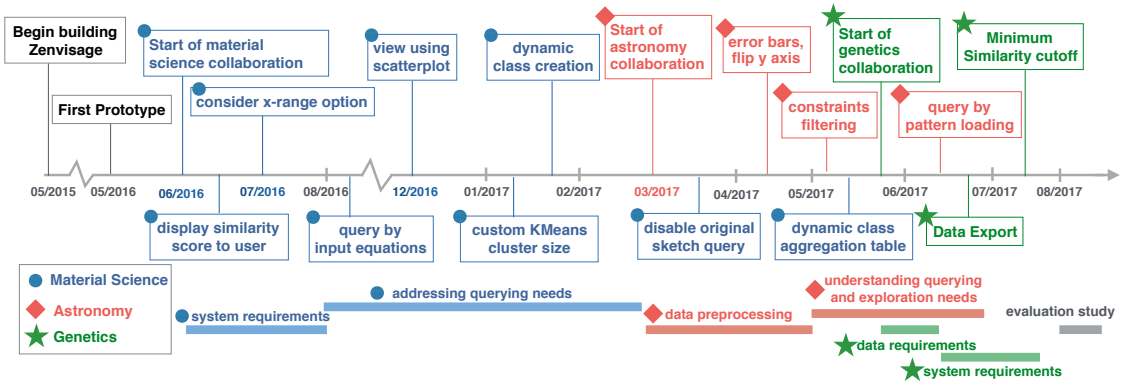


Figure 3: Participatory design timeline for the scientific use cases.

The pre-study survey with the participants showed that out of all of the steps in their data analysis workflow[2], they spend most of their time computing statistics and creating visualizations. The main bottlenecks cited in their existing workflows included the challenge of dealing with large amounts of data, writing custom processing and analysis scripts, and long turnaround times incurred by making modifications to an upstream operation in a segmented workflow.

During the participatory design process, we collaborated with each of the teams closely with an average of two meetings per month, where we learned about their datasets, objectives, and how VQSs could help address their research questions. A detailed timeline of our engagement with the participants and the features inspired by their use cases can be found in Figure 3. Participants provided datasets they were exploring from their domain, whereby they had a vested interest to use a VQS to address their own research questions. We describe the three scientific use cases below.

Astronomy (*astro*): The Dark Energy Survey (DES) is a multi-institutional project with over 400 scientists. Scientists use a multi-band telescope that takes images of 300 million galaxies over 525 nights to study dark energy [14]. The telescope also focuses on smaller patches of the sky on a weekly interval to discover astrophysical transients (objects whose brightness changes dramatically as a function of time), such as supernova explosions or quasars. The output is a time series of brightness observations associated with each object extracted from the images observed.

For over five months, we worked closely with an astronomer on the project's data management team working at a supercomputing facility. The scientific goal is to identify a smaller set of potential candidates that may be astrophysical transients in order to study their properties in more detail. These insights can help further constrain physical models regarding the formation of these objects.

Genetics (*genetics*): Gene expression is a common data type used in genomics and is obtained via microarray experiments. The data used in the participatory design sessions was the gene expression data over time for mouse stem cells aggregated over multiple experiments, downloaded from an online database[3].

We worked with a graduate student and a PI at a research university over three months who were using gene expression data to better understand how genes are related to phenotypes expressed during early development [16,37]. They were interested in using *zenvisage* to cluster gene expression data before conducting analysis with a downstream machine learning workflow.

Material Science (*matsci*): We collaborated with material scientists at a research university who are working to identify solvents that can improve battery performance and stability. These scientists work with large datasets containing over 25 chemical properties for more than 280,000 different solvents obtained from simulations. Once they have identified a solvent that also produces favorable results in an experiment, they identify other solvents with similar properties, which may be cheaper or safer to manufacture at an industrial scale.

We worked closely with a graduate students, a postdoctoral researcher, and a PI for over a year to design a sensible way of exploring their data using VQSs. Each row of their dataset represents a unique solvent, and consists of 25 different chemical attributes. They wanted to use *zenvisage* to identify solvents that not only have similar properties to known solvents but also are more favorable (e.g. cheaper or safer to manufacture), as well as to understand how changes in certain chemical attributes affects them.

4.2 Cognitive Walkthrough Sessions

During the design study, we observed the participants as they conducted a cognitive walkthrough demonstrating every component of their current data analysis workflow. Cognitive walkthroughs highlight the existing workflows and behavior that participants have adopted for conducting certain tasks [34].

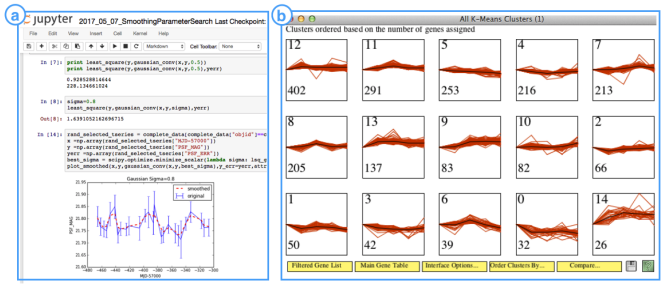


Figure 4: Examples of the scientists' original workflow: a) The astronomer performs various data analysis task using the Jupyter notebook environment, b) The geneticists uses a domain-specific software to examine clustering outputs.

Astronomy: Since astronomical datasets are often terabytes in scale, they are often processed and stored in highly specialized data management systems in supercomputing centers. The collaboration's data management team has created a command-line interface that enables users to easily query, browse, and download their data. After the data is downloaded, most of the work is done programmatically through Python in an interactive Jupyter notebook environment. The astronomer inspects the data schema, performs data cleaning and wrangling, computes relevant statistics, and generates visualizations to search for anomalies or objects of interest, as shown in Figure 4a.

While an experienced astronomer who has examined many transient light curves can often distinguish an interesting transient object from noise by sight, they must visually examine and iterate through large numbers of visualizations of candidate objects. Manual searching is time-consuming and error prone as the large majority of the objects are not astronomical transients. Participant A1 was interested in *zenvisage* as he recognized how specific pattern queries could help scientists directly search for these rare objects.

[2]This includes viewing and browsing data, data cleaning and wrangling, computing statistics, data visualization, and model building or machine learning.

[3]`ncbi.nlm.nih.gov/geo/`

Genetics: Participant G1 processes the raw microarray data by using a preprocessing script written in R. To analyze the data, the preprocessed data is loaded into a desktop application for visualizing and clustering gene expression data[4]. G1 sets several clustering and visualization parameters on the interface before pressing a button to execute the clustering algorithm. The cluster visualizations are then displayed as overlaid time series for each cluster, as shown in the visualization in Figure 4b. G1 visually inspects that all the patterns in each cluster looks "clean" and checks the number of outlier genes that do not fall into any of the clusters. If the number of outliers is high or the visualizations look unclean, she reruns the clustering by increasing the number of clusters. When the visualized clusters look "good enough", G1 exports the cluster patterns into a csv file to be used as features in their downstream regression tasks.

Prior to the study, the student (G1) and PI (G3) spent over a month attempting to determine the best number of clusters for their upstream analysis based on a series of static visualizations and statistics computed after clustering. While regenerating their results took no more than 15 minutes every time they made a change, the multi-step, segmented workflow meant that all changes had to be done offline, so that valuable meeting time was not wasted trying to regenerate results. The team had a vested interest in participating in the design of *zenvisage* as they saw how the interactive nature of VQSs and the ability to query other time series with clustering results could dramatically speed up their collaborative analysis process.

Material Science: Participant M1 starts his data exploration process with a list of known and proven solvents as a reference. For instance, he would search for solvents which have boiling point over 300 Kelvins and the lithium solvation energy under 10 kcal/mol using basic SQL queries. This helps him narrow down the list of solvents, and move on to the other properties for similar processing. The scientist also considers the availability and the cost of the solvents while exploring the dataset. When the remaining list of the solvents is sufficiently small, he drills down to more detail (e.g., such as looking at the chemical structure of the solvents to consider the feasibility of conducting experiments with the solvent). While he could identify potential solvents through manual lookup and comparison, the process lacked the ability to reveal complicated trends and patterns that might be hidden, such as how the change in one attribute can affect the behavior of other attributes of a solvent. M1 was interested in using a VQS as it was infeasible for him to manually compare between large numbers of solvents and their associated properties manually.

5 Themes Emerging from Participatory Design (RQ2)

In the previous section, we gained an understanding of the current analysis workflows employed in the three use cases. Next, to address RQ2, we employed participatory design with our scientists to incorporate key features missing in our original VQS, and unaddressed in their current workflows. We discovered three central themes encapsulating these features that are important to facilitate rapid hypothesis generation and insight discovery, but are missing in prior VQSs. While some of our findings echo prior work on system-level taxonomies of visualization tasks [4, 19], we highlight how specific analytic tasks and interaction features could be used to enhance VQSs in particular. In particular, we learned that *participants wanted more control over the internals of the systems and an integrated workflow that helped streamline their analysis when using VQSs.*

5.1 Uninterrupted workflow

Our cognitive walkthroughs revealed that in many participants' existing workflows, they switched between parameter specification, code execution, and visualization comparisons. The non-interactive nature of these segmented workflows has been shown to incur a large cognitive barrier during exploratory data analysis [24]. In addition, since scientific research often takes place in a collaborative setting, this means that the data sense-making process could be delayed by weeks because the analysis-to-results phases needed to be rerun offline based on changes that were suggested during a meeting. Moreover, data-cleaning emerged as a common pain-point, echoing prior work [21, 22].

Integrative preprocessing through interactive smoothing: While *zenvisage* does not attempt to solve all of the pre-processing issues that we faced during participatory design, we identified that data smoothing is a common data cleaning procedurethat would benefit from a tight integration between pre-processing and visual analysis.

Data smoothing is a denoising procedure that generates a smoothed pattern approximating key features of the visualized trend with less noise. Smoothing also raises an interesting trade-off between the smoothness of the curve and the quality of shape-matching for VQSs. If the visualization is over-smoothed, then shape matching would return results that only loosely resemble the query pattern. However, if no smoothing is applied, then the noise may dominate the overall trend, which could also lead to bad pattern matches. In addition, it is often hard to tell what the appropriate smoothing parameter should be applied simply by visualizing a small number of sampled visualization, as one would do in an offline analysis.

To address this issue, we developed an interface for users to interactively adjust the data smoothing algorithm and parameters on-the-fly to update the resulting visualizations accordingly (Figure 5a). This was applied to the *matsci* and *astro* use cases, as both had noisy and dense observational data.

Facilitating export for downstream analysis: Since VQSs are designed to be exploratory tools that suggest potential directions for further analysis, rather than for performing one-shot operations, we asked participants how they envisioned themselves using VQSs in their workflow. Both the *astro* and *genetics* participants wanted to use VQSs as a way to identify interesting objects or characteristic patterns, which they will later feed into a more advanced downstream pipeline.

To smoothen the transition between the VQS and their downstream analysis, we implemented export functionalities for downloading the similarity, representative trend and outlier results as csv files (Figure 5b). Individual visualizations can also be downloaded by double-clicking individual figures (Figure 5c) to facilitate easier sharing of visualization results with collaborators.

5.2 Increasing expressiveness of querying capabilities

While the interactions in our original prototype enabled simple visual queries, many scientists were interested in extending their querying capabilities, either through different querying modalities or through more flexible query specification methods.

Input Equations: Our *matsci* participants expressed that some solvents can have analytical models that characterize the relationships between chemical properties. They wanted to find solvents that satisfied these relationships. We implemented a feature that plots a given function (e.g. $y = x^2$) on the canvas, which is then used as input for similarity search (Figure 5d).

Upload Pattern as Query: While the input equation is useful when simple analytical models exist, this may not be true for other domains. In these cases, users can upload a query pattern of a sequence of points (Figure 5e). This is useful for patterns generated from advanced computational models used for understanding scientific processes, usually as part of the downstream analysis of the exploratory workflow.

Consider/Ignore x-range: We improved query specification by allowing users to change how the shape-matching criterion is applied. For finding supernovae, A1 primarily cared about the existence of a peak above a certain amplitude with an appropriate width of the curve, rather than the exact time that the event occurred, leading them to use the consider x-range feature. G1 also expressed that she does not really know what is the "trigger point" of when the expression level of a gene will rise and it would be interesting to find all "rising" profiles independent of the change-point. We implemented an option to ignore the x-range in shape matching (Figure 5f) and a brushing mechanism that enables users to select the specific x-region they want to perform their shape matching on (Figure 5g).

[4] www.cs.cmu.edu/~jernst/stem/

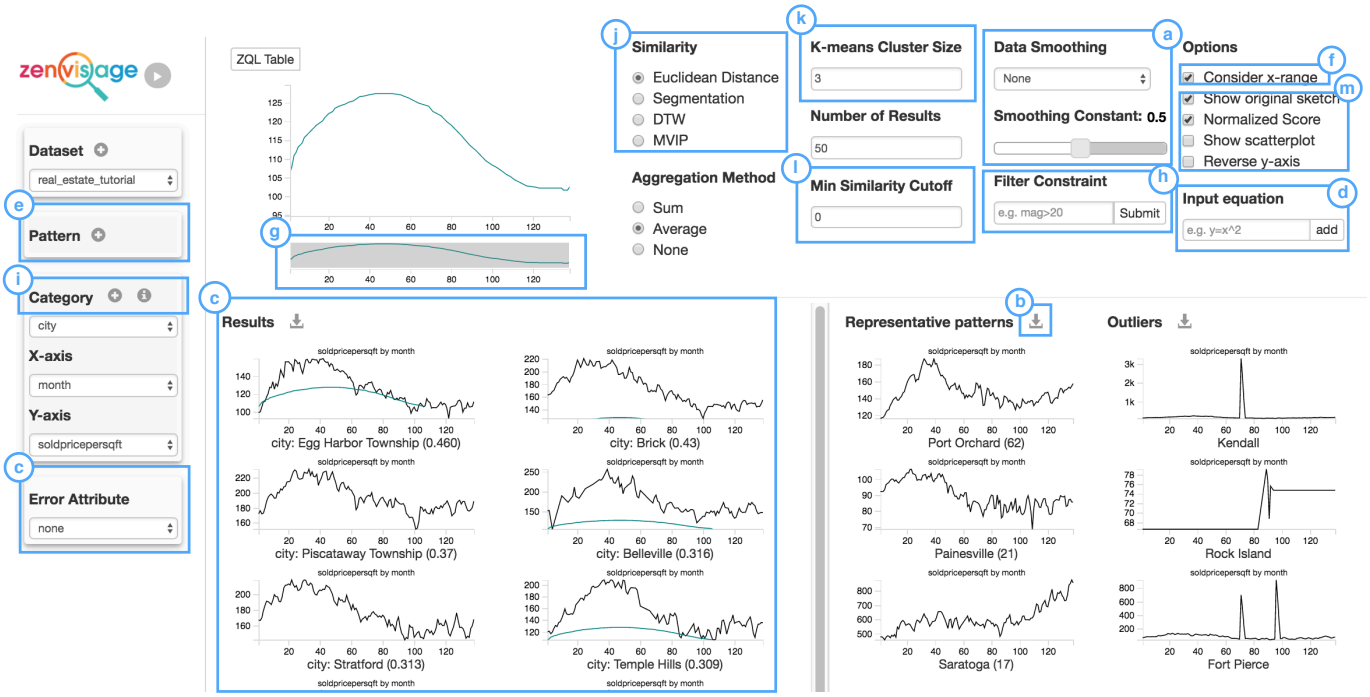


Figure 5: Our VQS after participatory design, which includes: the ability to preprocess via (a) interactive smoothing; (b, c) the ability to export data outputs ; querying functionalities via (d) equations and (e) patterns; query specification mechanisms including (f) x-range invariance, (g) x-range selection and filtering, (h) Filtering, and (i) Dynamic class creation; (j, k, l) system parameter options; (m) visualization display options. Prior to the participatory design, *zenvisage* only included a single sketch input with no additional options. *zenvisage* also displayed representative patterns and outlier patterns, as shown in Figure 2.

5.3 Ability to dynamically facet through subsets

Past studies in taxonomies of visualization tasks have shown that it is important to design features that enable users to select relevant subsets of data in visual analytics [4, 19]. We designed two dynamic faceting features coupled with coordinated views that enabled users to specify subsets of data they are querying on and see immediate changes updated in the query, representative, and outlier results.

Filtering Constraints: Users with large datasets first used their domain knowledge to narrow down their search to a subset of data. This would increase their chances of finding an interesting pattern for a given query. To filter data, users could submit one or more SQL-like `WHERE` conditions as filter constraints in a text field (Figure 5h).

Dynamic Class Creation: In order to address material scientists' needs for creating subsets (or classes) of data on-the-fly to make comparisons between them, we implemented dynamic class creation. This feature allows users to bucket data points into customized classes based on existing properties, and subsequently allows users to compare between the customized classes. For example, the scientists can create three different classes based on a single property alone: Solvents with ionization potential under -10 kJ/mol, over -8 kJ/mol, and ones that fall between -10 and -8 kJ/mol. Then, they could browse how the lithium solvation energy differed for the the three custom classes.

Scientists can utilize multiple properties to create custom classes, effectively slicing-and-dicing the data based on their needs. The information regarding the created classes is displayed in the dynamic class information table or as a tooltip over the aggregated visualizations, as shown in Figure 6.

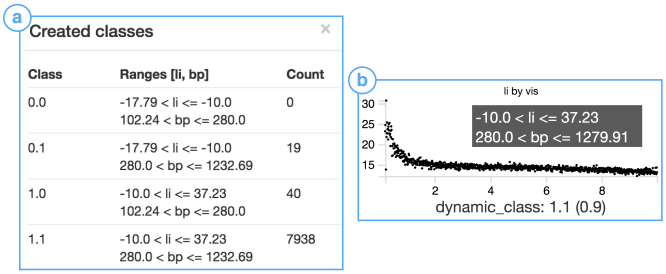


Figure 6: Example of dynamic classes. (a) Four different classes with different Lithium solvation energies (li) and boiling point (bp) attributes based on user-defined data ranges. (b) Users can hover over the visualizations for each dynamic class to see the corresponding attribute ranges for each class. The visualizations of dynamic classes are aggregate across all the visualizations that lie in that class based on the user-selected aggregation method.

5.4 Finer System-level Control and Understanding

During the participatory design exercise, we found that many of the features suggested by the participants indicated they wanted finer control of the system. Prior work in direct manipulation visual interfaces has suggested that finer-grained control enabled users to discover patterns and rapidly generate hypothesis based on visual feedback [44, 45].

Controlling VQS internals: In addition to query and dataset specifications, users also wanted the ability to modify the model parameters in *zenvisage*. Our findings echoed Chuang et al. [9], which showed that the ability to modify the model can facilitate interpretation and trust in model-driven visualizations, especially during early-stage exploration. These model parameter options include the ability to change the choice of similarity metrics (Figure 5j), the cluster size in the representative patterns (Figure 5k), setting a minimum similarity threshold for displaying the search results (Figure 5l), and the ability to tune the smoothing algorithm and parameter (Figure 5a).

Displaying interpretable explanations for VQS recommendations: Explanatory system outputs include displaying similarity scores of the outputs, the number of datapoints in each cluster, and overlaying the original query sketch on the return visualization for comparison (Figure 5m). We further provided display-related options for plotting modifications, including displaying error bars, and toggling between a scatterplot and line chart view, to help analysts better understand the visualizations.

6 Final Evaluation Study: Methodology and Results

Our final evaluation study addresses RQ3 and RQ4—whether our new and improved VQS helps accelerate insight, and how it could fit into real analysis workflows. Participants for the evaluation study were recruited from each of the three aforementioned research groups, as well as domain-specific mailing lists. Prior to the study, we asked the potential participants to fill out a pre-study survey to determine their eligibility. Eligibility criteria included: being an active researcher in the subject area with more than one year of research experience, and having worked on a research project involving data of the same nature as that used in the participatory design. Four of the user studies were conducted remotely.

Participants had the option of exploring their own dataset or an existing dataset that they provided to us during the participatory design process. All three blank-slate participants opted to explore their own datasets. After loading their dataset, we emailed them a screenshot of a visualization from our tool to verify that we configured the system to meet their needs.

At the start, participants were provided with an interactive walk-through explaining the details of the features offered in our VQS. The participants were then given approximately ten minutes to experience a guided exploration of our VQS with a preloaded real-estate example dataset from Zillow [2]. After familiarizing themselves with the tool, we loaded the participant's dataset and suggested an appropriate choice of axis to begin the exploration. Participants were encouraged to talk-aloud during the data exploration phase.

During the exploration phase, participants were informed that they could use other tools as needed. If the participant was out of ideas, we suggested one of the ten main functionalities[5] that they had not yet covered. If any of these operations were not applicable to their specific dataset, they were allowed to skip the operation after having considered how it may or may not be applicable to their workflow. The user study ended after they covered all ten main functionalities. On average, the main exploration phase lasted for 63 minutes. After the study, we asked them open-ended questions about their experience.

6.1 Data Collection & Analysis

We recorded audio, video screen captures, and click-stream logs of the participant's actions during the evaluation study. We analyzed the transcriptions of these recordings through open-coding and categorized every event in the user study using the coding labels:

- Insight (Science) **[IS]**: Insight that connected back to the science (e.g. "This cluster resembles a repressed gene.")
- Insight (Data) **[ID]**: Data-related insights (e.g. "A bug in my data cleaning code generated this peak artifact.")
- Provoke (Science) **[PS]**: Interactions or observations made while using the VQS that provoked a scientific hypothesis to be generated.
- Provoke (Data) **[PD]**: Interactions or observations made while using the VQS that provoked further data actions to continue the investigation.
- Confusion **[C]**: Participants were confused during this part of the analysis.
- Want **[W]**: Additional features that participant wants, which is not currently available on the system.
- External Tools **[E]**: The use of external tools outside of *zenvisage* to complement the analysis process.

In addition, based on the usage of each feature during the user study, we categorized the features into one of the three usage types:

- Practical usage **[P]**: Features used in a sensible and meaningful way.
- Envisioned usage **[E]**: Features which could be used practically if the envisioned data was available or if they conducted downstream analysis, but was not performed due to the limited time during the user study.
- Not useful **[N]**: Features that are not useful or do not make sense for the participant's research question and dataset.

We chose to derive these labels from the user study transcription rather than through self-reporting to circumvent the bias that users may have when self-reporting, which can often artificially inflate the usefulness of the feature or tool under examination.

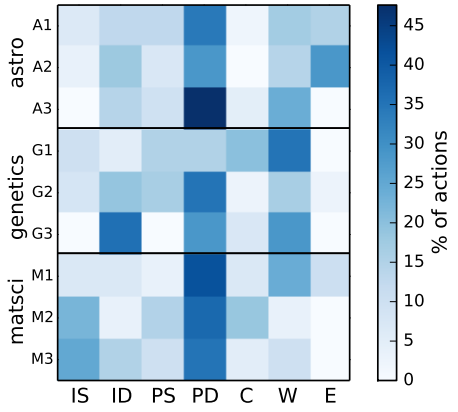


Figure 7: Heatmap showing percentage of participants' actions on *zenvisage* falling under each thematic encoding category. *zenvisage* provokes participants to perform many data actions **[PD]** and generate scientific hypothesis **[PS]**. Occasionally, the series of data operations and hypothesis can lead to an scientific **[IS]** or data-related insight **[ID]**.

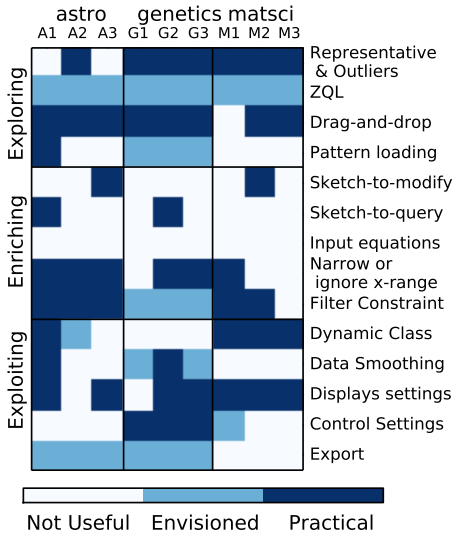


Figure 8: Heatmap of features categorized as practical usage (P), envisioned usage (E), and not useful (N). We find that participants preferred to query using bottom-up methods such as drag-and-drop over top-down approaches such as sketching or input equations. Participants found that data faceting via filter constraints and dynamic class creation were powerful ways to compare between subgroups or filtered subsets. The columns are arranged in the order of subject areas and the features are arranged in the order of the three foraging acts.

The audio recordings and transcriptions of pre- and post-study interview questions are thematically encoded and summarized in Figure 7 and 8. Our study results fall into two main themes, which will be discussed in subsequent sections. First, we discuss how VQS features were used in practice to achieve scientific insights (RQ3). We find that VQSs can enable rapid, fluid iteration, catalyzing new questions or insights; that different querying modalities in VQSs support different forms of exploration; and that expressive querying allowed participants to compose novel analysis patterns. Second, we determine where VQSs fit into real data analysis workflows (RQ4). We find that VQSs can be

[5] query by sketching, drag-and-drop, pattern loading, input equations, representative and outliers, narrow/ignore x-range options, filtering, data smoothing, creating dynamic classes, data export

used for a range of tasks that go beyond just exploration; that participants used the outputs from VQSs in various ways; and that VQSs are most appropriate for certain types of datasets.

7 Actionable VQS features for scientific insight (RQ3)

7.1 Top-down v.s. Bottom-up analysis

To contextualize our study results with respect to prior work on how analysts make sense of data, we employ Pirolli and Card's [38] information foraging framework for domain-experts. Pirolli and Card's notional model distinguishes between information processing tasks that are *top-down* (from theory to data) and *bottom-up* (from data to theory). Recent work have also used the top-down v.s. bottom-up framework in understanding visualization construction [27]. In the context of visualization querying, top-down approaches are attribute-level specification for a query or an action, whereas bottom-up approaches originate from the data (or equivalently, the visualization).

7.1.1 Querying Behavior

Our interactions with the scientists showed that different modalities for inputting a query can be useful for different problem contexts. To our surprise, despite the prevalence of sketch-to-query systems in literature, only two out of our nine users had a practical usage for querying by sketching. Overall, we found that bottom-up querying via drag-and-drop was more intuitive and more commonly used than top-down querying methods, such as sketching or input equations.

The main reason why participants did not find sketching useful was that they often do not start their analysis with a pattern in mind. Later, their intuition about what to query is derived from other visualizations that they see in the VQS, in which case it made more sense to query using those visualizations as examples directly (via drag-and-drop). In addition, even if a user has a query pattern in mind, sketch queries can be ambiguous [11] or impossible to draw by sketching (e.g. A2 looked for a highly-varying signal enveloped by a sinusoidal pattern indicating planetary rotation).

The latter case is also evident from the unexpected use cases where sketching was simply used as a mechanism to modify the dragged-and-dropped queries. As shown in Figure 9 (top), M2 first sketched a pattern to find solvent classes with anticorrelated properties. However, the sketched query did not return the visualization he was interested in. So, he instead dragged and dropped one of the peripheral visualizations that was close enough to his desired visualization to the sketchpad and then smoothed out the noise due to outlier datapoints by tracing a sketch over the visualization. M2 repeated this workflow twice in separate occurrences during the study and was able to derive insights from the results. Likewise, A3 was interested in pulsating stars characterized by dramatic changes in the amplitudes of the light curves. During the search, hotspots on stellar surfaces often show up as false positives as they also result in dramatic amplitude fluctuations, but happen at a regular intervals. In the VQS, A3 looked for patterns that exhibits amplitude variations, but also some irregularities. As shown in Figure 9 (bottom), she first picked out a regular pattern (suspected star spot), then modified it slightly so that the pattern looks more irregular.

Both of these use cases suggest that while sketching is an useful analogy for people to express their queries, *the existing ad-hoc, sketch-only model for visualization querying is insufficient without data examples that can help analysts jumpstart their exploration*. We suspect that this is why existing sketch-to-query systems are not commonly adopted in practice. This result, however, points to a potential need for high-level query modification interfaces that enable more precise visualization query specification or couple sketching with examples as a starting point in future VQSs.

Despite functional fitting being common in scientific data analysis, Figure 8 shows that querying by equation is also unpopular for similar reasons. In addition, the visualizations for both *astro* and *genetics* exhibit complex processes that could not be written down as an equation analytically. However, even when analytical relationships do exist (in the case of *matsci*), it is challenging to formulate functional forms in an prescriptive, ad-hoc manner.

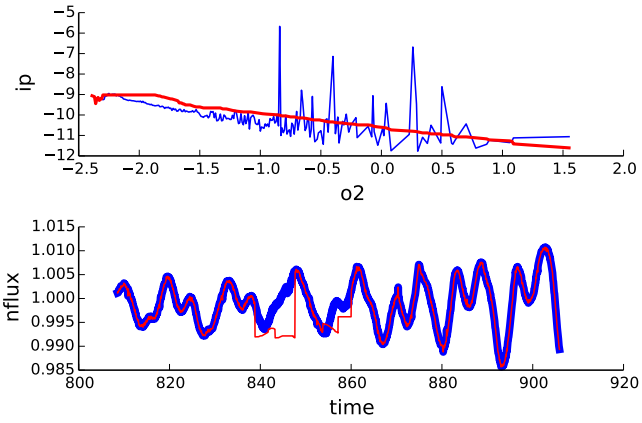


Figure 9: Examples of query modification by M2 (top) and A3 (bottom) performed during the study The inital drag-and-dropped query is shown in blue and the sketch-modified queries is shown in red.

In contrast, when we looked at bottom-up querying approaches, drag-and-drop was the most frequently used querying mechanism. Examples of practical uses of drag-and-drop includes inspecting the top-most similar visualizations that lie in a cluster and finding visualizations that are similar to an object of interest that exhibits a desired pattern. Likewise, many participants envisioned use cases for pattern loading. The ability to load in data patterns as a query would enable users to compare visualizations between different experiments, species, or surveys, query with known patterns from an external reference catalog (e.g. important genes of interest, objects labeled as supernovae), or verify the results of a simulation or downstream analysis by finding similar patterns in their existing dataset. In addition, users can also specify a more precise query that captures the essential shape features of a desired pattern (e.g. amplitude, width of peak), that cannot be precisely drawn as a sketched pattern. For example, Figure 1 (left) shows that the width of the light curve is characteristic to the type of supernovae that it is associated with, so querying with an exact pattern template would be helpful for distinguishing the patterns of interest from noise.

While the usage of each querying feature may vary from one participant to the next, generally, drag-and-drop and pattern upload are considered bottom-up approaches that go from data to theory by enabling users to query via examples of known visualizations. For top-down approaches such as query by sketching and input equations, the user starts with an intuition about how their desired patterns should look like based on theory, then queries based on that. Our results indicate that *bottom-up querying approaches are preferred over top-down when the users have no desired patterns in mind*, which is commonly the case for exploratory data analysis.

7.1.2 Faceted Exploration Behavior

All participants either envisioned a use case or utilized the data faceting functionalities offered in *zenvisage* (filtering, dynamic class) to explore and compare subsets of their data.

A1 expressed that even though the filtering step could be easily done programmatically on the dataset and reloaded into *zenvisage*, the filter constraint was a powerful way to dynamically test their hypothesis. Interactive filtering lowers the barrier between the iterative hypothesize-then-compare cycle, thereby enabling participants to test conditions and tune values that they would not have otherwise modified as much. During the study, participants used filtering to address questions such as: "Are there more genes similar to a known activator when we subselect only the differentially expressed genes (e.g. DIFFEXP=1)?" (G2) or "Can I find more supernovae candidates if I query only on objects that are bright and classified as a star (e.g. flux>10 AND CLASS_STAR=1)?" (A1). Three participants had also used filtering

as a way to pick out individual objects of interest to query with. For example, G2 set the filter as gene='9687' and explained that because "this gene is regulated by the estrogen receptor, when we search for other genes that resemble this gene, we can find other genes that are potentially affected by the same factors." As shown in Figures 7 and 8, participants with the top-most provoked data actions (A1, A3, G2) made heavy use of filtering to explore different subsets of the data.

While filtering enabled users to narrow down to a selected data subset, dynamic class creation enabled users to compare the relationships between multiple attributes and between subgroups of data. For example, M2 divided the solvents in the database to eight different classes based on the voltage properties, state of matter at room temperature, and viscosity levels, by dynamically setting the cutoff values to create these classes. By exploring the created custom classes, M2 learned that the relationship between viscosity and lithium solvation energy is independent of whether a solvent belongs to the class of high voltage or low voltage solvents. M2 cited that dynamic class creation was central to learning about this previously-unknown property of these two attributes:

> All this is really possible because of dynamic class creation, so this allows you to bucket your intuition and put that together. [...] I can now bucket things as high voltage stable, liquid stable, viscous, or not viscous and start doing this classification quickly and start to explore trends. [...] And look how quickly we can do it! Quite good!

Participants employed *a mix of bottom-up and top-down approaches when faceting through data in VQS*, including narrowing the search space based on some intuition about a phenomena, selecting individual visualizations, or specifying high-level groupings to compare and query with.

7.1.3 Novel workflows focusing on one central foraging act.

Pirolli and Card's notional model further characterizes the trade-offs between three central activities in the information foraging process: exploring, enriching, and exploiting [38]. *Exploring* involves gathering more information during the analysis. In the context of VQSs, exploring includes viewing representatives and outliers, incidental viewing of other visualizations in the ranked search results, and querying via drag-and-drop and pattern-loading. *Enriching* involves tasks that narrow down the space of analysis, such as filtering, dynamic class creation, query specification, and querying via input equations and sketching. *Exploiting* involves spending time inspecting the results in more detail, including interpreting each visualization in greater detail or making plotting changes that offer another perspective (smoothing, display, and interpretability settings).

In addition to observing how participants use features within the VQS, we also find that participants often create unexpected workflows that chain together multiple analysis steps, including interactions, controls, and queries in order to address a higher-level research question. These behaviors can be explained in terms of the foraging acts proposed by Piroli and Card. We find that participants often construct a central workflow, which they then iterate on while adding additional variations. Their *central workflow often resembles one of the three foraging acts* that aligns with the type of research question and dataset they are interested in. The variations are based on intermixing their central workflow with the other two foraging acts.

For example, geneticists were mostly interested in *exploring* clusters to gain an overall sense what profiles exist in the dataset through representative trends and therefore queried mainly through drag-and-drop. The variations to their main workflow include changing cluster sizes and display settings to offer them different perspectives on the dataset (*exploit*) and filtering on data attributes (*enriching*). The main workflow for the astronomers in our user study involves *enriching*, either through the creation of groups or via filtering data subsets. The main workflow for material scientists involves *exploiting*, since they spend the majority of their efforts performing "close-reading" of individual visualizations to understand the relationships between physical variables.

7.2 The need for comparisons across multiple collections of visualizations on different data attributes.

All participants wanted the ability to perform comparisons across multiple collections of visualizations on different data attributes, including comparison tasks such as: comparing between genes that belong to different functional groups (G1,G3); comparing between stars that harbor planets (which exhibits a transit pattern) and stars that don't (A2); comparing similarity and dissimilarity between two different sets of y measurements, such as time series of gene expression v.s. hypersensitivity (G2), stellar fluxes across different bands (A1); and comparing the correlation between different chemical properties for different solvent groups (M2).

These complex tasks are akin to the queries in the Zenvisage Query Language (ZQL) [47], which we did not demonstrate to our participants during the study to avoid confusion. Currently, to complete these tasks using *zenvisage*, participants would have had to use the querying interface multiple times while also remembering past results, which can be error-prone. Developing sensible interactions to map user intentions to sophisticated query languages like ZQL is an important direction for future work.

7.3 Integrated workflow results in more experimentation.

Our participants' original workflow often required them to compare between many visualizations manually through separate analysis and visualization steps. Three of the participants cited that this segmented analyze-then-visualize workflow was one of their chief bottlenecks. The cognitive overhead from the segmented workflow made them more hesitant to visualize the results of different parameters and data operations, as A2 noted:

> The quick visualization is something that I could not do on my current framework. I could not query as fast as you do; I need to wait for it, plot, and then compare. Every time I plot, I need to define subplots for 12 visualizations, then its slower. That's the reason why I sometimes plot less, and I rely more on the statistics from the likelihood tests. Sometimes I plot less than I really should be doing.

In Figure 7, we also find that *zenvisage* provoked users to generate many data operations (PD) and hypotheses (PS).

7.4 Rapid insights via VQSs catalyze new questions, hypotheses, and actions.

The ability to rapidly experiment with large numbers of hypotheses in real time is a crucial step in the agile creative process in helping analysts discover actionable insights [45]. Five out of nine participants discussed how the dynamic, interactive update of the visualization in *zenvisage* was the main advantage for using VQSs over their original workflow.

A common theme that we found across the geneticists is that they often gain their intuition about the data from the representative trends. One example of rapid insight discovery comes from G1, G2, and G3, who first identified that the three representative patterns shown in *zenvisage*—induced genes (profiles with expression levels staying up), repressed genes (started high but went down), and transients (go up and then come down at different time points)—corresponded to the same three groups of genes discussed in a recent publication [16].

The clusters provoked G2 to generate a hypothesis regarding the properties of transients: *"Is that because all the transient groups get clustered together, can I get sharp patterns that rise and ebb at different time points?"* To verify this hypothesis, G2 increased the parameter controlling the number of clusters and noticed that the cluster no longer exhibited the clean, intuitive patterns he had seen earlier. G3 expressed a similar sentiment and attempted to identify different clusters. He proceeded by inspecting the visualizations in the cluster via drag-and-drop and found a group of genes that all transitioned at the same timestep, while others transitioned at different timesteps. G3 described the process of using VQSs as doing "detective work" that provoked him to generate further scientific hypotheses as well as data actions.

8 VQSs WITHIN A SCIENTIFIC DATA ANALYSIS WORKFLOW (RQ4)

8.1 VQSs used beyond the exploration stage of analysis.

Two of the five datasets used in the user study were preliminary in the sense that the scientists had performed only basic data cleaning and had not explored this dataset in great detail themselves (A1, G2). This enabled us to get a sense of how VQSs can be used in practice at different stages of the analysis workflow. During the participatory design, we thought that our VQS would serve as a tool that help scientists find interesting patterns following which scientists could proceed to a more detailed and rigorous analysis based on these newly-obtained insights. We realized after the evaluation study that users were interested in using VQSs for more than one-step pattern-finding. For example, A2 commented:

> [*zenvisage* fits in] after the cleaning and after correcting for systematics, I will use *zenvisage* for a first visualization, not taking the advantage of all the features that *zenvisage* has, and then will perform a first analysis and then come back to *zenvisage* (in this analysis I will calculate the rotational period and some other values that could help me separate out the categories in *zenvisage*), then after I learn about the categories. In the end, I will again use *zenvisage* to visually inspect the data.

Participants also explained where they saw VQSs fit into their workflow, including *(i)* visualizing raw data before cleaning to learn about what types of outliers, representative trends, and artifacts are present in the data (A1, A2, G2, M3); *(ii)* data verification after cleaning to see if known patterns show up as expected (G1); and *(iii)* verifying the correctness of a simulation, by visualizing data from a simulation that is based on insights obtained using the VQS (G1, G3).

8.2 VQSs for verifying data fidelity and debugging.

Participants often used *zenvisage* to verify the fidelity of their data and perform debugging. For example, G2 crosschecked that there were no data artifacts of "genes with two peaks" via sketching. Via the visualizations displayed in the result, representative, and outlier panels in *zenvisage*, participants were able to gain a peripheral overview of the data and spot anomalies during exploration. For example, A1 spotted time series that were too faint to look like stars after applying a filter constraint of CLASS_STAR=1. After a series of visualizations of other query results and consultation with an external database, he concluded that the dataset had been incorrectly labelled with all the stars with CLASS_STAR=0 as 1 during data cleaning.

Explanatory display outputs in the VQS work closely in conjunction with finer control mechanisms so that users receive feedback on their data actions and immediately update their mental models. For example, the genetics participants modified the clustering parameters and verified that the size of each cluster still remained relatively even, in order to determine the best values of the parameters. This served as a verification mechanism that helped users build trust in the model outputs by cross checking with their intuition on what should happen to the result as they perform an operation [9]. Moreover, the dynamic, real-time update of VQSs aid rapid hypothesis generation and encourage scientists to try things that they would not have done otherwise, especially for exploratory tasks that had a low probability of producing interesting results, such as browsing for anomalies as a sanity check or data verification.

8.3 VQSs are used in conjunction with external tools.

During the user study, four of the scientists consulted external tools outside of the VQS as a reference. Two out of the four used the external tools to compute statistics, browse related datasets, or examine other data attributes. This shows that generic VQSs are not a one-size-fits-all solution, and domain-specific tools are often useful to provide context.

For example, A1 saw a strange dip in the time series in the VQS and wanted to see whether it was an artifact. He first checked the database to see if there is an error flag associated with the observation of this object and learned that this object is labelled as a galaxy. Upon comparing with the other visualization in the panel in the VQS, he noticed the difference in y values and hypothesized that object was so faint that it was below the detection limit of the instrument. After going to an external software to inspect the images, he verified his hypothesis as the image was so noisy and a bright nearby star may have saturated the imaging equipment and resulting in the dip in the observations.

We identified two main reason why scientists go to external tools to support their analysis in VQS. (1) Scientists often require multiple sources of data to test the validity of their hypothesis. For example, to determine whether a particular gene belongs to a regulatory network, G2 not only looks at the expression data in the VQS but also enrichment testing and knockout data. Visualizing these data often requires specialized tools and isn't supported by a generic VQS. To enable smoother transitions between tools, several participants expressed that VQSs should bookmark and track the history of visualizations that they had found interesting. (2) Scientists also use many different data attributes to better understand the data that they are visualizing in the VQS and further develop their hypotheses. Four participants wanted the VQS to support data summaries (histograms, statistics) on the non-visualized attributes to assist them with choosing appropriate values for filtering data subsets and class creation. For instance, A2 used his Jupyter Notebook during exploration to obtain summary statistics on the stellar radius.

8.4 Insights derived from VQS seen as preliminary and can be subjected to more robust testing or downstream modeling.

We ask participants the types of additional analysis they plan to run downstream after obtaining insights from *zenvisage*. Eight out of nine participants envisioned that the exporting functionalities in *zenvisage* are useful for directing them to the next step of their analysis workflow. For astronomers, the post-analysis tasks involve cross-checking and inspecting individual objects of interest more closely, including using external data types such as images. A1 discussed how he plans to perform a more rigorous "blind analysis", which involves taking the visualized data without any IDs or associated data attribute, and seeing if other statistical techniques yields the same set of interesting objects as the ones discovered visually through VQS. All genetics participants expressed that they will export the clusters and directly move onto the next stage of the analysis without additional verification, since they regard the results from VQS "simply as guidelines"(G3) that provide them with the intuition about what types of patterns exist in their dataset, before they start building advanced models. None of the material scientists wanted to export their data because they were more interested in insights gained from understanding relationships between chemical properties, rather than finding particular solvents. The question of how analysts understand and trust the outputs of VQS depending on the objectives of their analysis is an interesting direction for future work.

9 REFLECTIONS ON META-ANALYSIS METHODOLOGY AND LIMITATIONS

Design studies are common in visualization research and often culminate in the development of a domain-specific tool, such as visualization systems for micro-array data [12] or biological networks [7]. As discussed in Section 3.1, unlike these design studies, most prior work in VQSs is technique-driven and is rarely evaluated on real-world users, tasks, and datasets. Prior work distinguishes between problem-driven and technique-driven research and argues that the field of visualization research benefits from contributions made from both types of methodologies [28, 33]. Our work can be seen as a hybrid between problem and technique-driven work by using multiple case studies to more holistically address our research questions.

When considering a single use case study, our approach closely follows and inherits many of the benefits and pitfalls of MILCs [46], design studies [28], and insight-based evaluations [35]. However, there are several additional pros and cons that arises when we consider our multiple case studies approach.

P1: Conversation with potential collaborators establishes early-stage design specification and guidelines.

In selecting our three user groups, we spoke to participants from 12 different potential application domains ranging from connectonomics to protein networks, in a process similar to the *winnow* stage described in Sedlmair et al. [28]. These conversations drew us to consider what are the viable use cases for VQSs and where VQSs fail, which serves as a starting point for our design study. As a result, we distilled a set of characteristics that exemplifies use cases that may not be suitable for VQSs. We focused on the *astro*, *genetics*, *matsci* groups in this paper since their data can be viewed in a line chart format and their tasks required comparisons across many visualizations. It is also important to note that selecting a diverse set of use cases in terms of the types of users, data, challenges and analysis tasks as shown in Figure 1 is important for generalizing our results beyond a single use case.

There were many interesting potential scientific use cases that did not satisfy these criteria. For example, time-varying 2D maps representing the interactions between brain regions and protein-protein interactions are non-ordinal heatmaps, with no simple sketching analogy. While VQSs are not restricted to only line charts, supporting querying capabilities for other visualization types is a exciting direction of future work that is outside the scope of this study. Even when the data is time series-based, some potential application domains had data characteristics that made it difficult to use a VQS. For example, a neuroscientist noted that their time series only consists of 3-5 observations, since each observation required the dissection of a mouse brain. Visualizations with sparse datapoints can be difficult for existing shape matching algorithms.

P2: Parallel use case engagement results in more generalizable design and development.

One of the key benefits of having multiple case studies for a design study is that researchers can better evaluate a relevant set of analysis tasks to be addressed by the tool across use cases. During the design phase, researchers interact with multiple groups of participants to discuss a wide variety of problems present in their current analysis workflows and the types of tasks they want to perform on a VQS. As highlighted in Section 5, through a series of cognitive walkthroughs and regular meetings we can distill a set of common themes that emerges across multiple use cases and rank important and relevant problems to address with the tool. Of the 20 features we implemented, 16 were suggested by multiple use cases.

For example, we spoke to astronomers who wanted to preprocess the data so that sparse time series with few numbers of observations were not included in their dataset. In the same week, we spoke to material scientists who wanted to inspect only solvents with properties above a certain threshold. Through these use cases, data filtering arose as a crucial, common operation that we needed to incorporate into *zenvisage* in order to support these class of queries.

P3: Cross pollination of feature design leads to unexpected usage.

In addition to seeking common themes and solutions across use cases, we found that discussing newly-added features that addressed a particular use case with other groups of participants often results in unexpected usage of the feature. Continuing with the filter example, we found that participants wanted to use filter to do many types of tasks that was not originally envisioned in the design study. For example, astronomers reformatted their data and naming scheme to make use of the filter functionality to search across data of selected set of observational cycles, years, or bands. Astronomers and geneticists also used the filter operation to select specific known objects to query similar patterns. Since these unit features were inspired by multiple use cases, they cover a diverse range of foraging acts, as described Section 7.1.3, and thereby support analysts to easily create novel analysis workflows within the VQS.

P4: Conflicting design choices between different users results in feature generalization.

While there are common features across use cases, we also encounter cases where variations of a feature was required or use cases that did not want a specific feature. In building a generalized tool, we opted for a default choice that was transparent to the users and a tunable option that the individual users may chose to customize. Many of the options in the systems settings panel such as data smoothing and aggregation are examples of this. For example, G1 and G2 did not want to turn on smoothing since their data was already Loess smoothed, whereas A1 was used to Gaussian Kernel smoothing. Such feature generalization enables us to create features that could be useful across multiple use cases.

C1: Initial collaboration is time-consuming and requires significant effort.

The process of preparing the analyst's dataset to be ready to use in our prototype tool for the design study is time-consuming. This initial phase can largely be boiled down to data requirements and system requirements. In all three of the scientific use cases, there was a period of time for establishing the minimum data and systems requirements for understanding how a VQS can be used for the particular scientific use case. Data requirements includes gaining sufficient understanding of the problem domain, understanding the types of data suitable for the system, cleaning and loading of the dataset into the tool, so that it is appropriate for use in a VQS. The minimum system requirements include features that are required for the data to be visualized appropriately, such as displaying error bars or showing the time series as scatterplots. Often, participants can only envision the types of queries that they would like to make and how finer query specifications and system controls could help better address their science questions after seeing their data displayed for the first time in the prototype system.

C2: Failure to select the right problems and features during the design phase can result in wasted engineering efforts.

During the design phase, there were numerous problems and features proposed by participants, but not all were incorporated in the tool. Among our available meeting logs with participants, we found that the reasons for not taking a feature from the design stage to the implementation stage includes:

- Nice-to-haves: One of the most common reasons for unincorporated features comes from participant's requests for nice-to-have features. The amount of nice-to-have features that one could envision for the tool is endless. We use two criteria to heuristically judge whether to implement a particular feature:

    1. *Necessity:* Without this feature, can the analyst still work with this dataset using our tool and achieve what they want to do?
    2. *Generality:* If this feature was implemented, will it benefit only this specific use case or be potentially useful for other applications as well?

- "One-shot" operations: We decided not to include features that were "one-shot" operations that only needed to be performed once during the analysis workflow. For example, certain data preprocessing operations such as filtering null values only needed to be performed once unlike the data smoothing feature that we added, which was a preprocessing step that could be iteratively performed along with other operations in the VQS. This design choice is specific to our VQS design study.

- Substantial research or engineering effort: Some proposed features do not make sense in the context of VQS. For example, the *matsci* group proposed functional fitting to obtain fitting coefficients. Other features requires a completely different set of research questions. For example, the question of how to properly compute similarity between time series with non-uniform number of datapoints arose in the astronomy and genetics use case, but requires the development of a novel distance metric or algorithm that is out of the scope of the RQs of our design study.

- Underdeveloped ideas: Other feature requirements came from casual specification that were underspecified. For example, A1 wanted to look for objects that have deficiency in one band and high emission in another band, but what brightness levels qualifies as a deficiency is ambiguous.

Failure to identify these early signs in the design phase may result in feature implementations that turn out not to be useful for the participants. Our experimental evaluation shows that some of our feature choices also suffer from these pitfalls. For example, we incorporated the feature to reverse the y-axis so that astronomers could better understand magnitude measurements (as larger magnitudes counter-intuitively corresponds to dimmer objects). In hindsight, the feature was not crucial for the analysis since another derived measure present in dataset could be selected instead and the feature was solely specific to the astronomy use case. In the end, we found that this feature was not used by any of the participants (including the proposer) in the user study.

10 Conclusion and Future Work

In the face of a data deluge in science, many scientists struggle to leverage these large datasets to derive scientific insights. While VQSs hold tremendous promise in accelerating data exploration, they are rarely used in practice. In this paper, we worked closely with three groups of scientists to learn about the challenges they face when working with data. We extended our VQS *zenvisage* to the point where it could be effectively used for scientific data analysis.

From cognitive walkthroughs and interviews, we learned about the challenges faced in scientific data analysis, including the lack of experimentation due to segmented workflows, and having to compare between large collections of visualizations (RQ1). Through participatory design, we identified three classes of missing interface capabilities essential for employing VQSs for facilitating insight in real scientific applications, spanning expressive querying and dynamic faceting, as well as fine-grained control and understanding, along with the ability to compose flexible workflows in an integrated manner (RQ2). Finally, our evaluation study demonstrated how these features helped accelerate scientific insights (RQ3), as well as how they fit in the context of data analysis workflows (RQ4). One such finding is that bottom-up querying (e.g., drag-and-drop) is preferred over top-down (e.g., sketching) for exploratory data analysis, contrary to what is commonly supported in existing VQSs. Scientists were able to use *zenvisage* for debugging errors in their data, for discovering desired patterns and trends, and for obtaining scientific insights to address unanswered research questions. By extending and evaluating VQSs to support real data analysis workflows across multiple scientific domains, we believe this work can serve as a roadmap for the broad adoption of VQSs in data analysis.

References

[1] Google correlate. `https://www.google.com/trends/correlate/draw`, 2016. Accessed: August 30, 2016.

[2] Zillow. `https://www.zillow.com`, 2016. Accessed: February 1, 2016.

[3] Tableau. `https://www.tableau.com`, 2017. Accessed: August 5, 2017.

[4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 111–117. IEEE, 2005. doi: 10.1109/INFOVIS.2005.24

[5] C. R. Aragon, S. S. Poon, G. S. Aldering, R. C. Thomas, and R. Quimby. Using visual analytics to maintain situation awareness in astrophysics. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pp. 27–34. IEEE, 2008. doi: 10.1088/1742-6596/125/1/012091

[6] Austin Nothaft et al. Rethinking Data-Intensive Science Using Scalable Analytics Systems. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 631–646, 2015. doi: 10.1145/2723372.2742787

[7] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1253–1260, Nov 2008. doi: 10.1109/TVCG.2008.117

[8] N. C. Chen, S. Poon, L. Ramakrishnan, and C. R. Aragon. Considering Time in Designing Large-Scale Systems for Scientific Computing. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*, pp. 1533–1545, 2016. doi: 10.1145/2818048.2819988

[9] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 443–452. ACM, 2012. doi: 10.1145/2207676.2207738

[10] Ciolfi et al. Articulating Co-Design in Museums: Reflections on Two Participatory Processes. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*, pp. 13–25, 2016. doi: 10.1145/2818048.2819967

[11] M. Correll and M. Gleicher. The semantics of sketch: Flexibility in visual query systems for time series data. In *Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on*, pp. 131–140. IEEE, 2016. doi: 10.1109/VAST.2016.7883519

[12] P. Craig and J. Kennedy. Coordinated graph and scatter-plot views for the visual exploration of microarray time-series data. *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pp. 173–180, 2003. doi: 10.1109/INFVIS.2003.1249023

[13] Y. Demchenko, P. Grosso, and P. Membrey. Addressing Big Data Issues in Scientific Data Infrastructure. *Collaboration Technologies and Systems (CTS), 2013 International Conference on (pp. 48-55). IEEE.*, pp. 48–55, 2013. doi: 10.1109/CTS.2013.6567203

[14] Drlica Wagner et al. Dark Energy Survey Year 1 Results: Photometric Data Set for Cosmology. 2017.

[15] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 489–500. ACM, 2015. doi: 10.1145/2807442.2807478

[16] B. S. Gloss, B. Signal, S. W. Cheetham, F. Gruhl, D. C. Kaczorowski, A. C. Perkins, and M. E. Dinger. High resolution temporal transcriptomics of mouse embryoid body development reveals complex expression dynamics of coding and noncoding loci. *Scientific Reports*, 7(1):6731, 2017. doi: 10.1038/s41598-017-06110-5

[17] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 315–324. ACM, 2009. doi: 10.1145/1502650.1502695

[18] J. D. Gould and C. Lewis. Designing for usability—key principles and what designers think. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 28(3):50–53, 1983. doi: 10.1145/800045.801579

[19] J. Heer and B. Shneiderman. A taxonomy of tools that support the fluent and flexible use of visualizations. *Interactive Dynamics for Visual Analysis*, 10:1–26, 2012. doi: 10.1145/2133416.2146416

[20] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004. doi: 10.1057/palgrave.ivs.9500061

[21] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372. ACM, 2011. doi: 10.1145/1978942.1979444

[22] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 547–554. ACM, 2012. doi: 10.1145/2254556.2254659

[23] M. Kersten, S. Idreos, S. Manegold, and E. Liarou. The Researcher's Guide to the Data Deluge: Querying a Scientific Database in Just a Few Seconds. *Proceedings of the VLDB Endowment,*, p. 1474, 2011. doi: 10.1145/1409360.1409380

[24] M. B. Kery, A. Horvath, and B. A. Myers. Variolite: Supporting exploratory programming by data scientists. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1265–1276, 2017. doi: 10.1145/3025453.3025626

[25] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 681–684. ACM, 2012. doi: 10.1145/2213836.2213931

[26] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, 2012. doi: 10.1109/TVCG.2011.279

[27] G. G. Méndez, U. Hinrichs, and M. Nacenta. Bottom-up vs . Top-down : Trade-offs in Efficiency , Understanding , Freedom and Creativity with InfoVis Tools. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2017. doi: 10.1145/3025453.3025942

[28] T. M. Michael Sedlmair, Miriah Meyer. Design Study Methodolgy: Reflections from the Trenches and the Stacks. 1(12):2431–2440, 2012.

[29] D. C. Miller, N. J. Salkind, and D. C. Miller. *Handbook of research design and social measurement*. SAGE, 2002.

[30] Mohebbi et al. Google correlate whitepaper. 2011.

[31] I. Momcheva and E. Tollerud. Software use in astronomy: an informal survey. *arXiv preprint arXiv:1507.03989*, 2015.

[32] M. J. Muller and S. Kuhn. Participatory design. *Commun. ACM*, 36(6):24–28, June 1993. doi: 10.1145/153571.255960

[33] T. Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6), 2009. doi: 10.1109/TVCG.2009.111

[34] J. Nielsen. Usability inspection methods. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, pp. 413–414. ACM, New York, NY, USA, 1994. doi: 10.1145/259963.260531

[35] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. doi: 10.1109/MCG.2006.70

[36] Patel et al. Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User Interface Software and Technology*, pp. 37–46. ACM, 2010. doi: 10.1145/1866029.1866038

[37] P. C. Peng and S. Sinha. Quantitative modeling of gene expression using dna shape features of binding sites. *Nucleic Acids Research*, 44(13):e120, 2016. doi: 10.1093/nar/gkw446

[38] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, vol. 5, pp. 2–4, 2005.

[39] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 109–116. ACM, 2004. doi: 10.1145/989863.989880

[40] S. S. Poon, R. C. Thomas, C. R. Aragon, and B. Lee. Context-linked virtual assistants for distributed teams: an astrophysics case study. In *Proceedings of the 2008 ACM Conference on Computer supported Cooperative Work*, pp. 361–370. ACM, 2008. doi: 10.1145/1460563.1460623

[41] Prabhu et al. A survey of the practice of computational science. In *State of the Practice Reports*, SC '11, pp. 19:1–19:12. ACM, New York, NY, USA, 2011. doi: 10.1145/2063348.2063374

[42] Ryall et al. Querylines: approximate query for visual browsing. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1765–1768. ACM, 2005. doi: 10.1145/1056808.1057017

[43] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A Natural Language Interface for Visual Analysis. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, pp. 365–377, 2016. doi: 10.1145/2984511.2984588

[44] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994. doi: 10.1109/52.329404

[45] B. Shneiderman. Creativity Support Tools: Accelerating Discovery and Innovation. *Communications of the ACM*, 50(12):20–32, 2007. doi: 10.1145/1323688.1323689

[46] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pp. 1–7. ACM, 2006. doi: 10.1145/1168149.1168158

[47] T. Siddiqui, J. Lee, A. Kim, E. Xue, C. Wang, Y. Zou, L. Guo, C. Liu, X. Yu, K. Karahalios, and A. Parameswaran. Fast-Forwarding to Desired Visualizations with zenvisage. 2017. doi: 10.1145/1235

[48] Siddiqui et al. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *Proceedings of the VLDB Endowment*, 10(4):457–468, 2016. doi: 10.14778/3025111.3025126

[49] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment*, 8(13):2182–2193, 2015. doi: 10.14778/2831360.2831371

[50] M. Wattenberg. Sketching a graph to query a time-series database. In *CHI'01 Extended Abstracts on Human factors in Computing Systems*, pp. 381–382. ACM, 2001. doi: 10.1145/634067.634292

[51] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191

[52] Wongsuphasawat et al. Voyager 2 : Augmenting Visual Analysis with Partial View Specifications. 2017. doi: 10.1145/3025453.3025768

[53] J. S. Yi, Y.-A. Kang, J. T. Stasko, and J. A. Jacko. Understanding and characterizing insights: How Do People Gain Insights Using Information Visualization? *Proceedings of the 2008 conference on BEyond time and errors novel evaLuation methods for Information Visualization - BELIV '08*, p. 1, 2008. doi: 10.1145/1377966.1377971

[54] E. Zgraggen, S. M. Drucker, D. Fisher, and R. DeLine. (slqu)eries: Visual Regular Expressions for Querying and Exploring Event Sequences. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2683–2692, 2015. doi: 10.1145/2702123.2702262