

You can't always sketch what you want: Understanding Sensemaking in Visual Query Systems

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

Abstract— Visual query systems (VQSs) empower users to interactively search for line charts with desired visual patterns, typically specified using intuitive sketch-based interfaces. Despite decades of past work on VQSs, these efforts have not translated to adoption in practice, possibly because VQSs are largely evaluated in unrealistic lab-based settings. To remedy this gap in adoption, we collaborated with experts from three diverse domains—astronomy, genetics, and material science—via a year-long **user-centered** design process to develop a VQS that supports their workflow and analytical needs, and evaluate how VQSs can be used in practice. Our study results reveal that ad-hoc sketch-only querying is not as commonly used as prior work suggests, since analysts are often unable to precisely express their patterns of interest. In addition, we characterize three essential sensemaking processes supported by our enhanced VQS. We discover that participants employ all three processes, but in different proportions, depending on the analytical needs in each domain. Our findings discovered the need for integrating all three sensemaking processes into the design of future VQSs to make them useful and practical, by addressing a wide range of analytical inquiries.

Index Terms—Visual analytics, exploratory analysis, visual queries

1 INTRODUCTION

Line charts are commonly employed during data exploration—the intuitive connected patterns often illustrate complex underlying processes and yield interpretable and visually compelling data-driven narratives [13]. However, discovering line charts that display certain meaningful patterns, trends, or characteristics of interest is often an overwhelming and error-prone process, consisting of manual examination of large numbers of line charts. For example, when trying to find supernovae, which exhibits a unique pattern of brightness over time (an initial peak followed by a long-tail decay), astronomers often have to manually construct and inspect thousands of line chart visualizations to find ones with their desired pattern. To address this exploration challenge, there has been a large number of papers dedicated to building *Visual Query Systems* (VQSs)—a term coined by Ryall et al. [44] to describe systems that allow users to specify and search for desired line chart patterns via visual interfaces [10, 12, 19, 21, 26, 29, 44, 50, 56]. These interfaces typically include a sketching canvas where users can draw a visual pattern of interest, with the system automatically traversing all potential visualization candidates to find those that match the specification.

While these intuitive specification interfaces were proposed as a promising solution to the problem of painful manual exploration of visualizations for time-series analysis [44, 56], to the best of our knowledge, VQSs have not lived up to these expectations and are not very commonly used in practice. One likely reason for the lack of VQS adoption may be attributed to how prior work **has** focused almost exclusively on optimizing the pattern-matching algorithms and interactions, with few invested in understanding actual user needs and how VQSs can be used for solving real-world problems. *Our paper seeks to understand how VQSs can actually be used in practice, as a first step towards the broad adoption of VQSs in data analysis.* Unlike prior work on VQSs, we set out to not only evaluate VQSs in-situ on real problem domains, but also involve participants from these domains in the VQS design. We present findings from a series of interviews, contextual inquiry, participatory design, and user studies with scientists from three different domains—*astronomy, genetics, and material science*—over the course of a year-long collaboration. The amount of time we invested in each of these three diverse domains surpasses the norm in this field and is key to uncovering the insights presented in this paper. These domains were selected to capture a diverse set of goals and datasets wherein VQSs can help address important scientific questions, 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?

In this work, we adapt methods from user-centered design

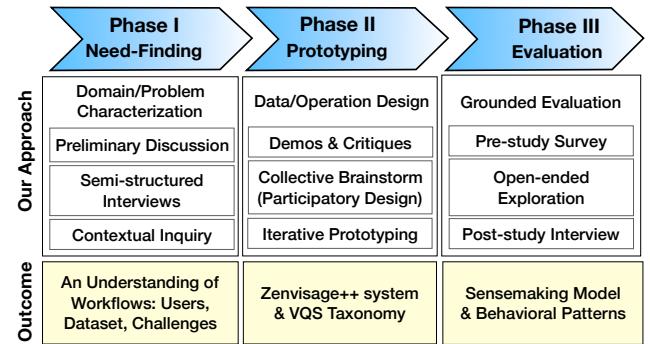


Fig. 1: Lifecycle model summarizing our research approach and the outcome of each phase.

(UCD) [15,33,34], such as interviews, contextual inquiry, and participatory design, into our design-implementation-evaluation cycle [47]. Via contextual inquiry and interviews, we first identified challenges in existing data analysis workflows in these domains that could be potentially addressed by a VQS. Building on top of an existing open-source VQS, Zenvisage [50,51], we iterated on the design of the VQS with participants over the course of a year to better compose data exploration workflows that lead to insight discovery. Rather than targeting a domain-specific solution, we engaged with multiple domains to observe differences and commonalities across domains and synthesize high-level insights regarding the use of VQSs. While designing and performing this multi-phased, mixed-methods research agenda across three diverse use cases was challenging, this endeavor was necessary for addressing the qualitative, participant-centered research questions investigated.

We organize our **design study** findings into a taxonomy of VQS capabilities, involving three sensemaking processes inspired by Pirolli and Card's notional model of analyst sensemaking [40]. The sensemaking processes include *top-down pattern search* (translating a pattern “in-the-head” into a visual query), *bottom-up data-driven inquiries* (querying or recommending based on data), and *context-creation* (navigating across different collections of visualizations). We find that prior VQSs have focused on enabling top-down processes (via sketching capabilities), but have largely overlooked the two other processes that we found to be essential in all three domains. These missing capabilities partially explain why prior VQSs have not been widely adopted in practice.

To study how various VQSs are used in practice, we conducted a final evaluation study with nine participants using our final VQS prototype to address their research questions on their own datasets. During this study, participants gained novel scientific insights, such as identifying a star that was known to harbor a Jupiter-sized planet, discovering a previously-unknown relationship between solvent properties, and

• All authors are with University of Illinois, Urbana-Champaign.

• E-mail: {jlee782, lee98, tsiddiq2, jkim475, kkarahal, adityagp}@illinois.edu

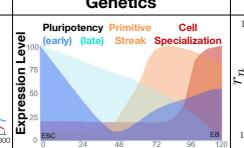
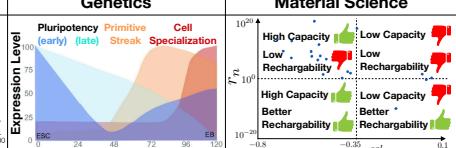
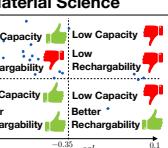
	Astronomy	Genetics	Material Science
Desired Insights			
Challenges	Discover rare astronomical objects with specific pattern properties in a large dataset containing noisy, non-uniform time series data.	Understand characteristic profiles amongst a large number of genes that can rise and peak at different time points .	Identify battery candidates from a large, noisy, multidimensional dataset by comparing functional relationships and tradeoffs between multiple attributes .

Fig. 2: Desired insights, problem and dataset challenges for each of the three application domains in our study.

finding characteristic gene expression profiles confirming the results of a related publication.

During this evaluation study, we were somewhat surprised to discover that sketching a pattern for querying is often ineffective on its own. This is due to the fact that sketching makes the assumption that users know the pattern that they want to sketch and are able to sketch it precisely. However, this is typically not the case in practice. For example, the geneticists from our study often did not have a preconceived knowledge of what to sketch for and relied heavily on VQS-recommended common and outlying patterns to jumpstart their queries. Likewise, while the material scientists from our study were interested in datapoints that fall within specific value-ranges, they did not have an apriori notion of what their desired patterns would look like. Overall, participants typically opted to combine sketching with other means of pattern specification—one common mechanism was to drag-and-drop a recommended pattern onto the canvas, and then modify it (e.g., by smoothing it out).

To further understand how participants engaged with VQSs in their analytical workflows, we constructed a Markov model to characterize how participants transitioned between different sensemaking processes during their analysis. We found that participants often constructed a diverse set of analytical workflows tailored to their domains by focusing around a primary sensemaking process, while iteratively interleaving their analysis with the two other processes. This finding points to how all three sensemaking processes, along with seamless transitions between them, are crucial for enabling the effective use and adoption of VQSs for addressing real-world challenges.

To the best of our knowledge, our study is the *first to holistically examine how VQSs can be designed to fit the needs of real-world analysts, and how they are actually used in practice*. Working with participants from multiple domains enabled us to compare the differences and commonalities across different domains, thereby identifying general VQS challenges and requirements for supporting common analytical goals, as illustrated in Figure 2. Our contributions include:

- a characterization of the problems addressable by VQSs through design studies with three different domains,
- a taxonomy of essential VQSs capabilities, leading to a sensemaking model for VQSs,
- an integrative VQS, *zenvisage++* capable of facilitating rapid hypothesis generation and insight discovery, resulting from iteration with end-users,
- study findings on how VQSs are used in practice, leading to the development of a novel sensemaking model for VQSs.

Our work not only opens up a new space of opportunities beyond the narrow use cases considered by prior studies, but also advocates common design guidelines and end-user considerations for building next-generation VQSs.

2 RELATED WORK

We will now describe past work in visual query systems and existing evaluation methods of visualization systems to provide background and motivation for our work.

Visual Query Systems: Definition and Brief Survey

The term *visual query system* (VQS) was introduced by Ryall et al. [44] and Correll and Gleicher [10] to describe systems that enable analysts to directly search for **line chart** visualizations matching a queried pattern, constructed through a visual specification interface. Examples of such systems include TimeSearcher [18, 19], where the query specification mechanism is a rectangular box, with the tool filtering out all of the **line charts** that do not pass through it, and QuerySketch [56] and Google Correlate [29], where the query is sketched as a pattern on canvas, with the tool filtering out all of the **line charts** that have a different shape. Subsequent work, including TimeSketch [12], SketchQuery [10], and Qetch [26], recognized the ambiguity in sketching by studying how humans rank similarity in patterns. Finer-grained specification interfaces and pattern-matching algorithms have also been developed to improve the expressiveness of sketched queries and clarify how a sketch should be interpreted. These VQSs include QueryLines [44] where queries can be flexibly composed of soft constraints and preferences and SoftSelect [21] where users can vary the level of sketch similarity across a search pattern. Beyond sketching, Zenvisage [50, 51], SketchQuery, and TimeSearcher allow users to submit an existing visualization as the query, either via drag-and-drop or double-clicking on the existing visualization. In our work, we built on our existing system, Zenvisage, since it was open-source, extensible, and included features beyond the **pattern match** specification typically found in existing systems, such as the ability to add data filters and examine recommended patterns [51].

Design and Evaluation Methodologies for Visualization Systems

Visualization systems are typically evaluated via in-lab usability studies or controlled studies against existing visualization baselines [35, 41, 58]. However, successful lab-tested systems do not always translate to community acceptance and adoption. For instance, while decades of work have shown VQSs to be effective in controlled lab studies, they have not gained widespread adoption. The unrealistic nature of controlled studies has prompted the visualization research community to develop more participant-centered, ethnographic approaches for understanding how analysts perform visual data analysis and reasoning [25, 32, 41, 46, 49]. For example, multi-dimensional, in-depth, long-term case studies (MILCs) combine interviews, surveys, logging, and other empirical artifacts to create a holistic understanding of how a visualization system can be used in its intended environment [49].

In the VQS literature, even though the development and evaluation of advanced VQS algorithms and interactions has been well underway for many years, prior work has yet to characterize and understand the needs of target users and observe how VQSs may be used as part of a real-world workflow, in order to address the initial questions of: 1) whether the problems that VQSs aim to address are even the right ones to address and 2) whether the chosen operations actually solve the user's problems. In the context of Munzner's nested model for visualization design and evaluation [32], this gap between research and adoption stems from the common "*downstream threat*" of jumping prematurely into the deep levels of *encoding, interaction, or algorithm design*, before a proper *domain problem characterization* and *data/operation abstraction design* is performed. Our work fills this crucial gap in the existing literature and highlights how incorrect assumptions adopted by most prior work in this space regarding the first two stages of Munzner's model may have led to the present-day failures in VQS adoption.

We performed design studies [25, 46, 49] with three different subject areas for *domain problem characterization* by adopting **user-centered design practices**. **User-centered design (UCD)** [15, 33, 34] is a class of techniques for iteratively designing a product that fits the needs and desires of users. In UCD, users convey their needs to inform design decisions. Through participatory design (PD) [31, 45], we engaged potential stakeholders as active co-designers early on and during every step of the design process, in order to develop a system that they may eventually adopt in their analytical workflows. Participatory design is a well-established UCD approach in the CHI and CSCW communities and has been successfully applied to develop systems for visual analytics [2, 8], tangible museum experiences [9], and scientific collaborations [7, 42].

In order to “[develop] a system model that will support [the] user’s work” that subsequently “fosters participatory design”, Holzblatt and

Jones [20] describe contextual inquiry as a technique where researchers observe **participants** in their own work environment. Likewise, we first perform **contextual inquiry** and interviews with participants to understand their research questions and the challenges associated with their existing analytical workflows, and to identify design opportunities for VQSs. To better understand how VQSs can be used *in-situ* in participant's existing **workflows**, we regularly gathered feedback from participants and collaboratively envisioned potential designs by demonstrating preliminary versions of our prototype *zenvisage++*. Based on our design findings, we contribute to the *data/operation abstraction design* of VQSs in Munzner's model by developing a taxonomy for characterizing how analysts make use of VQSs to accomplish their analytical tasks. Finally, we validated our abstraction design with grounded evaluation [22, 41], where participants were invited to bring in their own datasets and research problems that they have a vested interest in to test our final deployed system.

3 METHODS

Via interviews and contextual **inquiry** in participants' normal work environments, we first identified the needs and challenges in participants' existing data analysis workflows. Given these challenges, we collaboratively designed VQS functionalities by engaging with experts from three different domains **throughout the design process**, leading to a final prototype *zenvisage++*. After the design phase, we conducted an evaluation study to understand how VQSs are used in the real-world analytical workflows. Our research methodology is illustrated in Figure 1; we now describe the study procedure in more detail.

3.1 Phase I: Need-finding

We recruited participants by reaching out to research groups who have experienced challenges in data exploration, via email and word-of-mouth. Based on early conversations with analysts from 12 different potential application areas, we narrowed down to three use cases in astronomy, genetics, and material science **through a process similar to the “winnow” stage in Sedlmair et al. [46]**. The domains were chosen based on their suitability for VQSs as well as diversity in use cases. Six scientists, with extensive research experience in their respective fields, participated in the design process. We interviewed participants to learn about their dataset and research questions, shadowed participants in conducting their existing analysis workflows, and subsequently discussed the needs and challenges of their use cases. The interviews were semi-structured and focused on how the analytical tasks in their workflows relate to the scientific questions they were interested in.

3.2 Phase II: Collaborative Prototyping

For **iterative prototyping**, we built on top of an existing open-source VQS, Zenvisage [50,51], to create a functional prototype to showcase the capabilities of VQSs. The use of functional prototypes is a common and effective way of engaging with participants, by providing a starting point for collaborative design [9]. We collaborated with each team closely with approximately two 1-hour-long meetings per month, where we learned more about their datasets, objectives, and what additional VQS functionalities could help address their research questions. During these meetings, we collectively brainstormed with participants on the design of the prototype. Participants also had the opportunity to interact with the prototype through the help of a guided facilitator. Through these exercises, we elicited feedback from participants on how the VQS could better support their scientific goals and identified and incorporated several crucial capabilities into *zenvisage++*.

3.3 Phase III: Grounded Evaluation

After the prototyping phase, we performed a qualitative evaluation to study how analysts interact with different VQS components in practice. Participants used datasets that they have a vested interest in exploring to address unanswered research questions (a total of six different datasets across nine participants). As shown in Table 1, the evaluation study participants included the six scientists from Phase I and II, along with three additional “blank-slate” participants who had never encountered *zenvisage++* before. The use of all or a subset of the project stakeholders

	ID	Dataset	Participated in Design	Position	Years of Experience	Dataset Familiarity
Genetics	A1	DES	✓	Researcher	10+	3
	A2	Kepler		Postdoc	8	5
	A3	Kepler		Postdoc	8	5
Astronomy	G1	Mouse	✓	Grad Student	4	4
	G2	Cancer		Grad Student	2	2
	G3	Mouse	✓	Professor	10+	2
Material Science	M1	Solvent (8k)	✓	Postdoc	4	5
	M2	Solvent (Full)	✓	Professor	10+	5
	M3	Solvent (Full)	✓	Grad Student	3	5

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

as evaluation participants is typical in participatory design [6]. While the small sample size of participants may be viewed as a limitation, this is a pervading challenge when recruiting domain-experts [3, 27]. Nevertheless, as Sedlmair et al. [46] argue, even studies with a small group of domain experts involved are invaluable for understanding expert needs.

Evaluation study participants were recruited from each of the three aforementioned research groups, as well as domain-specific mailing lists. Prior to the study, we asked potential participants to fill out a pre-study survey to determine eligibility. Eligibility criteria included: being an active researcher in the subject area with more than one year of experience, and having worked on a research project involving data of the same nature used in the design phase.

At the start of the **in-lab evaluation** study, participants were provided with an interactive walk-through of *zenvisage++* and given approximately ten minutes for a guided exploration of a preloaded real-estate example dataset. After familiarizing themselves with the tool, we loaded the participant's dataset and encouraged them to talk-aloud during data exploration, and use external tools or other resources as needed. If the participant was out of ideas, we suggested one of the main VQS functionalities that they had not yet used. If any of these operations were not applicable to their specific dataset, they were allowed to skip the operation after having considered it. The user study lasted for about an hour and ended after they covered all the main functionalities. After the study, we asked participants open-ended questions about their experience.

4 CURRENT PARTICIPANT WORKFLOWS AND OPPORTUNITIES

In this section, we describe our study participants, their scientific goals, and their preferred analysis workflows, based on Phase I of our study. While we collaborated with each application domain in depth, we focus on the key findings from each domain to highlight their commonalities and differences, in order to provide a backdrop for our VQS findings described later on. Comparing and contrasting between the diverse set of questions, datasets, and challenges across these three use cases revealed new cross-disciplinary insights essential to better understand how VQSs can be extended for novel and unforeseen use cases.

4.1 Astronomy

Participants and Goals:

The Dark Energy Survey (DES) is a multi-institution project that surveys 300 million galaxies over 525 nights to study dark energy [11]. The telescope used to survey these galaxies also focuses on smaller patches of the sky on a weekly interval to discover astronomical transients, i.e., objects whose brightness changes dramatically as a function of time, such as supernovae or quasars. Their dataset consisted of a large collection of *light curves*: brightness observations over time, one associated with each astronomical object, plotted as a time series. Over five months, we worked closely with A1, an astronomer on the project's data management team at a supercomputing facility. Their scientific goal was to identify potential astronomical transients in order to study their properties, i.e., identify patterns in line charts.

Existing Workflow and Design Opportunities:

Since astronomical datasets are often terabytes in scale, they are often processed and stored in highly specialized data management systems

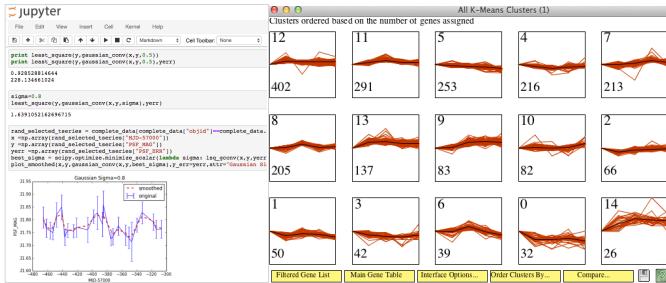


Fig. 3: Screenshots from contextual inquiry. Left: A1 performs data smoothing to clean the data and then examines a light curve manually using a Jupyter notebook. Right: G2 uses a domain-specific software to perform clustering and visualize the outputs.

in supercomputing centers. First, the astronomer downloads a data sample to explore in a Jupyter notebook, performs data cleaning and wrangling, and verifies data fidelity by computing a set of relevant statistics. To identify transients, the primary scientific goal of their exploration, the astronomer programmatically generates visualizations of candidate objects with `matplotlib` and visually examines each light curve. Figure 3 (left) shows an example of such a manually-generated light curve. If an object of interest is identified through visual analysis, the astronomer may inspect the image of the object for verifying that the significant change in brightness was not due to an imaging artifact. While an experienced astronomer who has examined many transient light curves can often distinguish an interesting transient object from noise by sight, manual searching for transients is time-consuming and error-prone, since the large majority of objects are false-positives. A1 was interested in VQSSs as he recognized how specific pattern search queries could help astronomers directly search for these rare transients without cumbersome manual examination.

4.2 Genetics

Participants and Goals:

Gene expression is a common measurement in genetics obtained via microarray experiments [38]. We worked with a graduate student (G1) and professor (G3) at a research university who were using gene expression data to understand how genes are related to phenotypes expressed during early embryonic development. Their data consisted of a collection of gene expression profiles over time for mouse stem cells, aggregated over multiple experiments. Their scientific goal was to correlate gene functionalities with their expression profiles by gaining an overview and summary of patterns amongst the gene expression profiles, i.e., line charts.

Existing Workflow and Design Opportunities:

G1 downloads the raw microarray data from a public database and preprocesses the data using a script written in R. Then, to explore this data, G1 loads the preprocessed gene expression data into a custom desktop application to visualize and cluster the gene expression profiles, as shown in Figure 3 (right). Prior to the study, G1 and G3 spent over a month searching for the optimal number of groups to cluster the profiles, by iteratively tuning the parameters on the clustering application and evaluating the output via a mix of application-provided visualizations and programmatically-generated statistics. 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, this is, they could only test out a few variations per week. When we first demonstrated the capabilities of a VQS to in our introductory meeting, G3 was very surprised that upon changing the cluster size, the recommended visualizations updates almost instantaneously. They were interested in VQSSs, since the tool had the potential to dramatically speed up their collaborative analysis process.

4.3 Material Science

Participants and Goals:

We collaborated with material scientists at a research university who identify solvents for energy-efficient and safe batteries. These scientists worked on a large simulation dataset containing chemical properties

for more than 280,000 solvents [24]. Each row of their dataset corresponded to a unique solvent with 25 different chemical attributes. We worked closely with a postdoctoral researcher (M1), professor (M2), and graduate student (M3) to design a sensible way of exploring their data. They wanted to use VQSSs to discover solvents that not only have similar properties to known solvents, but are also more favorable (e.g., cheaper or safer to manufacture). To search for these desired solvents, they needed to understand how changes in certain chemical attributes affect other properties, i.e. trends in line charts, manifests under specific conditions.

Existing Workflow and Design Opportunities:

M1 typically starts his data exploration process by applying filters to a list of potential battery solvents using SQL queries (e.g., find solvents with boiling point over 300 Kelvins and lithium solvation energy under 10 kcal/mol). By iteratively applying and adjusting different (often complementary) sets of filters, he compares between different groups of solvents by observing their properties across a small sample. He manually examines the properties of each individual solvent by inspecting the 3D chemical structure of the solvent in a custom software, as well as gathering information regarding the solvent by cross-referencing an external chemical database and existing uses of this solvent in literature. The collected information, including cost, availability, and other physical properties, enabled researchers to select the final set of desirable solvents that could be feasibly experimented with in their lab. While M1 could identify potential solvents through manual lookups and comparisons, he was interested in VQSSs since it was impossible to uncover hidden relationships between different attributes, such as how changes in one attribute affects the behavior of other properties of a solvent, across large numbers of solvents manually.

4.4 Themes Emerging From Need-finding Phase

Across the domains, several themes emerged around the bottlenecks that participants experienced in existing workflows.

- Need for Expressive Querying:** There is often a need to compare among large numbers of data instances, yet it is difficult to express and search for a desired shape-based pattern through standard programming languages such as SQL or Python. And yet, none of the participants have heard of VQSSs, let alone use them.
- Need for Integrative Workflows:** Users often switched between different analytical tasks, including preprocessing, parameter specification, code execution, and visualization comparisons. The non-interactive nature of their segmented workflows impedes exploratory data analysis and hinders collaboration by incurring a high cognitive cost to task-switching.
- Need for Faceted Exploration:** To deal with the large volume of data present, users have to select particular samples or subsets of data that are “worth investigating”. Often, the choice of what criteria to apply as filters is also exploratory.

These themes seeded the collaborative feature discovery process, leading to the iterative development of the system prototype, as described next.

5 DESIGN PROCESS AND SYSTEM OVERVIEW

Given the need for a VQS highlighted via contextual inquiry, we further collaborated with participants to develop features to address their problems and challenges in Phase II of our study. In this section, we first provide a high-level system overview of the design product, `zenvisage++`, then we reflect on our feature discovery process to introduce our design findings.

5.1 System Overview

Table 2 highlights the major capabilities relevant to the study findings that were incrementally incorporated and improved over time, leading to our final prototype, `zenvisage++`. Details of other features in `zenvisage++` can be found in the appendix and online documentation¹. The `zenvisage++` interface is organized into 5 major regions all of which dynamically update upon user interactions. Typically, participants begin

¹Documentation: <http://github.com/zenvisage/zenvisage/wiki>

	Component	Feature	Purpose	Task Example	Similar Features in Past VQSS
Top-Down	Pattern Specification: <i>What is the shape of the pattern query?</i>	Query by Sketch (Figure 4B1)	Freehand sketching for specifying pattern query.	A: Find patterns with a peak and long-tail decay that may be supernovae candidates.	All include sketch canvas except [19].
		Input Equation (Figure 4A1)	Specify a exact functional form as a pattern query (e.g., $y=x^2$).	M: Find patterns exhibiting inversely proportional chemical relationship.	—
		Pattern Upload (Figure 4D2)	Upload a pattern consisting of a sequence of points as a query.	A: Find supernovae based on previously discovered sources.	Upload CSV [29]
		Smoothing (Figure 4D2)	Interactively adjusting the level of denoising on visualizations, effectively changing the degree of shape approximation when performing pattern matching.	A, M: Eliminate patterns matched to spurious noise.	Smoothing [26] Angular slope queries [19] Trend querylines [44]
	Match Specification: <i>How should the pattern query be matched with other visualizations?</i>	Range Selection (Figure 4B2, D4)	Restrict to query only in specific x/y ranges of interest through brushing selected x-range and filtering selected y-range.	A: Matching only around shape exhibiting a peak. M: Matching only around shape region that exhibit linear or exponential relationships	Text Entry [26, 56] Min/max boundaries [44] Range Brushing [18]
		Range Invariance (Figure 4D1,4)	Ignoring vertical or horizontal differences in pattern matching through option for x-range normalization and y-invariant similarity metrics .	A: Searching for existence of a peak above a certain amplitude. G: Searching for a "generally-rising" pattern.	Temporal invariants [10]
		Data selection (Figure 4A)	Changing the collection of visualizations to iterate over.	M: Explore tradeoffs and relationships between physical attributes.	—
		Display control (Figure 4D4)	Changing the details of how visualizations should be displayed.	M: Non-time-series data should be displayed as scatterplot.	—
Context Creation	View Specification: <i>What data to visualize and how should it be displayed?</i>	Filter (Figure 4D3)	Display and query only on data that satisfies the composed filter constraints.	A: Eliminate unlikely candidates by navigating to more probable data regions. M, G: Compare how overall patterns change when filtered to particular data subsets.	—
		Dynamic Class (Figure 9)	Create custom classes of data that satisfies one or more specified range constraints. Display aggregate visualizations for separate data classes.	A, M: Examine aggregate patterns of different data classes.	—
		Drag-and-drop (Figure 4C, E)	Querying with any selected result visualization as pattern query (either from recommendations or results).	A, G, M: Find other objects that are similar to X; Examine what other objects similar to X look like overall. A: Examine anomalies and debug data errors through outliers.	Drag-and-drop [18] Double-Click [10]
		Representative and Outliers (Figure 4E)	Displaying visualizations of representative trends and outlier instances based on clustering.	G, M: Understand representative trends common to this dataset (or filtered subset).	—
Bottom-Up	Result Querying: <i>What other visualizations "look similar" to the selected pattern?</i>	Drag-and-drop (Figure 4C, E)	Querying with any selected result visualization as pattern query (either from recommendations or results).	A, G, M: Find other objects that are similar to X; Examine what other objects similar to X look like overall. A: Examine anomalies and debug data errors through outliers.	Drag-and-drop [18] Double-Click [10]
	Recommendation: <i>What are the key patterns in this dataset?</i>	Representative and Outliers (Figure 4E)	Displaying visualizations of representative trends and outlier instances based on clustering.	G, M: Understand representative trends common to this dataset (or filtered subset).	—

Table 2: Taxonomy of key capabilities essential to VQSSs and major features incorporated via user-centered design. We organize each feature based on its functional component. From left to right, each of the three sensemaking processes (first column) is broken down into key functional components (second column) in VQSSs. Each component addresses a pro-forma question from the system’s perspective. Table cells are further colored according to the sensemaking process that each component corresponds to (Blue: Top-down, Yellow: Context creation, Green: Bottom-up). We list the functional purpose of each feature based on how it is implemented in *zenvisage++*, example use cases from participatory design (**A**: astronomy, **M**: material science, **G**: genetics), and similar features incorporated in past VQSSs. Given the exhaustive nature of Table 2, each motivated by example use cases from one or more domains, we further organize the features in terms of the Section 6 sensemaking framework and assess their effectiveness in the Section 7 evaluation study.

their analysis by selecting the dataset and attributes to visualize in the *data selection panel* (Figure 4A). Then, they specify a pattern of interest as a query (hereafter referred to as *pattern query*), through either sketching, inputting an equation, uploading a data pattern, or dragging and dropping an existing visualization, displayed on the *query canvas* (Figure 4B). *zenvisage++* performs shape-matching between the queried pattern and other possible visualizations, and returns a ranked list of visualizations that are most similar to the queried pattern, displayed in the *results panel* (Figure 4C). At any point during the analysis, analysts can adjust various system-level settings through the *control panel* (Figure 4D) or browse through the list of *recommendations* provided by *zenvisage++* (Figure 4E). For comparison, the existing Zenvisage system from [51] allowed users to query via sketching or drag-and-drop and displayed representative and outlier pattern recommendations, but had limited capabilities to navigate across different data subsets and had few control settings. Our *zenvisage++* system is open source and

available at: <http://github.com/zenvisage/zenvisage>.

5.2 The Collaborative Feature Discovery Process

Throughout the *design* process, we worked closely with participants to discover VQS capabilities that were essential for addressing their high-level domain challenges. We identified various subtasks based on the participant’s workflows, designed sensible features for accomplishing these subtasks that could be used in conjunction with existing VQS capabilities, and elicited feedback on intermediate feature prototypes. Bodker et al. [5] cite the importance of encouraging user participation and creativity in cooperative design through different techniques, such as future workshops, critiques, and situational role-playing. Similarly, our objective was to collect as many feature proposals as possible, while being inclusive across different domains. We further organized these features into Table 2 through an iterative coding process [30] by one of the authors.

We first collected the list of features, example usage scenarios, and similar capabilities in existing VQSSs as open codes, corresponding to individual rows in Table 2. Then, we further organized this list into axial codes representing “components”: core functionalities essential to VQSSs (second column in Table 2). Finally, the selective codes capture each of the sensemaking processes (leftmost column in Table 2). Instead of describing this table in detail, we present a typical example of how this table is organized. From right to left, consider the row corresponding to the Smoothing feature in Table 2: one of the common challenges in astronomy and material science is that noise in the dataset can result in large numbers of false-positive matches. To address this issue, smoothing is a feature in *zenvisage++* that enables users to adjust data smoothing algorithms and parameters on-the-fly to both denoise the data and change the degree of shape approximation applied to all visualizations when performing pattern matching. Smoothing, along with range selection and range invariance, is part of the *match specification* component: VQS mechanisms for clarifying how matching should be performed. Both match specification and *pattern specification* (a description of what the pattern query should look like) are essential components for supporting the sensemaking process top-down pattern search (in blue, as labeled in the leftmost column).

Our collaboration with participants is illustrated in Figure 7

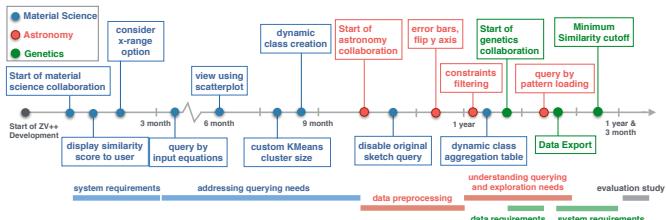


Fig. 4: Timeline for progress in participatory design studies.

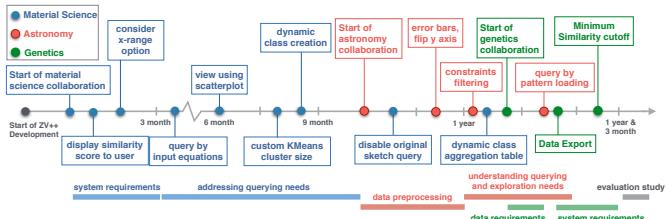


Fig. 5: Timeline for progress in participatory design studies.

It is important to note that while some of the proposed features in Table 2 (such as data filtering and view specification) are pervasive in general visual analytics (VA) systems [1, 17], they have not been incorporated in present-day VQSSs. In fact, one of the key contributions of our work is recognizing the need for an *integrative* VQS whose sum is greater than its parts, that encourages analysts to rapidly generate hypotheses and discover insights by facilitating all three sensemaking processes. This finding is partially enabled by the unexpected benefits that come with collaborating with multiple groups of participants during the feature discovery process, described next.

Given the highly-evolving, ad-hoc nature of exploratory data analysis [23, 54], our collaborative feature discovery approach for aiding such analysis comes with its advantages and limitations. We distill these experiences into four separate themes to inform future visual analytics research that adopts a user-centered design approach. **Parallel domain engagement results in cross-pollination and generalization of feature design.** One such advantage is that introducing the newly-added features to *zenvisage++* that addressed a particular domain often resulted in unexpected use cases for other domains. Considering feature proposals from multiple domains can also result in cross-pollination of feature designs, often leading to more generalized design choices. For example, around the same time when we spoke to astronomers who wanted to eliminate sparse time series from their search results,

our material science collaborators also expressed a need for inspecting only solvents with properties above a certain threshold. Instead of developing separate domain-specific features, data filtering arose as a crucial, common operation that was later incorporated into *zenvisage++* to support this class of queries. **The Hidden Upfront Cost in Domain-Specific Integration** While we expected to spend most of our collaborative time for designing the analytics of visual querying (such as fine-tuning what constitutes a match), the process of preparing the analyst’s dataset to be ready to use in our system was the most time-consuming aspect of the prototyping phase and required significant effort. As shown in the Figure 7 timeline, for each domain, this initial phase can largely be boiled down to a period of time for satisfying the minimum data and systems requirements for establishing how a VQS can be used for the particular 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. System requirements include features that are required for the data to be visualized appropriately (i.e., in a way that participants can make the necessary inference based on the displayed visualization), 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 inquiries after seeing their data displayed for the first time in the prototype. It is also evident in the Figure 7 timeline that the amount of time it took to satisfy the system and data requirements decreases as we progressed to the later domains, due to how later use cases leveraged existing features in the prototype to satisfy some of these requirements. In summary, it is important to allocate sufficient time when working with real-world datasets and users, as well as developing generalizable integration features (such as data uploading tools) that could generalize across use cases to decrease the upfront cost for future collaborations with new domains. **Failure to select the right problems and features during the design phase can result in wasted engineering efforts**

While our collective brainstorming led to the cross-pollination and generalization of features, this technique can also lead to unnecessary features that result in wasted engineering effort. During co-design, there were numerous features proposed by participants, not all of which were incorporated. Based on our meeting logs with participants, we found that reasons for not carrying a feature from the design to implementation stage included:

- **Nice-to-haves:** One of the most common reasons for unincorporated features comes from participant’s requests for nice-to-have features. We use two criteria (necessity and generality across domains) to judge whether to invest in developing a particular feature.
- **“One-shot” operations:** We decided not to include features that only needed to be performed once and remain fixed thereafter in the analysis workflow. For example, certain preprocessing operations such as filtering null values only needed to be performed once with an external tool, whereas data smoothing is a procedure that requires some degree of tuning and adjustments.
- **Substantial research or engineering effort:** Some proposed features did not make sense in the context of VQS or required 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 and algorithm that is out of the scope of our design study objective.
- **Underdeveloped ideas:** Other feature requirements came from causal 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 the scientific definition of “deficiency” in terms of brightness levels was 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 or result in feature bloat.

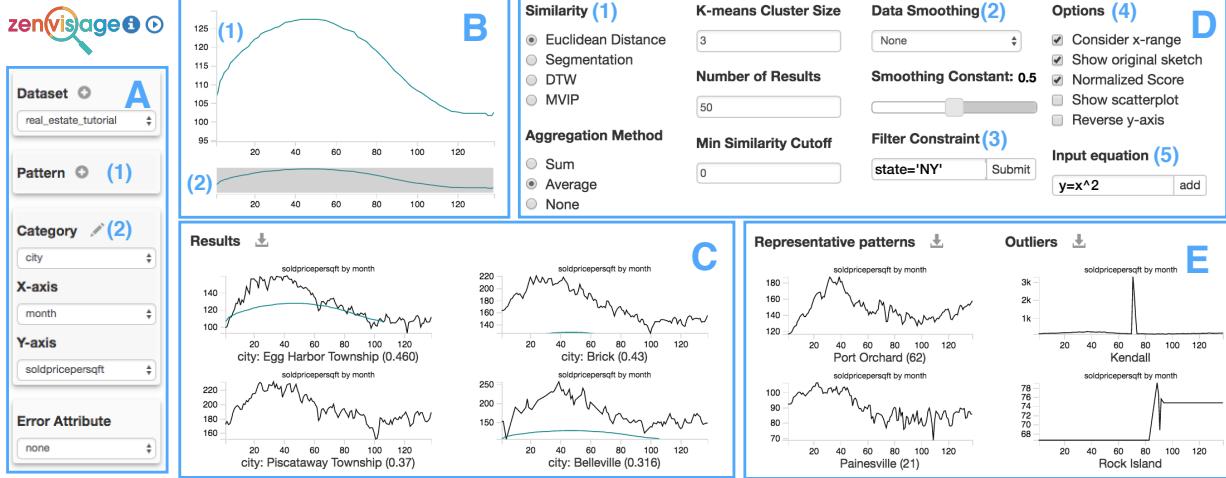


Fig. 6: The *zenvisage++* system consists of : (A) data selection panel (where users can select visualized dataset and attributes), (B) query canvas (where the queried data pattern is submitted and displayed), (C) results panel (where the visualizations most similar to the queried pattern are displayed as a ranked list), (D) control panel (where users can adjust various system-level settings), and (E) recommendations (where the typical and outlying trends in the dataset is displayed).

6 A SENSEMAKING MODEL FOR VQSs

Now that we have described our eventual prototype *zenvisage++*, we revisit Table 2 in an effort to contextualize our findings using Pirolli and Card’s sensemaking framework [40]. We demonstrate how features in *zenvisage++* address the analytical needs posed by each domain. We organize the components in Table 2 along a taxonomy of three sensemaking processes. **Pirolli and Card’s sensemaking model for expert intelligence analysis distinguishes between information processing tasks that are top-down (from theory to data) and bottom-up (from data to theory).** Correspondingly, in the context of VQSs, analysts can query either directly based on a pattern “in their head” [46] via *top-down pattern specification* or based on the data or visualizations presented to them by the system via *bottom-up data-driven inquiry*. In addition, when analysts do not know what attributes to visualize, *context creation* helps analysts navigate across different collections of visualizations to seek visualization attributes of interest. In this section, we first describe the objectives of each sensemaking process, then we discuss how each sensemaking process is comprised of functional components that address the problem and dataset characteristics of each domain.

6.1 Top-Down Pattern Search

Top-down processes are “*goal-oriented*” tasks that make use of “*analysis or re-evaluation of theories [and] hypotheses [to] generate new searches*” [40]. Applying this notion to the context of VQSs, the goal of top-down pattern search is to search for data instances that exhibit a specified pattern, based on analyst’s intuition about how the desired patterns should look like “in theory” (including visualizations from past experience or abstract conceptions based on external knowledge). Based on this preconceived notion of what patterns to search for, the design challenge is to translate the pattern query from the analyst’s head to a query executable by the VQS. This requires both components for specifying the pattern (*pattern specification*), as well as controls governing how the pattern-matching is performed (*match specification*). **Pattern Specification** interfaces allow users to submit exact descriptions of a pattern query. This is useful when the dataset contains *large numbers of potentially-relevant pattern instances*. Since it is often difficult to sketch precisely, additional shape characteristics of the pattern query (e.g., patterns containing a peak with a known amplitude, or expressible as a functional form) can be used to further winnow the list of undesired matches.

Match Specification addresses the well-known problem in VQSs where pattern queries are imprecise [10, 12, 21] by enabling users to clarify how pattern matching should be performed. Match specification is useful when the dataset is *noisy*. When the pattern query satisfies some additional constraints (e.g., the pattern is *horizontally* invariant), adjusting these knobs prune away matches that are false-positives to help analysts discover true desired candidates.

Usage Scenario: A1 knows intuitively what a supernovae pattern should look like and its detailed shape characteristics, such as the amplitude of the peak and the level of error tolerance for defining a match. He *first* performs top-down pattern search by querying for transient patterns through sketching, *then adjusts* the match criterion by choosing to ignore differences along the temporal dimension and changing the similarity metric for flexible matching.

6.2 Bottom-Up Data-Driven Inquiry

In Pirolli and Card’s sensemaking model, bottom-up processes are “*data-driven*” tasks initiated by “*noticing something of interest in data*” [40]. Likewise in VQSs, bottom-up data-driven inquiry is a browsing-oriented sensemaking process that involves tasks that are inspired by system-generated visualizations or results. The design challenge for VQSs to support bottom-up inquiries is to develop the right set of “stimuli” through *recommendations* that could provoke further data-driven inquiries, as well as low-effort mechanisms to search via these pattern instances through *result querying*. As we will discuss later, this process is crucial but underexplored in past work on VQSs. **Recommendations** display visualizations that may be of interest to users based on the current data context. In *zenvisage++*, recommendations comprise of representative trends and outliers, which are useful for understanding common and outlying behaviors when a *small number of common patterns* is exhibited in the dataset.

Result querying enables users to query for patterns similar to a selected data pattern from the ranked list of results or recommendations. Typically, analysts select visualizations with *semantic or visual properties* of interest and make use of result querying to understand characteristic properties of similar instances.

Usage Scenario: G2 does not have a preconceived knowledge of what to search for in the dataset. She learns about the characteristic patterns that exist in the dataset through the representative trends, a form of bottom-up inquiry, as a means to jump-start further queries via result querying, as well as understand groups of data instances with shared characteristics.

6.3 Context creation

While top-down and bottom-up processes operate on a collection of visualizations with fixed X and Y attributes, context creation operates in the regime where the analyst may be investigating the relationships between multiple different attributes or values of interest. Context creation enables analysts to navigate across different visualization collections to learn about patterns in different regions of the data. The design challenge of context creation is to help users visualize and compare how data changes between these different contexts by constructing visualization collections with different visual encodings (*view specification*) or different data subsets (*slice-and-dice*).

View specification settings alter the encoding for all of the visualizations on the VQS currently being examined. This ability to work with different collections of visualizations is useful when the dataset is *multidimensional* and the axes of interest are *unknown*. Modifying the view specification offers analysts different perspectives on the data to locate visualization collections of interest.

Slice-and-Dice empowers users to navigate and compare collections of visualizations constructed from different subsets of the data. Data navigation capabilities are essential when the dataset has *large numbers of “support attributes”* that may be related to the visualization attributes (e.g., geographical location may influence the time series pattern for housing prices). Analysts can either make use of pre-existing knowledge regarding these support attributes to navigate to a data region that is more likely to contain the desired pattern (e.g., filtering to suburbs to find cheaper housing) or discover unknown patterns and relationships between different data subsets (e.g., housing prices are lower in winter than compared to summer).

Usage Scenario: M1 recognizes salient trends in his dataset such as inverse or linear correlations, but does not have fixed attributes that he wants to visualize or a pattern in mind to query with. Given a list of physical properties of potential interest, he performs context creation by switching between different visualized attributes to understand the dataset from alternative perspectives. He can also dynamically create different classes of data (e.g., solvents with low solubility or have high capacity) to examine their aggregate patterns.

The three aforementioned sensemaking processes are akin to the well-studied sensemaking paradigms of search (top-down), browse (bottom-up), and faceted navigation (context creation) on the Web [16, 36]. Due to each of their advantages and limitations given different information seeking tasks, search interfaces have been designed to support all three complementary acts and transition smoothly between them to combine the strength of all three sensemaking processes. Similarly for VQSSs, our design objective is to enable all three sensemaking processes in *zenvisage++*. Our evaluation study reveals that this integrative approach not only accelerates the process of visualization discovery, but also encourages hypotheses generation and experimentation.

7 EVALUATION STUDY FINDINGS

Based on audio, video screen capture, and click-stream logs recorded during our Phase III evaluation study, we performed thematic analysis via open coding to label every event with a descriptive code. Event codes included specific feature usage, insights, provoked actions, confusion, need for capabilities unaddressed by the system, and use of external tools². To characterize the usefulness of each feature, we further labeled whether each feature was useful to a particular participant’s analysis. A feature was deemed *useful* if the feature was either used in a sensible and meaningful way to accomplish a task or address a question during the study, or has envisioned usage outside of the constrained time limit during the study (e.g., if data was available or downstream analysis was conducted). We derived these labels from the study transcript to circumvent self-reporting bias [57], which can often artificially inflate the usefulness of the feature under examination. In this section, we will apply our thematic analysis results to understand how each sensemaking process occurs in practice.

7.1 Uncovering the Myth of Sketch-to-Insight

To understand the usefulness of different visual querying modalities, we analyzed their frequency of use in our evaluation study. To our surprise, despite the prevalence of sketch-to-query systems in the literature, only two out of our nine participants found it useful to directly sketch a desired pattern onto the canvas. The reason why most participants did not find direct sketching useful was that they often do not start their analysis with a specific pattern in mind. Instead, their intuition about what to query is derived from other visualizations they encountered during exploration, in which case it makes more sense to query using

those visualizations as examples directly (e.g., by dragging and dropping that visualization onto the canvas to submit the query). Even if a user has a pattern in mind, translating that pattern into a sketch is often hard to do. For example, A2 wanted to search for a highly-varying signal enveloped by a sinusoidal pattern indicating planetary rotation , which was hard to draw by hand.

Given these initial findings, we further investigated the processes that participants engaged in to construct pattern queries. Pattern queries can be generated by either top-down (sketching **based on user’s in-the-head pattern**) or bottom-up (drag-and-drop **based on what user observes from data**) processes, driven by various different querying intentions. While our study is not intended as a quantitative study with different querying modalities as conditions, we wanted to get an estimate of the relative frequency of use of different querying mechanism across all users. We examined the sequence of interactions that led to each pattern query and labeled each pattern query based on one of the five ways it can be generated³. We find that **bottom-up processes are 40% more commonly used than top-down processes for generating a pattern query**. We will describe the different ways in which the pattern queries are generated in this subsection and the next. Within top-down processes, a pattern query could arise from users directly sketching a new pattern or by modifying an existing sketch . For example, M2 first sketched a pattern to find solvent classes with anticorrelated properties (pattern as a straight line with negative slope) without much success in returning a desired match. So he instead dragged and dropped one of the peripheral visualizations similar to his desired visualization and then smoothed out the noise in the visualization via sketching yielding a straight line, as shown in Figure 5 (left). M2 repeated this workflow twice in separate occurrences during the study and was able to derive insights from the results. Likewise, A3 was searching for pulsating stars characterized by dramatic changes in the amplitudes of the light curves. She knows that stellar hotspots also exhibit dramatic amplitude fluctuations, but unlike pulsating stars, the variations happen at regular intervals. Figure 5 (right) illustrates how A3 first picked out a regular pattern (suspected starspot), then modified it slightly so that the pattern looks more “irregular” (to find pulsating stars).

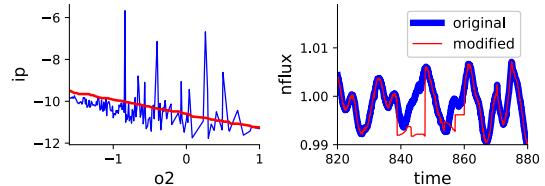


Fig. 7: Example of sketch-to-modify, based on canvas traces from M2 (left) and A3 (right). The original drag-and-dropped query is shown in blue and sketch-modified queries in red.

The infrequent use of top-down pattern specification was also reflected in the fact that none of the participants queried using an equation. In both astronomy and genetics, the visualization patterns resulted from complex physical processes that could not be written down as equations analytically. Even in the case of material science when analytical relationships do exist, it is challenging to formulate patterns as functional forms in a prescriptive manner.

We found that some users employed match specification to remedy undesired results from their top-down pattern queries. While we did not rigorously study the effects of different analytical parameter settings, we observed that more users refined their matches by adjusting the range and degree of approximation, rather than opting for a different similarity metric. This points to future work in developing more flexible and intuitive vocabularies for modifying the match along the research directions pursued in [10, 26] over incorporating additional complex, off-the-shelf matching objectives in VQSSs.

Our findings suggest that while sketching is a useful construct for people to express their queries, *the existing ad-hoc, sketch-only model for VQSSs is insufficient on its own* without data examples that can help

³Top-down: sketch-to-query, sketch-to-modify; Bottom-up: Result querying via object of interest, via ranked result, or via recommendations. See Appendix Figure 12 for more details.

²See Appendix D for details on our coding protocol.

	Process & Component	Top-Down	Context Creation	Bottom-Up	Open-Source	Recommendation	Result Querying	Slice-and-Dice	View Specification	Pattern Specification	Match Specification	Existing VQSS
TimeSearcher			✓	✓								✓
QuerySketch		✓	✓									
QueryLines		✓	✓									
SoftSelect		✓	✓									
Google Correlate		✓	✓									
TimeSketch		✓	✓									
SketchQuery		✓	✓			✓						✓
Zenvisage		✓	✓			✓	✓					✓
Qetch		✓	✓									✓
Zenvisage ++		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3: Table summarizing whether key functional components (columns) are covered by past systems (row, ordered by recency), indicated by checked cells. Column header colors blue, yellow, green represent three sensemaking process (top-down querying, search with context, and bottom-up querying) described in Section 5. The heavily-used, practical features in our study for context-creation and bottom-up inquiry is largely missing from prior VQSSs.

analysts jumpstart their exploration. In fact, **we found that sketch-to-query only accounted for about a fifth of the total number of visual queries performed during the study**. This finding has profound implications on the design of future VQSSs, since **our comparison of VQS features across existing work (as shown in Table 3)** suggests that past work has primarily focused on optimizing top-down process components, without considering how useful these features are in real-world analytic tasks. We suspect that these limitations may be why existing VQSSs are not commonly adopted in practice. Note that we are not advocating for removing the natural and intuitive sketch capabilities from future VQSSs completely, but instead focusing future research and design efforts to examine other (often underexplored) VQS sensemaking processes. Such processes could be applied in conjunction with sketching to help analysts more flexibly express their analytical goals, described next.

7.2 Insights via Context Creation and Bottom-up Approaches

As alluded to earlier, *bottom-up data-driven inquiries and context creation are far more commonly used than top-down pattern search when users have no desired patterns in mind*, which is typically the case for exploratory data analysis. In particular, top-down approaches were only useful for 29% of the use cases, whereas they were useful for 70% of the use cases for bottom-up approaches and 67% for context creation⁴. We now highlight some exemplary workflows demonstrating the efficacy of the latter two sensemaking processes.

Bottom-up pattern queries can come from either the ranked list of results, recommendations, or by selecting a particular object of interest as a drag-and-drop query. The most common use of bottom-up querying is via recommended visualizations. For example, G2 and G3 identified that the three representative patterns recommended in *zenvisage++* corresponded to the same three groups of genes discussed in a recent publication [14]: induced genes (profiles with expression levels going up ↗), repressed genes (starting high then decreasing ↘), and transients (rising first then dropping at another time point ↙). The clusters provoked G2 to generate a hypothesis regarding the properties of transients: *“Is that because all the transient groups get clustered together, or 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 clusters no longer exhibited the clean, intuitive patterns he had seen

earlier. G3 expressed a similar sentiment and proceeded by inspecting the visualizations in the cluster via drag-and-drop. He found a group of genes that all transitioned at the same timestep, while others transitioned at different timesteps.

By browsing through the ranked list of results , participants were also 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 the filter CLASS_STAR=1, which led him to discover that all stars have been mislabeled with CLASS_STAR=0 as 1 during data cleaning.

Context creation in VQSSs enables users to change the “lens” by which they look through the data when performing visual querying, thereby creating more opportunities to explore the data from different perspectives. Echoing the sentiment from past studies in visual analytics regarding the importance of designing features that enable users to select relevant subsets of data [1, 17, 48], we found that all participants found at least one of the features in context creation to be useful.

Both A1 and A2 expressed that context creation through interactive filtering was a powerful way to dynamically test conditions and tune values that they would not have otherwise experimented with, effectively lowering the barrier between the iterative hypothesize-then-compare cycle during sensemaking. 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?* (G2) and *Can I find more supernovae candidates if I query only on objects that are bright and classified as a star?* (A1). Three participants had also used filtering as a way to query with known individual objects of interest. For example, G2 set the filter as gene=9687 and explained that since *“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.”*

While filtering enabled users to narrow down to a selected data subset, dynamic classes (buckets of data points that satisfies one or more range constraints) enabled users to compare relationships between multiple attributes and subgroups of data. For example, M2 divided solvents in the database into eight different categories based on voltage properties, state of matter, and viscosity levels, by dynamically setting the cutoff values on the quantitative variables to create these classes. By exploring these custom classes, M2 discovered 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. He cited that dynamic class creation was central to learning about this previously-unknown attribute properties:

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. [...] look how quickly we can do it!

7.3 Combining Sensemaking Processes in VQS Workflows

Given our observations so far as to how participants make use of each sensemaking process in practice, we construct a Markov model to further investigate the interplay between these sensemaking processes in the context of an analysis workflow. **Markov models have been used in the past by Reda et al. [43] in a similar manner to analyze interaction sequences from open-ended, exploratory analysis evaluation studies. The goal of such analysis is to quantitatively capture how users “transitions between mental, interaction, and computational states” to afford researchers to qualitatively characterize the processes and behavioral patterns “essential to insight acquisition” [43].**

To compute the state transition probabilities in the Markov model, we make use of event sequences from the evaluation study, where each event consists of labels describing when specific features were used. Using the taxonomy in Table 2, we map each usage of a feature in *zenvisage++* to one of the three sensemaking processes. Each participant’s event sequence is divided into sessions, each indicating a separate line of inquiry during the analysis. Based on these event sequences—one for each session, we compute the aggregate state transition probabilities (edge weight labels in Figure 6) to characterize how participants from

⁴See Appendix D for details on how this measure was computed.

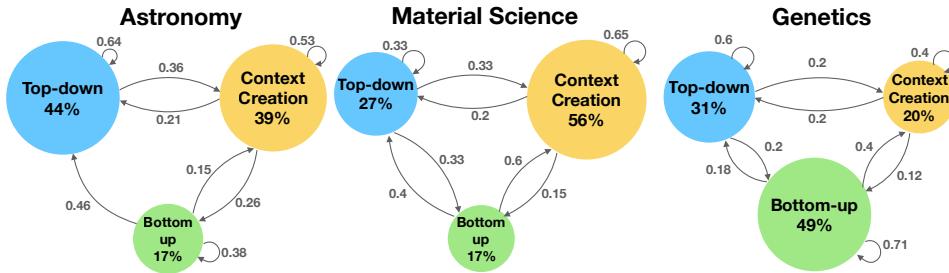


Fig. 8: Markov models computed based on evaluation study event sequences, with edges denoting the probability that participant in the particular domain will go from one sensemaking process to the next. Nodes are scaled according to their eigenvector centrality, representing the percentage of time participants would spend in a particular sensemaking process in steady state. Markov models are computed based on a total of 206 event actions taken by participants during the evaluation study (80 for astronomy, 65 for genetics, and 61 for material science).

each domain move between different sensemaking processes⁵.

The transition probability represents the probability that an action from one class would be followed by one from the other. For example, in material science, 60% of events that started with bottom-up exploration lead to context creation and to top-down pattern search the rest of the time. Self-directed edges indicate the probability that the participant would continue with the same type of sensemaking process. For example, when an astronomer performs top-down pattern search, 64% of the transitions were followed by another top-down process and by context creation the rest of the time, but never followed by a bottom-up process. This high self-directed transition probability reflects how astronomers often need to iteratively refine their top-down query through pattern or match specification when looking for a specific pattern.

To study how important each sensemaking process is for participant's overall analysis, we compute the eigenvector centrality of each graph, displayed as node labels in Figure 6. These values represent the percentage of time the participants spend in each of the sensemaking processes when the transition model has evolved to a steady state [39]. Given that nodes in Figure 6 are scaled by this value, in all domains, we observe that there is always a prominent node connected to two less prominent ones—but it is also clear that all three nodes are essential to all domains. Our observation demonstrates how *participants often construct a central workflow around a main sensemaking process based on their analytical goals and interleave variations with the two other support processes as they iterate on the analytic task*. This finding is further illustrated in Table 4, where the usage scenarios exemplify how each sensemaking process supports essential subtasks and enables participants' scientific goals. For example, the material scientists focus on context creation 56% of the time, mainly through dynamic class creation, followed by bottom-up inquiries (such as drag-and-drop) and top-down pattern searches (such as sketch modification). The central process adopted by each domain is tightly coupled with the problem characteristics associated with each domain. For example, without an initial query in mind, geneticists relied heavily on bottom-up querying through recommendations to jumpstart their queries.

The Markov transition model exemplifies how participants adopted a diverse set of workflows based on their unique set of research questions. The bi-directional and cyclical nature of the transition graphs in Figure 6 highlight how the three sensemaking processes do not simply follow a linear progression towards finding a single pattern or attribute of interest. Instead, the high connectivity of the transition model illustrates how these three equally-important processes form a sensemaking loop, representing iterative acts of dynamic foraging and hypothesis generation. This finding reinforces the importance of each sensemaking process and indicates that future VQSSs need to be *integrative* in supporting all three sensemaking process to enable a diverse set of potential workflows for addressing a wide range of analytical inquiries.

7.4 Limitations

Although evidence from our evaluation study points to the infrequent use of direct sketch, we have not performed controlled studies with

⁵We chose to break down these results by the domain, rather than on an individual basis, since we found that the analytical patterns are very similar within the domains (possibly due to how similar the analytical inquiries and datasets were within the domains).

a sketch-only system as a baseline to validate this hypothesis. While we employed quantitative comparisons in various analysis throughout this section, our goal is to gain a formative understanding of VQS usage behavior across our small sample. Future studies with larger sample sizes and more representative samples are required to generalize these findings. The goal of our study is to uncover qualitative insights that might reveal why VQSSs are not widely used in practice; further validation of specific findings is out of the scope of this paper. While concerns regarding study results being focused on *zenvisage++* must be acknowledged, we note that *zenvisage++* is one of the most comprehensive VQSSs to-date, covering many of the features from past systems and more (as evident from Table 1). We believe that our integrative VQS, *zenvisage++*, can serve as a baseline for future research in VQS to evaluate against and build upon. Given that this paper covered three design studies along with one evaluation study, we were unable to cover each domain to the level of detail typically found in a dedicated design study paper. Instead, our focus was to highlight the differences and similarities among these domains relevant to the capabilities required in VQS and we defer domain-specific participatory design details and artifacts to Appendix A. Future longitudinal studies may also help alleviate the novelty effects that participants may have experienced during the evaluation study. While we have generalized our findings beyond existing work by employing three different and diverse domains, our case studies have so far been focused on scientific data analysis with domain-experts, as a first step towards greater adoption of VQSSs. Other potential domains that could benefit from VQSSs include: financial data for business intelligence, electronic medical records for healthcare, and personal data for "Quantified Self". These different domains may each pose different sets of challenges (such as designing for novices as end-users) unaddressed by the findings in this paper, pointing to a promising direction for future work.

8 CONCLUSION

While VQSSs hold tremendous promise in accelerating data exploration, they are rarely used in practice. In this paper, we worked closely with analysts from three diverse domains to characterize how VQSSs can address their analytic challenges, collaboratively design VQS capabilities, and evaluate how VQSSs are used in practice. Participants were able to use our final system, *zenvisage++*, for discovering desired patterns, trends, and valuable insights to address unanswered research questions. Based on these experiences, we developed a sensemaking model for how analysts make use of VQSSs. Contrary to past work, we found that sketch-to-query is not as effective in practice as past work may suggest. Beyond sketching, we find that each sensemaking process fulfills a central role in participants' analysis workflows to address their high-level research objectives. We advocate that future VQSSs should invest in understanding and supporting all three sensemaking processes to effectively "close the loop" in how analysts interact and perform sensemaking with VQSSs. While more work certainly remains to be done, by contributing to a better understanding of how VQSSs are used in practice across domains, our paper can serve as a roadmap towards the broader adoption of VQSSs for novel future use cases.

Sensemaking Process																
	Top-Down	Context Creation	Bottom-Up													
Domain	<p>Astronomy</p> <p>Goal: Discover potential supernovae candidates that exhibits peak-then-decay pattern</p>	<p>Support: Examine data regions that are more likely to have supernovae candidates</p> <p>Filter Constraint: derived_class_star_1</p>	<p>Support: Identify and eliminate sources of data anomalies to improve match accuracy for finding candidates</p> <p>Outliers</p>													
	<p>Material Science</p> <p>Support: Find data classes that follows desired functional pattern to understand which solvent types exhibit certain tradeoffs and relationships</p> <p>Input Equation: $y=x^2-3$</p>	<p>Goal: Compare characteristics from different data classes to find a solvent that satisfies desirable properties</p> <p>Created classes</p> <table border="1"> <thead> <tr> <th>Ranges [l, b]</th> <th>Count</th> </tr> </thead> <tbody> <tr><td>-10.0 < l <= 37.23</td><td>7938</td></tr> <tr><td>280.0 < bp <= 1232.69</td><td>19</td></tr> <tr><td>-17.9 < l < -10.0</td><td>19</td></tr> <tr><td>280.0 < bp <= 1232.69</td><td>40</td></tr> <tr><td>-10.0 < l <= 37.23</td><td>40</td></tr> <tr><td>102.24 < bp < 280.0</td><td>1</td></tr> </tbody> </table>	Ranges [l, b]	Count	-10.0 < l <= 37.23	7938	280.0 < bp <= 1232.69	19	-17.9 < l < -10.0	19	280.0 < bp <= 1232.69	40	-10.0 < l <= 37.23	40	102.24 < bp < 280.0	1
Ranges [l, b]	Count															
-10.0 < l <= 37.23	7938															
280.0 < bp <= 1232.69	19															
-17.9 < l < -10.0	19															
280.0 < bp <= 1232.69	40															
-10.0 < l <= 37.23	40															
102.24 < bp < 280.0	1															
<p>Genetics</p> <p>Support: Search and browse for genes belonging to the same cluster</p>	<p>Support: Compare known properties of genes belonging to different clusters</p> <p>Filter Constraint: diffexp=1</p>	<p>Goal: Understand characteristic pattern profiles in the dataset</p> <p>Representative patterns</p>														

Table 4: Table of example usage scenarios from each domain for each sensemaking process. Each VQS sensemaking process maps to scientific tasks and goals from each use case, from pattern search to comparing visualization collections to improving overall data understanding. We find that our participants typically have one focused goal expressible through a single sensemaking process, but since their desired insights may not always be achievable with a single class of operation, they make use of the two other sensemaking processes to support (Support) them in accomplishing their main goal (Goal).

REFERENCES

- [1] 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
- [2] 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
- [3] A. Batch and N. Elmquist. The Interactive Visualization Gap in Initial Exploratory Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):278–287, 2018. doi: 10.1109/TVCG.2017.2743990
- [4] L. Battle, R. Chang, and M. Stonebraker. Dynamic Prefetching of Data Tiles for Interactive Visualization. *Proceedings of the 2016 International Conference on Management of Data - SIGMOD ’16*, pp. 1363–1375, 2016. doi: 10.1145/2882903.2882919
- [5] S. Bodker, K. Gronbaek, and M. Kyng. Cooperative design: Techniques and experiences from the scandinavian scene. chap. 8. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1993.
- [6] C. Bossen, C. Dindler, and O. S. Iversen. Evaluation in participatory design: A literature survey. In *Proceedings of the 14th Participatory Design Conference: Full Papers - Volume 1, PDC ’16*, pp. 151–160. ACM, New York, NY, USA, 2016. doi: 10.1145/2940299.2940303
- [7] 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
- [8] 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
- [9] L. Ciolfi, G. Avram, L. Maye, N. Dulake, M. T. Marshall, D. van Dijk, and F. McDermott. 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
- [10] 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.
- [11] Drlica Wagner et al. Dark energy survey year 1 results: The photometric data set for cosmology. *The Astrophysical Journal Supplement Series*, 235(2):33, apr 2018. doi: 10.3847/1538-4365/aab4f5
- [12] P. Eichmann and E. Zgraggen. Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings. *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI ’15*, pp. 28–37, 2015. doi: 10.1145/2678025.2701379
- [13] S. Few. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2012.
- [14] 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
- [15] 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
- [16] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st ed., 2009.
- [17] 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
- [18] H. Hochheiser and B. Shneiderman. Interactive exploration of time series data. In *Discovery Science*, pp. 441–446. Springer, Berlin, Heidelberg, 2001.
- [19] 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.
- [20] K. Holtzblatt and S. Jones. Contextual inquiry: A participatory technique for system design. chap. 9. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1993.
- [21] C. Holz and S. Feiner. Relaxed selection techniques for querying time-series graphs. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology, UIST ’09*, pp. 213–222. ACM, New York, NY, USA, 2009. doi: 10.1145/1622176.1622217
- [22] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded Evaluation of Information Visualization. *Proceedings of the 2008 conference on BEyond time and errors novel evalution methods for Information Visualization - BELIV ’08*, p. 1, 2008. doi: 10.1145/1377966.1377974

doi: 10.1109/VAST.2016.7883519

- [23] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in Visual Data Analysis. *Tenth International Conference on Information Visualization, 2006*, (Iv 2006):9–16, 2006. doi: 10.1109/IV.2006.31
- [24] A. Khetan, D. Krishnamurthy, and V. Viswanathan. Towards synergistic electrode-electrolyte design principles for nonaqueous li-o₂ batteries. *Topics in Current Chemistry*, 376, 04 2018.
- [25] 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
- [26] M. Mannino and A. Abouzied. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. pp. 1–12, 2018. doi: 10.1145/3173574.3173962
- [27] P. McLachlan, T. Munzner, and F. Park. LiveRAC : Interactive Visual Exploration of System Management Time-Series Data. *CHI 08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1483–1492, 2008.
- [28] B. McNally, M. L. Mauriello, M. L. Guha, and A. Druin. Gains from participatory design team membership as perceived by child alumni and their parents. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17, pp. 5730–5741. ACM, New York, NY, USA, 2017. doi: 10.1145/3025453.3025622
- [29] M. Mohebbi, D. Vanderkam, J. Kodysh, R. Schonberger, H. Choi, and S. Kumar. Google correlate whitepaper. 2011.
- [30] M. J. Muller and S. Kogan. Grounded Theory Method in HCI and CSCW. *Human Computer Interaction Handbook*, pp. 1003–1024, 2012.
- [31] M. J. Muller and S. Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, June 1993. doi: 10.1145/153571.255960
- [32] 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
- [33] 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
- [34] D. A. Norman and S. W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.
- [35] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. doi: 10.1109/MCG.2006.70
- [36] C. Olston and E. H. Chi. ScentTrails. *ACM Transactions on Computer-Human Interaction*, 10(3):177–197, 2003. doi: 10.1145/937549.937550
- [37] N. O. Ommen, M. Blut, C. Backhaus, and D. M. Woisetschläger. Toward a better understanding of stakeholder participation in the service innovation process: More than one path to success. *Journal of Business Research*, 69(7):2409 – 2416, 2016. doi: 10.1016/j.jbusres.2016.01.010
- [38] 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
- [39] B. Pierie. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 2011.
- [40] 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.
- [41] 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
- [42] 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
- [43] K. Reda, A. E. Johnson, M. E. Papka, and J. Leigh. Modeling and evaluating user behavior in exploratory visual analysis. *Information Visualization*, 15(4):325–339, 2016. doi: 10.1177/1473871616638546
- [44] K. Ryall, N. Lesh, T. Lanning, D. Leigh, H. Miyashita, and S. Makino. 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
- [45] D. Schuler and A. Namioka, eds. *Participatory Design: Principles and Practices*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1993.
- [46] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012. doi: 10.1109/TVCG.2012.213
- [47] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human Computer Interaction*. John Wiley & Sons, Inc., USA, 2007.
- [48] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994. doi: 10.1109/52.329404
- [49] 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
- [50] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran. 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
- [51] T. Siddiqui, J. Lee, A. Kim, E. Xue, X. Yu, S. Zou, L. Guo, C. Liu, C. Wang, K. Karahalios, and A. Parameswaran. Fast-forwarding to desired visualizations with zenvisage. In *The biennial Conference on Innovative Data Systems Research (CIDR)*, 2017.
- [52] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672 – 681, 2019.
- [53] Susanne Bodker and K. Grønbæk. Cooperative Prototyping. *International Journal of man-machine studies*, pp. 1–23, 1991.
- [54] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1970.
- [55] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics. *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2017.
- [56] 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
- [57] P. Williams, J. Jenkins, J. Valacich, and M. Byrd. Measuring Actual Behaviors in HCI Research—A call to Action and an Example. *AIS Transactions on Human-Computer Interaction*, 9(4):339–352, 2017. doi: 10.17705/1thci.00101
- [58] 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 evalution methods for Information Visualization - BELIV ’08*, p. 1, 2008. doi: 10.1145/1377966.1377971

In Appendix A, we first describe additional details about the participatory design process, as well as domain-specific artifacts collected from contextual inquiry. Next, in Appendix B, we articulate the space of problems amenable to VQSSs and describe how the sensemaking processes (introduced in Section 6) fit into different parts of the problem space. Finally, in Appendix C, we provide supplementary information regarding our analysis methods and results for the evaluation study.

A ARTIFACTS FROM PARTICIPATORY DESIGN

During the contextual inquiry, participants demonstrated the use of domain-specific tools for conducting analysis in their existing workflow, including:

- Image Cutout Service (Astronomy)
- Short Time-series Expression Miner (Genetics)
- Solubility Database (Material Science)

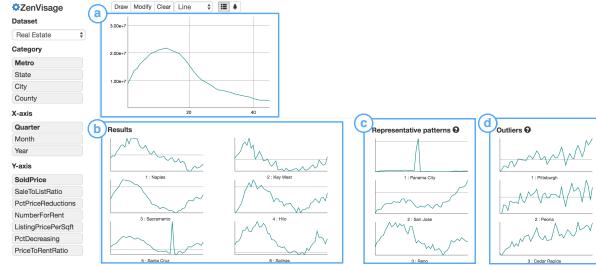


Fig. 9: The existing Zenvisage prototype allowed users to sketch a pattern in (a), which would then return (b) results that had the closest Euclidean distance from the sketched pattern. The system also displays (c) representative patterns obtained through K-Means clustering and (d) outlier patterns to help the users gain an overview of the dataset.

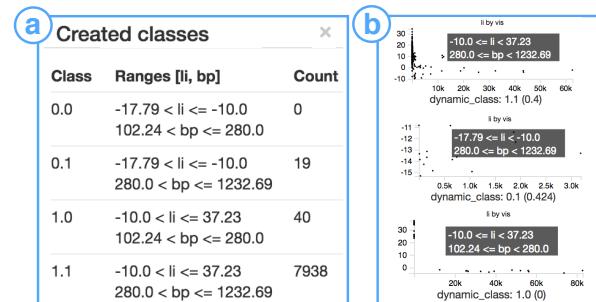


Fig. 10: Example of dynamic classes. (a) Four different classes with different Lithium solvation energies (li) and boiling point (bp) attributes based on user-defined data ranges. (b) Users can hover over the visualizations for each dynamic class to see the corresponding attribute ranges for each class. The visualizations of dynamic classes are aggregate across all the visualizations that lie in that class based on the user-selected aggregation method.

B CHARACTERIZING THE PROBLEM SPACE FOR VQSSs

We now characterize the space of problems addressable by VQSSs and describe how each sensemaking process fits into different problem areas that VQSSs are aimed to solve. Visual querying often consists of searching for a desired pattern instance (Z) across a visualization collection specified by some given attributes (X, Y). Correspondingly, we introduce two axes depicting the amount of information known about the visualized attribute and pattern instance as shown in Figure 10.

Along the **pattern instance** axis, the visualization that contains the desired pattern may already be known to the analyst, exist as a pattern *in-the-head* of the analyst, or be completely *unknown* to the analyst. In the known pattern instance region (Figure 10 grey cell), systems such as Tableau, where analysts manually create and examine each visualization one at a time, is more well-suited than VQSSs, since analysts can directly work with the selected instance without having to search for which visualization exhibits the desired pattern. We define *top-down pattern search* as the process where analysts query a fixed collection of

visualizations based on their in-the-head pattern (Figure 10 blue). On the other hand, *bottom-up data-driven inquiries* (Figure 10 green) are driven by recommendations or queries that originate from the data (or equivalently, the visualization), since the pattern of interest is unknown and external to the user.

The second axis, **visualized attributes**, depicts how much the analyst knows about which X and Y axes they are interested in visualizing. In both the astronomy and genetics use cases, as well as past work in this space, the attribute to be visualized is known, as data was in the form of a time series. In the case of our material science participants, they wanted to explore relationships between different X and Y variables. In this realm of *unknown* attributes, context creation (Figure 10 yellow) is essential for allowing users to pivot across different visualization collections.

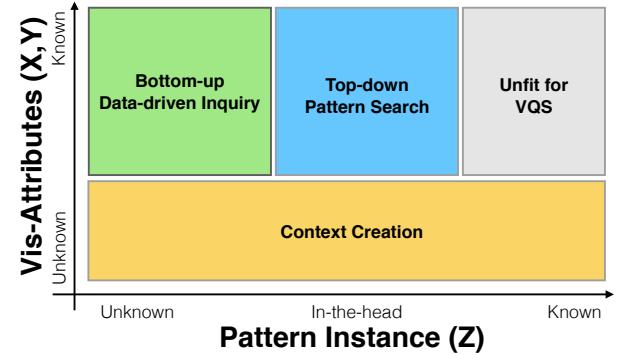


Fig. 11: The problem space for VQSSs is characterized by how much the analyst knows about the visualized attributes and the pattern instance. Colored areas highlight the three sensemaking processes in VQSSs for addressing these characteristic problems. While prior work has focused solely on use cases in the blue region, we envision opportunities for VQSSs beyond this to a larger space of use cases covered by the yellow and green regions.

C EVALUATION STUDY PROTOCOL

Here, we detail the procedures that were conducted during the evaluation study. At the beginning of the study, participants were asked a set of pre-study survey questions to collect basic information about participant's dataset, scientific questions, and existing workflows. While this information was similar to the ones collected through participatory design and contextual inquiry (Section 4), the pre-study survey ensured that we have background information even for the “blank-slate” participants (who were not part of the earlier design study).

- What is your current role as a scientist? What are some examples of recent questions you've researched?
- Describe the workflow that you currently use to analyze and make sense of this type of data.
- Can you describe an interesting finding you found with your current workflow and the process you took to obtain this insight?

After the tutorial and overview of the system, participant's selected dataset was loaded in. Participants were asked about their familiarity with the dataset and their analytical goals for the session.

- How familiar are you with this dataset? If you have worked with this dataset before, is there any insight that you already know from this dataset?
- What is your goal for this dataset? How long have you been working with this dataset? What are you hoping to accomplish with this dataset?

During the main experiment, participants engaged in talk-aloud exercises as they explored their data. These two semi-structured interview questions were often posed when participants begin a new line of analytical inquiry.

- What is your current goal in this phase of the exploration? What type of insights are you hoping to obtain?
- What actions are you planning to perform? How are you operationalize to achieve those goals?

In addition, we occasionally remind participants that they ask for help on something they want to accomplish on *zenvisage++*, but were not

sure about the sequence of interactions. They were also encouraged to use other tools in their existing workflow alongside *zenvisage++* to perform their analysis.

At the end of the study, we interviewed participants with a set of open-ended questions regarding their experience with *zenvisage++*, as listed below.

- How was *zenvisage++* different from your existing workflow?
- Can you describe how you would use *zenvisage++* in your current workflow?
- On a scale of 1-10, how interested would you be in adopting this tool for your day-to-day workflow?
- What were some insights that you have gained from this session?
- Given the insights that you have obtained from *zenvisage++*, what additional analysis will you run downstream before you publish these results?
- What are the pros/cons for using *zenvisage++*?
- Were there any queries that you were unable to address with *zenvisage++* during today’s session?
- What are additional features in *zenvisage++* that would help with your scientific workflow or serve your scientific need?

D EVALUATION STUDY ANALYSIS DETAILS

We analyzed the transcriptions of the evaluation study recordings through open-coding and categorized every event in the user study using the following 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 that provoked a scientific hypothesis to be generated.
- Provoke (Data) [PD]: Interactions or observations 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 Tool [E]: The use of external tools outside of *zenvisage++* to complement the analysis process.
- Feature Usage [F]: One of the features in *zenvisage++* was used.
- Session Break [BR]: Transition to a new line of inquiry.

Domain	IS	ID	PS	PD	C	W	E	BR	F
astro	4	12	13	57	2	18	20	22	67
genetics	8	12	7	35	4	13	1	21	52
mat sci	14	8	7	44	8	11	3	12	48

Table 5: Count summary of thematic event code across all participants of the same subject area.

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 [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.

The feature usage labels for each user is summarized in Figure 11. A feature is regarded as *useful* if it has a P or E code label. Using the matrix from Figure 11, we compute the percentage of useful features for each sensemaking process as:

$$\frac{\# \text{ of useful features in process}}{\text{total } \# \text{ of features in process} \times \text{total } \# \text{ of users}}$$

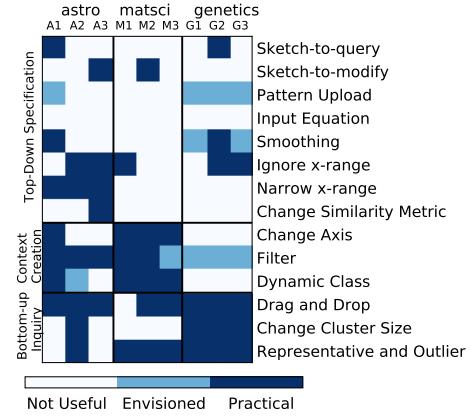


Fig. 12: Heatmap of features categorized as practical usage (P), envisioned usage (E), and not useful (N). Columns are arranged in the order of subject areas and the features are arranged in the order of the three foraging acts. Participants preferred to query using bottom-up methods such as drag-and-drop over top-down approaches such as sketching or input equations. Participants found that context creation via filter constraints and dynamic class creation were powerful ways to compare between subgroups or filtered subsets.

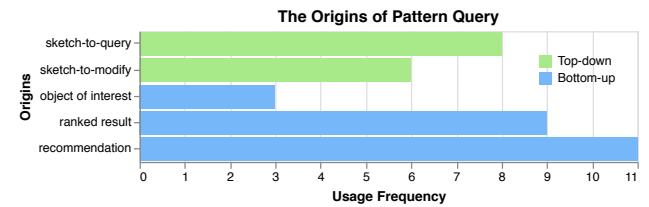


Fig. 13: The number of times a pattern query originates from one of the workflows. Pattern queries are far more commonly generated via bottom-up than top-down processes.