

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

Category: Research

Paper Type: application/design study

**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 their potential in accelerating data exploration, more than a decade of past work on VQSs has not been translated to adoption in practice. Through a year-long collaboration with experts from three diverse domains, we examine the role of VQSs in real data exploration workflows, enhance an existing VQS to support these workflows via a participatory design process, and evaluate how VQS components are used in practice. Via these observations, we formalize a taxonomy of key capabilities for VQSs, organized by three sensemaking processes. Perhaps somewhat surprisingly, we find that ad-hoc sketch-based querying is not commonly used during data exploration, since analysts are often unable to precisely articulate the patterns they are interested in. We find that there is a spectrum of VQS-centric data exploration workflows, depending on the application domain, and that many of these workflows are not effectively supported in present-day VQSs. Our insights can pave the way for next-generation VQSs to be adopted in a variety of real-world applications. **Doris:** We might want to merge the rebuttal 1st paragraph into the abstract or intro (Pasted below). Despite decades of past work on visual query systems (VQSs), they have not been adopted in practice. Our paper answers the question “why?”, via participatory design and evaluation of a VQS with three groups of domain experts over a year. Our findings: 1) Sketching is not as commonly used as prior work suggests; 2) Prior VQSs are not effective because they don't support the three important types of sensemaking processes we identified. In fact, each domain exhibits different sensemaking patterns—indicating the need for an integrative VQS, like ours, that addresses a range of analytical inquiries.

**Index Terms**—Visual analytics, exploratory analysis, visual query

## 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]. To discover patterns of interest, analysts construct line chart visualizations using toolkits like *ggplot* or *matplotlib*, or visualization construction interfaces like Excel or Tableau, by specifying *exactly* what they want to visualize. For example, when trying to find celestial objects corresponding to supernovae, which have a specific pattern of brightness over time, astronomers individually inspect the corresponding line chart for each object—numbering in the hundreds—until they find ones that match the pattern. This process of manual exploration of large numbers of line charts to identify patterns is not only error-prone, but also overwhelming for analysts.

To address this challenge, there has been a large number of papers dedicated to building *Visual Query Systems* (VQSs)—the term coined by Ryall et al. [39] to describe systems that allow users to specify and search for desired *line chart patterns via visual* interfaces [10, 12, 19, 21, 26, 28, 39, 42, 48]. These interactive interfaces often 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. For example, since the intent of a sketch can be ambiguous, follow-up work has developed mechanisms to enable users to clarify how a sketch should be interpreted [10, 12, 21, 26, 39].

While these intuitive specification interfaces were proposed as a promising solution to the problem of painful manual exploration of visualizations for time-series analysis [39, 48], to the best of our knowledge, VQSs have not lived up to these expectations and are not very commonly used in practice. Our paper seeks to bridge this gap 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. As illustrated in Figure 1, 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?

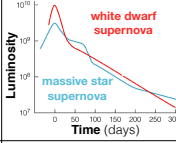
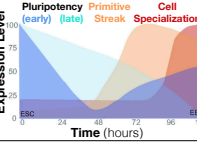
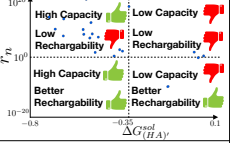
	Astronomy	Genetics	Material Science
Desired Insights			
Challenges	Discover <i>rare</i> astronomical objects with <i>specific</i> pattern properties in a <i>large</i> dataset containing <i>noisy, non-uniform</i> time series data.	Understand <i>characteristic</i> profiles amongst a <i>large</i> number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

Fig. 1: Desired insights, problem and dataset challenges for each of the three application domains in our study.

Via contextual inquiries 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* [42, 43], we engaged participants in a process of participatory design (PD) [5, 20, 30] to enable them to better compose data exploration workflows that lead to insight discovery, over the course of a year. We chose to perform participatory design across multiple domains (an uncommon practice for visualization design studies) in order to observe differences and commonalities across domains to obtain generalizable insights regarding VQSs, rather than targeting for a domain-specific solution. We organize our PD findings into a taxonomy of VQS capabilities, involving three sensemaking processes inspired by Pirolli and Card’s notional model of analyst sensemaking [36]. 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), while largely ignoring the two other processes that we found to be essential in all three domains, partially explaining why such systems 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 1.5-hour study, participants were able to gain novel scientific insights, such as identifying a star with a transient pattern that was known to harbor a Jupiter-sized planet and finding characteristic gene expression profiles confirming the results of a related publication.

By analyzing the evaluation study results, we discovered that sketching a pattern for querying is often ineffective on its own. This is due to the fact that sketching makes the problematic assumption that users know the pattern that they want to sketch and are able to sketch it precisely. However, the geneticist from our study does not have a preconceived knowledge of what to sketch and search for and relied heavily on recommended patterns to jumpstart their queries; likewise, while material scientists from our study were interested in datapoints that falls within specific value-ranges, they did not have an apriori pattern in mind to sketch. As a result, 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). However, most VQSs do not support these other mechanisms (as we argued earlier, they typically focus only on top-down sensemaking processes, without covering bottom-up and context creation). Doris: cutting this out since already mentioned 2 paragraphs ago.

To further understand how participants engage with VQSs in their analytical workflows, we used a Markov model to study how participants transition between different sensemaking processes during their analysis. Our empirical task analysis illustrates how participants often construct a diverse set of analysis workflows tailored to their domains by focussing 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 essential for enabling users to effectively use VQSs for data exploration.

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, domain-expert 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. Our contributions include:

- a characterization of the problems addressable by VQSs through design studies with three different domains,
- the construction of a taxonomy of essential VQSs capabilities leading to a sensemaking model for VQSs, grounded in participatory design findings,
- an integrative VQS, *zenvisage++*, post participatory design capable of facilitating rapid hypothesis generation and insight discovery,
- 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 WORKS

We will now describe past work in visual query systems and existing evaluation methods of visualization systems to provide background and motivation to our work. Then, we will introduce Pirolli and Card’s sensemaking model, which serves as a framework for contextualizing our study findings.

### Visual Query Systems: Definition and Brief Survey

*Visual query system* (VQS) is a term introduced by Ryall et al. and Correll and Gleicher [10, 39] to describe systems that enable analysts to directly search for time-series visualizations matching certain patterns through an intuitive 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 time series that does not pass through it, and QuerySketch [48] and Google Correlate [28], where the query is sketched as a pattern on

canvas, with the tool filtering out all of the time series that have a different shape. Subsequent work, including TimeSketch, SketchQuery, and Qetch, recognized the ambiguity in sketching by studying how humans rank similarity in patterns [10, 12, 26]. These VQSs include finer-grained specification interfaces and pattern-matching algorithms that improved the expressiveness of sketched queries, including QueryLines [39] where queries can be flexibly composed from soft constraints and preferences and SoftSelect [21] where users can vary the level of sketch similarity across a search pattern. Beyond sketching, both *zenvisage* [42, 43], SketchQuery, and TimeSearcher allow users to select an existing visualization as a query, either via drag-and-drop or double-clicking on the existing visualization. In our work, we built on an existing VQS, *zenvisage*, since it was open-source, extensible, and included features beyond pattern and match specification typically found in existing systems, as compared in Table 1. Doris: I didn’t incorporate Aditya’s comment on stating that “(the details of each component is described in Section 5)” here because it is already stated in the Table caption.

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

Table 1: Table summarizing whether key functional components (columns) are covered by past systems (row, ordered by recency), indicated by checked cells. Column header colors blue, orange, green represents 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 VQSs.

### Design and Evaluation Methodologies for Visualization Systems

Visualization systems are typically evaluated via in-lab usability studies or controlled studies against existing visualization baselines [32, 37, 50]. 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 have prompted the visualization research community to develop more participant-centered, ethnographic approaches for understanding how analysts perform visual data analysis and reasoning [25, 31, 37, 40, 41]. 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 [41].

In the context of Munzner’s nested model for visualization design and evaluation [31], 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. In other words, even though advanced VQS algorithms and interactions have been developed and shown to be effective in lab studies, no prior work have yet to address the first two steps of the research problem to evaluate: 1) whether this is even the right problem to solve, by characterizing and interviewing target users and 2) whether

the chosen data and operations acutally solve the user’s problems, by observing how the tool is used as a part of a real-world workflow. Our work fills this crucial gap in the existing literature and demonstrates how incorrect assumptions adopted by most prior work in this space regarding the first two stages of Munzner’s model may have led to failure in VQS adoption.

In this work, we performed design studies [25, 40, 41] with three different subject areas for *domain problem characterization* by adopting participatory design practices [15, 30] to engage potential stakeholders in designing a VQS that they may eventually use in their analytical workflow. Participatory design is well-established in the CHI and CSCW community and has been successfully applied to develop systems for visual analytics [2, 8], tangible museum experiences [9], and scientific collaborations [7, 38]. Past research on participatory design has found that the use of functional prototypes is a common and effective way of engaging with participants and providing a starting point for participatory design [9]. Similarly, we provide a functional prototype at the beginning of the participatory design sessions to showcase capabilities of VQSSs. Since our participants were not aware of the existence of VQSSs, let alone using them in their workflows, they would not have been able to imagine use cases for VQS without a starting point. Likewise, the use of “*simulated future work situation*” (where users are introduced to the envisioned use of the prototype) is also a common practice in cooperative prototyping when the real use of the prototype is not feasible [45]. To better understand how VQSSs can be used in-situ participant’s existing workflow, we regularly gathered feedback from participants and collaboratively envisioned potential designs based on preview demos of preliminary versions of our prototype *zenvisage++*. Finally, we validated our abstraction design with grounded evaluation [22, 37], 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. Doris: Regarding Aditya’s question on whether we could argue that PD is the only way to truly get at the heart of the problem underlying adoption. I don’t think we should say this, there are many other possible techniques that Munzner’s paper lists for dealing with domain and abstraction threats (for domain threats: ethnographic field studies, semi-structured interviews; for abstraction threats: formative testing collecting anecdotal evidence, long-term field study with deployed system). It would be hard to argue that PD is the \*right\* approach amongst these.

### Sensemaking Models for Visual Analytics

Based on our participatory design findings, we contribute to the *data/operation abstraction design* of VQSSs in Munzner’s model by developing a taxonomy for understanding how analysts make use of VQSSs to accomplish their analytical tasks. To develop a sensemaking model for VQS, we draw from Pirolli and Card’s seminal paper on information sensemaking of intelligence analysts based on cognitive task analysis [36]. The sensemaking framework was designed to capture how expert analysts iteratively search and re-represent gathered evidence into a conceptual model (*schema*). Many papers in visual analytics have also applied the sensemaking framework to motivate tool designs, such as for exploratory browsing of visualizations in large datasets [4] and of the Web [33]. In addition, the sensemaking framework has also been used for understanding and modeling user behavior in visual analytics, including how analysts gain insights from visualizations [50], how biases can be introduced during visual analysis [47], and how analysts transition between natural-language generated data facts and visualizations [44]. Based on this framework, the sensemaking process can be organized into: 1) a foraging loop that searches for information to further the organization into a schema and 2) a sense-making loop for constructing a schema that best aligns with the insights obtained from the analysis. Overall, the model distinguishes between information processing tasks that are *top-down* (from theory to data) and *bottom-up* (from data to theory), described more in Section 6. We were inspired by this model for expert intelligence analysis as it bears semblance to our work for studying how domain experts perform visual analysis using VQSSs.

## 3 METHODS

Via interviews and cognitive walkthroughs, 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 in the process of participatory design, leading to a final prototype *zenvisage++*. At the end of the design study, we conducted a realistic evaluation study to study how VQSSs are used in the real-world analytical workflows.

### 3.1 Phase I: Contextual Inquiry and Participatory Design

We recruited participants by reaching out to research groups, who have experienced challenges in data exploration, via email and word of mouth. Based on our early conversations with analysts from 12 different potential application areas, we narrowed down to three use cases in astronomy, genetics, and material science for our design study, chosen based on their suitability for VQSSs as well as diversity in use cases. Six scientists (1 female, 5 male), with an average of more than 6 years of research experience in their respective fields, participated in the design process. Via interviews and contextual inquiries, we identified the needs and challenges of these use cases based on each participant’s existing analysis workflow.

For the participatory design study, we built on top of an existing VQS, *zenvisage* [42, 43], to create a functional prototype that served as a starting point for discussion around VQSSs. During participatory design, we collaborated with each team closely with an average of two meetings per month, where we learned about their datasets, objectives, and what additional VQS functionalities could help address their research questions. Since some of the essential features that were crucial for effective exploration were lacking in *zenvisage* and still under development in the new version of our VQS, *zenvisage++*, we did not provide a deployed prototype for participants to actively use on their own during the participatory design period. Instead, as we iterated on the design of these features, relevant capabilities from intermediate versions of *zenvisage++* were demonstrated to the participants. Participants also had the opportunity to interact with the low-fidelity prototype through the help of a guided facilitator. During this process, we elicited feedback from participants on how the VQS could better support their scientific goals. A summary timeline of our collaboration with participants over a year can be found in Figure 8 in Appendix A. Through this process, we identified and incorporated several crucial capabilities into *zenvisage++*, as listed in Table 3. Given the space limitations, we will focus our discussion in Section 5 on the major capabilities relevant to the study findings, and defer the details of other features to the appendix and online documentation<sup>1</sup>.

	ID	Dataset	Design Participant	Position	Years of Experience	Dataset Familiarity
Astronomy	A1	DES	✓	Researcher	10+	3
	A2	Kepler		Postdoc	8	5
	A3	Kepler		Postdoc	8	5
Genetics	G1	Mouse	✓	Grad Student	4	4
	G2	Breast 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 2: Participant information. The Likert scale used for dataset familiarity ranges from 1 (not familiar) to 5 (extremely familiar).

### 3.2 Phase II: Evaluation Study

At the end of our participatory design study, we performed a qualitative evaluation to study how analysts interact with different VQS components in practice. In order to make the evaluation more realistic, we invited participants to use datasets that they have a vested interest in exploring to address unanswered research questions. As shown in Table 2, the evaluation study participants included the six scientists from participatory design, along with three additional “blank-slate” participants who had never encountered *zenvisage++* before. The use of all or a subset of the project *stakeholders* as evaluation participants is common in participatory design [6]. While the small sample size of participant is a limitation to our study generalizability, this is a common challenge

<sup>1</sup>github.com/[Anonymized for Submission]/wiki



when recruiting domain-experts, whose specific expertise and skills are rare in quantity and have limited time due to their workplace demands relative to the general population, as echoed in prior work [3, 27].

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 participatory design. None of the participants received monetary compensation for the study, as this is not a common practice for participatory design with stakeholders. As detailed in Table 2, the nine participants brought a total of six different datasets to the study.

At the start, participants were provided with an interactive walk-through of system details and given approximately ten minutes for a guided exploration of *zenvisage++* with 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<sup>2</sup> 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 how it may or may not be applicable to their workflow. The user study ended after they covered all the main functionalities and lasted on average for 63 minutes. After the study, we asked participants open-ended questions about their experience.

#### 4 PARTICIPANTS AND DATASETS

In this section, we describe our study participants, their scientific goals, and their preferred analysis workflows, *based on the contextual inquiry that we conducted at the start of the design study*. While we collaborated with each application domain in depth, we focus on the key findings in each domain to highlight their commonalities and differences, in order to provide a backdrop for our generalized VQSs findings described later on. *Comparing and contrasting between the diverse set of questions, datasets, and challenges across these three use cases revealed new generalizable insights and was essential in enabling us to better understand how VQSs can be extended for novel and unforeseen use cases.*

**Astronomy:** 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 (objects whose brightness changes dramatically as a function of time), such as supernovae or quasars. Their dataset consists of a large collection of *light curves: brightness observations over time, one associated with each astronomical object, plotted as 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 is to identify potential astronomical transients in order to study their properties.

To identify transients, astronomers programmatically generate visualizations of candidate objects with *matplotlib* and visually examine each 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 is 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 VQSs as he recognized how specific pattern queries could help astronomers directly search for these rare transients.

**Genetics:** Gene expression is a common measurement in genetics obtained via microarray experiments [34]. 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 typical workflow is as follows: G1 first loads the preprocessed gene expression data into custom desktop application to visualize and cluster the profiles

Prior to the study, G1 and G3 spent over a month attempting to determine the best number of clusters based on a series of static visualizations and statistics computed after clustering. While regenerating their results took no more than 15 minutes every time they made a change, the multi-step, segmented workflow meant that all changes had to be done offline. They were interested in VQSs, as interactively querying time series with clustering results had the potential to dramatically speed up their collaborative analysis process.

**Material Science:** We collaborated with material scientists at a research university who are working to identify solvents for energy efficient and safe batteries. These scientists work on a large simulation dataset containing chemical properties for more than 280,000 solvents [24]. Each row of their dataset represents a unique solvent with 25 different chemical attributes. We worked closely with a postdoctoral researcher (M1), professor (M2), and graduate student (M3) for over a year to design a sensible way of exploring their data. They wanted to use VQSs to identify 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 need to understand how changes in certain chemical attributes affect other properties under specific conditions.

M1 typically starts his data exploration process by iteratively applying filters to a list of potential battery solvents using SQL queries. Once the remaining solvent list is sufficiently small, he manually examines the properties of each solvent individually by examining 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, enable researchers to select the final set of desirable solvents that they can feasibly experiment with in lab. They were interested in VQSs as it was impossible for them to uncover hidden relationships between different attributes across large numbers of solvents manually.

Next, we describe the collaborative feature discovery process and the system prototype from participatory design.

#### 5 DESIGN PROCESS AND SYSTEM OVERVIEW

Holzblatt and Jones [20] describes contextual inquiry as a technique that forms the basis for “*developing a system model that will support [a] user’s work*” that subsequently “*fosters participatory design*”. Given the need for a VQS highlighted via contextual inquiry, we further collaborate with participants to develop features to address their problems and challenges. In this section, we first reflect on our feature discovery process to introduce the participatory design (PD) findings, then we provide a high-level system overview of the product of PD, *zenvisage++*.

##### 5.1 The Collaborative Feature Discovery Process

Throughout the PD process, we worked closely with participants to discover VQS capabilities that are essential for addressing their high-level domain challenges. We identified various subtasks based on the participant’s workflow, 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] cites 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 PD objective was to collect as many feature proposals as possible, while being inclusive across different domains. We further organized these features into Table 3 through an iterative coding process by one of the authors.

In grounded theory methods [29], researchers first create open codes to assign descriptive labels to raw data, followed by grouping open codes together by relationships or categories to form axial codes. Fi-

<sup>2</sup>query by sketching, drag-and-drop, pattern loading, input equations, representative and outliers, narrow/ignore x-range options, filtering, data smoothing, creating dynamic classes, data export

Component	Task Example	Feature	Purpose	Similar Features in Past VQSs
Pattern Specification	<b>A:</b> Find supernovae candidates with peak-then-long-tail-decay pattern. <b>M:</b> Find patterns exhibiting inversely proportional chemical relationship. <b>A:</b> Find supernovae based on previously discovered sources.	Query by Sketch (Figure 2B1) Input Equation (Figure 2A1) Pattern Upload (Figure 2D2)	• Freehand sketching of a pattern query. • Specify a exact functional form as a pattern query (e.g., $y=x^2$ ). • Upload a pattern consisting of a sequence of X,Y points as a query.	All include sketch canvas except [19]. — Upload CSV [28]
Match Specification	<b>A, M:</b> Eliminate patterns matched to spurious noise. <b>A:</b> Match only around peaked region. <b>M:</b> Match only around regions exhibiting linear or exponential relationships. <b>A:</b> Search for existence of a peak above a certain amplitude. <b>G:</b> Search for “generally-rising” patterns.	Smoothing (Figure 2D2) Range Selection (Figure 2B2, D4) Range Invariance (Figure 2D1,4)	• Adjust the level of denoising on visualizations, effectively changing the degree of shape approximation when performing pattern matching. • Restrict to query only in specific x/y ranges of interest through brushing x-range and filtering y-range selections. • Ignore vertical or horizontal differences in pattern matching through option for x-range normalization and y-invariant similarity metrics.	Smoothing [26] Angular slope queries [19] Trend querylines [39] Text Entry [26, 48] Min/max bounds [39] Range Brushing [18] Temporal invariants [10]
View Specification	<b>M:</b> Explore tradeoffs and relationships between physical attributes. <b>M:</b> Non-time-series data should be displayed as scatterplot.	Data selection (Figure 2A) Display control (Figure 2D4)	• Alter the collection of visualizations to search over. • Modify how visualizations are displayed.	— —
Slice-and-Dice	<b>A:</b> Eliminate unlikely candidates by navigating to more probable data regions. <b>M, G:</b> Compare overall patterns in different data subsets. <b>A, M:</b> Examine aggregate patterns of different data classes.	Filter (Figure 2D3) Dynamic Class (Figure 10)	• Display and query only on data that satisfies the composed filter constraints. • Create custom classes of data that satisfies one or more specified range constraints. • Display aggregate visualizations for separate data classes.	— —
Result Querying	<b>A, G, M:</b> Find other similar objects and examine their overall properties.	Drag-and-drop (Figure 2C, E)	• Query with selected visualization (either from recommendations or results).	Drag-and-drop [18] Double-Click [10]
Recommendation	<b>A:</b> Examine anomalies and debug data errors through outliers. <b>G, M:</b> Understand representative trends common in dataset (or filtered subset).	Representative and Outliers (Figure 2E)	• Display visualizations of common trends and outlier instances based on clustering.	—

Table 3: List of major features incorporated via participatory design. We organize each feature based on its functional component. 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 VQSs. Given the exhaustive nature of Table 3, 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 effect

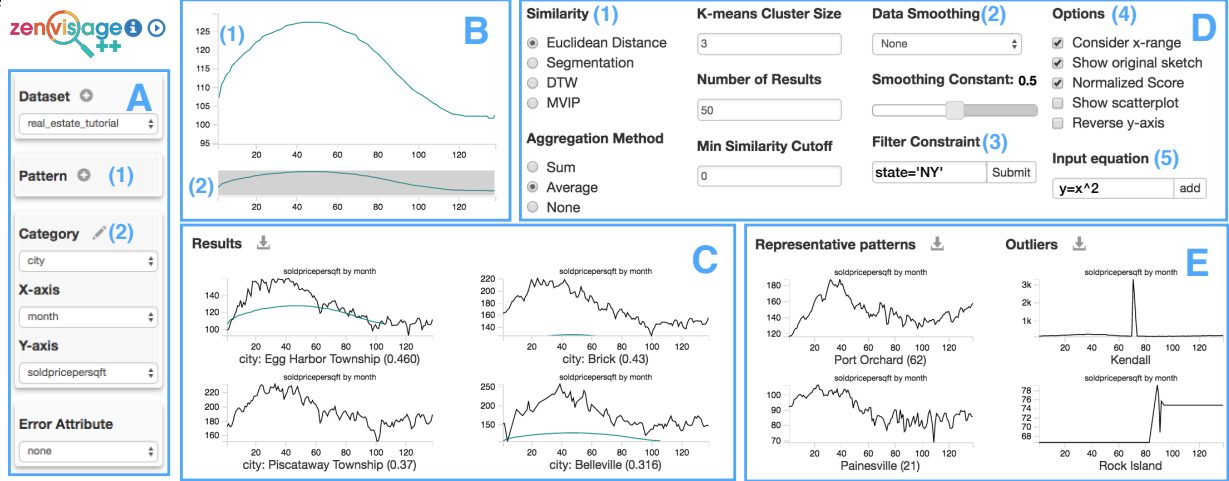


Fig. 2: The *zenvisage++* system consists of : (A) data selection panel (where users can select visualized dataset and attributes), (B) query canvas (where the queried data pattern is submitted and displayed), (C) results panel (where the visualizations most similar to the queried pattern are displayed 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).

nally, selective codes are obtained by focusing on specific sets of axial codes to come up with a set of core emerging concepts. Inspired by grounded theory methods, we first collect the list of features and example usage scenarios from PD and similar capabilities in existing VQSs as open codes. Then, we further organize this list into axial codes representing “components” (first column in Table 3): core functionalities that are essential in VQSs. Finally, as we will describe in Section 6, the selective codes capture each of the sensemaking processes (denoted by cell colors in Table 3). Instead of describing this table in detail, we present a typical example of how this table is organized. As shown in row 4 of Table 3, **one of the common challenge 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 (row 5 and 6), 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), described in Section 6.

It is important to note that while some of the proposed features in Ta-

ble 3 are pervasive in other general visual analytics (VA) systems [1,17], they have not been incorporated in present-day VQSs. 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 comes 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 [23,46], our collaborative feature discovery approach for aiding such analysis comes with its advantages and limitations. One such advantage is that introducing the newly-added features to *zenvisage++* that addressed a particular domain often results in unexpected use cases for other domains. Considering feature proposals from multiple domains can also lead 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. Through these use cases, data filtering arose as a crucial, common operation that was later incorporated into *zenvisage++* to support this class of queries.

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 the design phase, there were numerous problems and associated features proposed by participants, not all of which were incorporated. We detail the list of criteria that was used to determine whether to implement a proposed feature (including eliminating features that were nice-to-have, one-shot operations, non-essential, or required a substantially different set of research questions) in Appendix A. 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.

## 5.2 System Overview

The features in Table 3 were incrementally incorporated and improved over time, leading to our final PD product, *zenvisage++*. The *zenvisage++* interface is organized into 5 major regions all of which dynamically update upon user interactions. Typically, users begin analysis by selecting the dataset and attributes to visualize in the *data selection panel* (Figure 2A). 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 2B). *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 2C). At any point during the analysis, analysts can adjust various system-level settings through the *control panel* (Figure 2D) or browse through the list of *recommendations* provided by *zenvisage++* (Figure 2E). For comparison, the existing *zenvisage* system (Figure 9 in Appendix A) from [43] 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: [github.com/\[AnonymizedforSubmission\]](https://github.com/[AnonymizedforSubmission]).

## 6 A SENSEMAKING MODEL FOR VQSs

Now that we have described our eventual PD prototype *zenvisage++*, we revisit Table 3 in an effort to contextualize our PD findings using Pirolli and Card’s sensemaking framework [36]. We demonstrate how features in *zenvisage++* address the analytical needs posed by each domain. We organize the components in Table 3 along a taxonomy of three sensemaking processes, as shown in Figure 3. **Analogous to top-down and bottom-up information processing tasks in the sensemaking framework**, in the context of VQSs, analysts can query either directly based on a pattern “in their head” [40] 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. A more detailed articulation of the problem space addressable by VQS and how each sensemaking process fits into this space can be found in Appendix B.

In this section, we first describe the design objectives of each sensemaking process (the top level in Figure 3). Proceeding to the lower level of the Figure 3 taxonomy, we then discuss how each sensemaking process is comprised of functional components that address the problem and dataset characteristics of each domain. **Usage scenarios from Table 4 exemplifies how each sensemaking process supports essential subtasks and enables participants’ scientific goals**. For reference, the mapping between specific *zenvisage++* features and these components and processes can be found in Table 3.

Sensemaking Process

Domain	Sensemaking Process		
	Top-Down Pattern Search	Context Creation	Bottom-Up Data-Driven Inquiry
	<b>Astronomy</b> <b>Goal:</b> Discover potential supernovae candidates that exhibits peak-then-decay pattern.	<b>Support:</b> Examine data regions that are more likely to have supernovae candidates.	<b>Support:</b> Identify and eliminate sources of data anomaly to improve match accuracy for finding candidates.
	<b>Material Science</b> <b>Support:</b> Find data classes that follows desired functional pattern to understand which solvent types exhibit certain tradeoffs and relationships.	<b>Goal:</b> Compare characteristics from different data classes to find a solvent (datapoint) that satisfies desirable properties.	<b>Support:</b> Understand the overall tradeoffs and relationships between data attributes.
<b>Genetics</b>	<b>Support:</b> Search and browse for genes belonging to the same cluster.	<b>Support:</b> Compare genes belonging to different clusters and their known properties.	<b>Goal:</b> Understand characteristic pattern profiles in dataset.

Table 4: Each VQS sensemaking process maps to scientific tasks and goals from each use case, from pattern search to comparing visualization collections to gaining overall data understanding. We find that our scientific 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 operation, they make use of the two other sensemaking processes to support them in accomplishing their main goal.

### 6.1 Top-Down Pattern Search

Top-down processes are “goal-oriented” tasks that makes use of “analysis or re-evaluation of theories [and] hypotheses [to] generate new searches” [36]. 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 characteristics of the pattern query (e.g., patterns with specific shape characteristics, or expressible in 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 x,y invariant), adjusting these knobs helps 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 performs top-down pattern search by querying for transient patterns through sketching and adjusting the match criterion by choosing to ignore differences along the temporal dimension and changing the similarity metric for flexible matching.



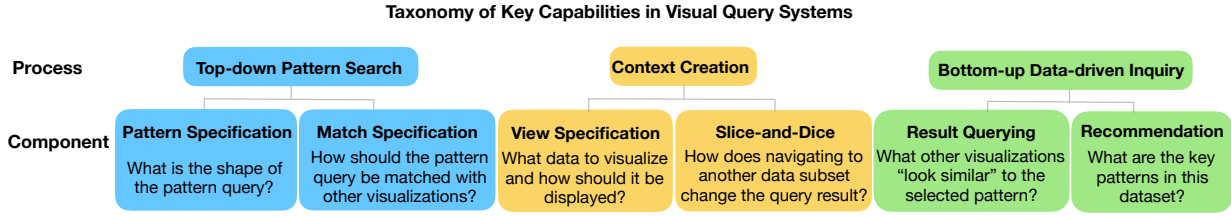


Fig. 3: Taxonomy of key capabilities essential to VQSS. Each of the three sensemaking process is broken down into key functional components in VQSS. Each component addresses a pro-forma question from a system’s perspective.

## 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*” [36]. 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 data-driven 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 results 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 engages with bottom-up data-driven inquiries to learn about the characteristic patterns that exist in the dataset through representative trends, as a means to jump-start further queries via result querying, as well as to 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 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 subselections of the data. Data navigation capabilities is 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, 33]. 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 VQSS, our design objective is to enable all three sensemaking processes in *zenvisage++*. Our Section 7 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 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 tools, detailed in Appendix C. 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 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 [49], 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 based on 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 encounter 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 is hard to draw by hand.

Given these initial findings, we further investigated the process that analyst engaged in to construct pattern queries, as presented in Figure 4. Pattern queries can be generated by either top-down (sketching) or bottom-up (drag-and-drop) processes, driven by various different querying intentions. Within top-down processes, a pattern query could arise from users directly sketching a new pattern (sketch-to-query) or by modifying an existing sketch (sketch-to-modify). For example, M2

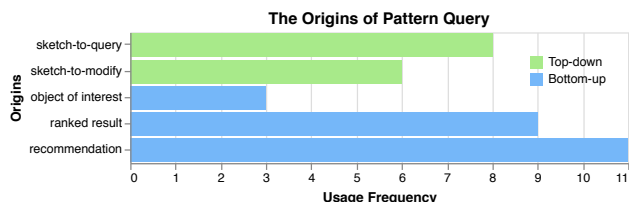


Fig. 4: The number of times a pattern query originates from one of the workflows. We find that pattern queries are far more commonly generated via bottom-up than top-down processes.

first sketched a pattern to find solvent classes with anticorrelated properties 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, Figure 5 (right) illustrates how A3 first picked out a regular pattern (suspected star spot), then modified it slightly so that the pattern looks more irregular (to find pulsating stars). As described in the following section, 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. Figure 4 shows that *bottom-up processes are more common than top-down processes for generating a pattern query*.

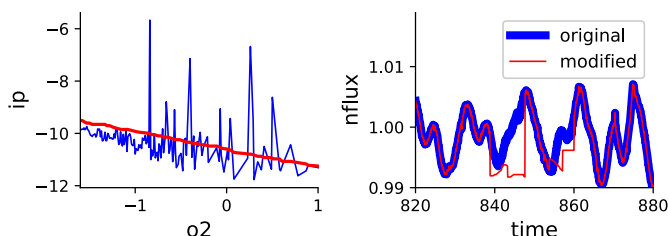


Fig. 5: Example of sketch-to-modify, based on canvas traces from M2 (left) and A3 (right). The original drag-and-dropped query is shown in blue and sketch-modified queries in red.

The lack of practical use of top-down pattern specification is also reflected in the fact that none of the participants queried using an equation. In both astronomy and genetics, the visualization patterns resulting from complex physical processes that could not be written down as an equation 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.

Our findings suggest that while sketching is a useful construct for people to express their queries, *the existing ad-hoc, sketch-only model for VQSs is insufficient on its own* without data examples that can help analysts jumpstart their exploration. In fact, from Figure 4, we can see that sketch-to-query was only used 8 times, while the remaining querying modalities were used 29 times altogether, more than three times as much as sketch-to-query. This finding has profound implications on the design of future VQSs, since Table 1 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 VQSs are not commonly adopted in practice. **Note that we are not advocating for removing the sketch capabilities from future VQSs completely, but instead focusing future research and design efforts to examine other (often overlooked) VQS sensemaking processes that could be used 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 it was useful for

70% of the use cases for bottom-up approaches and 67% for context creation<sup>3</sup>. We now highlight some exemplary workflows demonstrating the efficacy of the latter two sensemaking processes.

As shown in Figure 4, 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  $\nearrow$ ), repressed genes (starting high then decreasing  $\searrow$ ), and transients (rising first then dropping at another time point  $\nearrow\searrow$ ). 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 in *zenvisage++*, 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.

Past studies in visual analytics have shown that it is important to design features that enable users to select relevant subsets of data [1, 17]. Context creation in VQSs 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. 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 enabled them to test conditions and tune values that they would not have otherwise modified, 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) or *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, as shown in Figure 4. 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 further investigate the interplay between these sensemaking processes in the context of an analysis workflow. The event sequences from the evaluation study consist of

<sup>3</sup>See Appendix C for details on how this measure was computed.



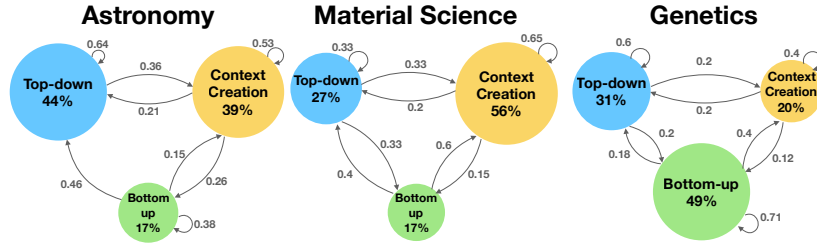


Fig. 6: Markov models computed based on evaluation study event sequences, with edges denoting the probability that participant in the particular domain will go from one sensemaking process to the next. Nodes are scaled according to their eigenvector centrality, representing the percentage of time participants would spend in a particular sensemaking process in steady state.

labels describing when specific features were used. Using the taxonomy in Table 3, 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 each domain move between different sensemaking processes. For example, in material science, bottom-up exploration leads to context creation 60% of the time 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, it is followed by another top-down process 64% of the time and context creation the rest of the time, but never followed by a bottom-up processes. 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 [35]. 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, as illustrated in Appendix Table 4. For example, 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 VQSs 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 a sketch-only system as a baseline to validate this hypothesis. The goal of our study is to uncover qualitative insights that might reveal why VQSs 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 VQSs to-date, covering many of the features from past system and much more (as evident from Table 1). Moreover, 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. While we have generalized our findings by employing three different and diverse domains (see Figure 6), our case studies have so far been focused on scientific data analysis with domain-experts, as a first step towards greater adoption of VQSs. Other potential domains that could benefit from VQSs 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 VQSs 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 VQSs can address their analytic challenges, collaboratively design VQS capabilities, and evaluate how VQSs 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 VQSs. 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 VQSs should invest in understanding and supporting all three sensemaking processes to effectively “close the loop” in how analysts interact and perform sensemaking with VQSs. While more work certainly remains to be done, by contributing to a better understanding of how VQSs are used in practice across domains, our paper can serve as a roadmap towards the broader adoption of VQSs for novel future use cases.

#### 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. Elmqvist. 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. doi: 10.1109/VAST.2016.7883519
- [11] Drlica Wagner et al. Dark Energy Survey Year 1 Results: Photometric Data Set for Cosmology. 2017.
- [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 evaluation methods for Information Visualization - BELIV '08*, p. 1, 2008. doi: 10.1145/1377966.1377974
- [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. 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] M. Mohebbi, D. Vanderkam, J. Kodysh, R. Schonberger, H. Choi, and S. Kumar. Google correlate whitepaper. 2011.
- [29] M. J. Muller and S. Kogan. Grounded Theory Method in HCI and CSCW. *Human Computer Interaction Handbook*, pp. 1003–1024, 2012.
- [30] M. J. Muller and S. Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, June 1993. doi: 10.1145/153571.255960
- [31] 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
- [32] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. doi: 10.1109/MCG.2006.70
- [33] C. Olston and E. H. Chi. ScentTrails. *ACM Transactions on Computer-Human Interaction*, 10(3):177–197, 2003. doi: 10.1145/937549.937550
- [34] 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
- [35] B. Pierre. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 2011.
- [36] 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.
- [37] 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
- [38] 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
- [39] 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
- [40] 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
- [41] 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
- [42] 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
- [43] 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.
- [44] 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.
- [45] Susanne Bodker and K. Grønbaek. Cooperative Prototyping -. *International Journal of man-machine studies*, pp. 1–23, 1991.
- [46] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1970.
- [47] 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.

- [48] 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
- [49] 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
- [50] J. S. Yi, Y.-A. Kang, J. T. Stasko, and J. A. Jacko. Understanding and characterizing insights: How Do People Gain Insights Using Information Visualization? *Proceedings of the 2008 conference on BEyond time and errors novel evaluation methods for Information Visualization - BELIV '08*, p. 1, 2008. doi: 10.1145/1377966.1377971

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 VQSs 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 external tools for conducting analysis in their existing workflow, as shown in Figure 7, including:

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

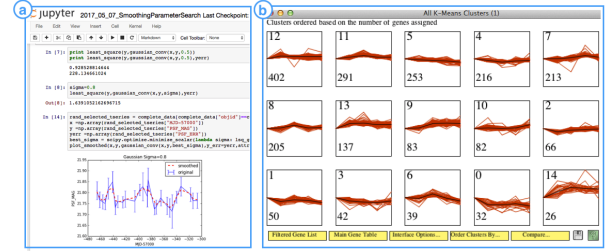


Fig. 7: Screenshots from contextual inquiry: a) A1 examines a light curve manually using the Jupyter notebook environment, b) G2 uses a domain-specific software to examine clustering outputs.

Our collaboration with participants is illustrated in Figure 8, where we began with an existing VQS (*zenvisage*, as illustrated in Figure 9) and incrementally incorporated features, such as dynamic class creation (Figure 10), throughout the PD process.

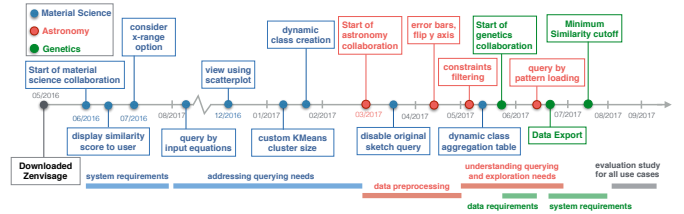


Fig. 8: Timeline for progress in participatory design studies.

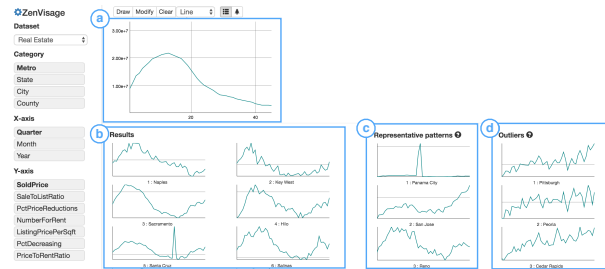


Fig. 9: The existing *zenvisage* prototype allowed users to sketch a pattern in (a), which would then return (b) results that had the closest Euclidean distance from the sketched pattern. The system also displays (c) representative patterns obtained through K-Means clustering and (d) outlier patterns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

As discussed in Section 5.1, not all of the features proposed by participants during PD were incorporated in the *zenvisage++* prototype. 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.



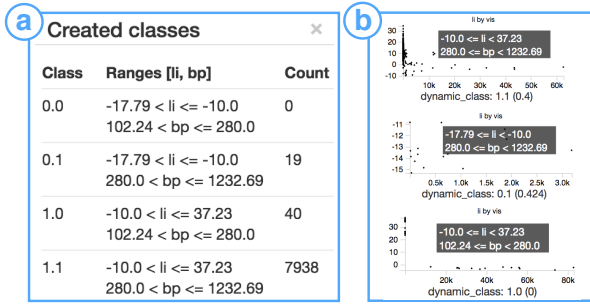


Fig. 10: Example of dynamic classes. (a) Four different classes with different Lithium solvation energies (li) and boiling point (bp) attributes based on user-defined data ranges. (b) Users can hover over the visualizations for each dynamic class to see the corresponding attribute ranges for each class.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

To this end, we use two criteria to heuristically judge whether to implement a particular feature:

1. *Necessity*: Without this feature, can participants still work with this dataset using the tool and meet their information needs?
  2. *Generality*: Will this feature benefit only this specific use case or be potentially useful for other domains as well?
- “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.
  - 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 casual specification that were underspecified. For example, A1 wanted to look for objects that have deficiency in one band and high emission in another band, but the scientific definition of “deficiency” in terms of brightness levels was ambiguous.

## B CHARACTERIZING THE PROBLEM SPACE FOR VQSs

We now characterize the space of problems addressable by VQSs and describe how each sensemaking process fits into different problem areas that VQSs 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 11. 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 11 grey cell), systems such as Tableau, where analysts manually create and examine each visualization one at a time, is more well-suited than VQSs, 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 ??). On the other hand, *bottom-up data-driven inquiries* (Figure 11 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 11 yellow) is essential for allowing users to pivot across different visualization collections.

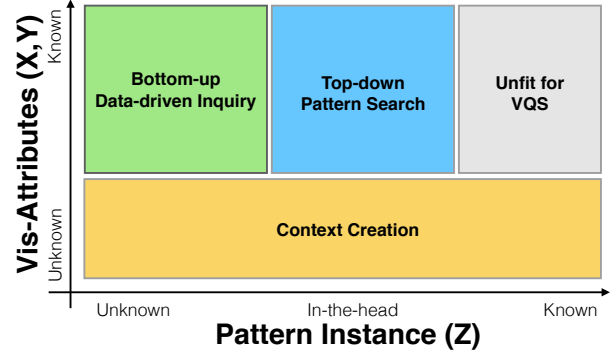


Fig. 11: The problem space for VQSs is characterized by how much the analyst knows about the visualized attributes and the pattern instance. Colored areas highlight the three sensemaking processes in VQSs for addressing these characteristic problems. While prior work has focused solely on use cases in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

## C 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 12. A feature is regarded as *useful* if it has a **P** or **E** code label. Using the matrix from Figure 12, we compute the percentage of useful features for each sensemaking process as:

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

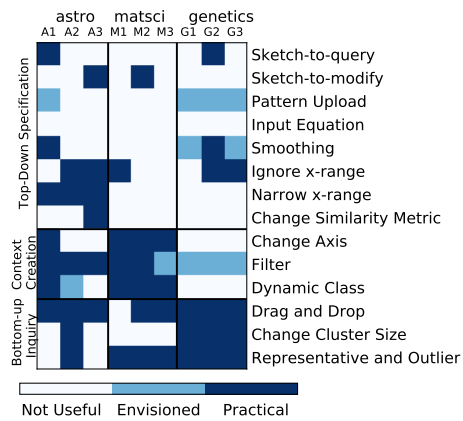


Fig. 12: Heatmap of features categorized as practical usage (P), envisioned usage (E), and not useful (N). Columns are arranged in the order of subject areas and the features are arranged in the order of the three foraging acts. Participants preferred to query using bottom-up methods such as drag-and-drop 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.