Accelerating Scientific Data Exploration via Visual Query System

ABSTRACT

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

KEYWORDS

Visual analytics, visualization, exploratory data analysis, visual query, scientific data.

ACM Reference Format:
. 1997. Accelerating Scientific Data Exploration via Visual Query System. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, 9 pages. **https://doi.org/10.475/123_4**

1 INTRODUCTION

2 RELATED WORKS

3 METHODS

Motivation

Visualization systems are often evaluated using controlled studies that measure the user's performance against an existing visualization baseline [30]. Techniques such as artificially inserting "insights" or setting predefined tasks for example datasets work well for objective tasks, such as debugging data errors [15, 27], but these contrived methods are unsuitable for trying to learn about the types of real-world queries users may want to pose on VQSs. Due to the unrealistic nature of controlled studies, many have proposed using a more multi-faceted, ethnographic approach to understand how analysts perform visual data analysis and reasoning [18, 21, 25, 30, 33]. In order to make the user study more realistic, we opted for a qualitative evaluation where we allowed participants to bring datasets that they have vested interests in to address unanswered research questions. Participatory design has been successfully used in the development of interactive visualization systems in the past [3, 5]. Sedlmair et al. [21] advocate that design study methodology is suitable for use cases in which the data is available for prototyping, but the task is only partially known and the information is partially in the user's head. In that regard, our scientific use cases with VQS is well-suited for a design study methodology, as we learn about the scientist's data and analysis requirements and design interactions that helps users translate their "in-the-head" specifications into actionable visual queries.

Participatory Design

We adopted a mixed methods research methodology that draws inspiration from ethnographic methods, iterative and participatory design, and controlled studies [22, 24, 33] to understand how VQSs can be used for scientific data analysis. Working with researchers from three different scientific research groups, we identified the needs and challenges of scientific data analysis and the potential opportunities for VQSs, via interviews and cognitive walkthroughs.

We recruited participants by reaching out to research groups via email and word of mouth, who have experienced challenges in dealing with large amounts of data. We initially spoke to analysts from 12 different potential application areas and narrowed down to three use cases in astronomy, genetics, and material science for our participatory design study. Six scientists from three research groups participated in the design of *zenvisage*. On average, the participants had more than 8 years of research experience working in their respective fields.

Given our early conversations with the participants, we built a basic VQS to serve as the functional prototype in the design study. This early VQS prototype allowed users to sketch a pattern or drag-and-drop an existing visualization as a query, then the system would return visualizations that had the closest Euclidean distance from the queried pattern. The details of the system is described in [34, 35], which focused on the system and scalability aspects of the VQSs.

WOODSTOCK'97, July 1997, El Paso, Texas USA
2016. ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

The use of functional prototypes is common in participatory design to provide a starting point for the participants. For example, Ciolfi et al.[6] studied two different alternatives to co-design (starting with open brief versus functional prototype) in the development of museum guidance systems and found that while both approaches were equally fruitful, functional prototypes can make addressing a specific challenge more immediate and focused. Our motivation for providing a functional prototype at the beginning of the participatory design sessions is to showcase capabilities of VQSs. Especially since VQSs are not common in the existing workflows of these scientists, participants may not be able to imagine their use cases without a starting point.

During the participatory design process, we collaborated with each of the teams closely with an average of two meetings per month, where we learned about their datasets, objectives, and how VQSs could help address their research questions. A detailed timeline of our engagement with the participants and the features inspired by their use cases can be found in Figure 1. Participants provided datasets they were exploring from their domain, whereby they had a vested interest in using a VQS to address their own research questions. Through this process, we identified and incorporated more than 20 desired features into the VQS prototype over the period of a year.

Evaluation Study

Finally, we conducted a realistic, qualitative evaluation to study how analysts interact with different VQS components in practice. The evaluation study participants included the six scientists from the participatory design study, along with three additional "blank-slate" participants who had never encountered *zenvisage* before. While participatory design subjects actively provided feedback on *zenvisage* with their data, they only saw us demonstrating their requested features and explaining the system to them, rather than actively using the system on their own. So the evaluation study was the first time that all nine of the participants used *zenvisage* to explore their datasets.

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

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

At the start, participants were provided with an interactive walk-through explaining the details of the features offered in our VQS. The participants were then given approximately ten minutes to experience a guided exploration of our VQS with a preloaded real-estate example dataset from Zillow [1]. After familiarizing themselves with the tool, we loaded the participant's dataset and suggested an appropriate choice of axis to begin the exploration. Participants were encouraged to talk-aloud during the data exploration phase.

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

4 PARTICIPANTS AND DATASETS

During the design study, we observed the participants as they conducted a cognitive walkthrough demonstrating every component of their current data analysis workflow. In this section, we describe our study participants and their use cases to highlight the existing workflow and behavior that participants have adopted for conducting certain analysis tasks.

Astronomy

The Dark Energy Survey (DES) is a multi-institutional project with over 400 scientists. Scientists use a multi-band telescope that takes images of 300 million galaxies over 525 nights to study dark energy[8]. The telescope also focuses on smaller patches of the sky on a weekly interval to discover astrophysical transients (objects whose brightness changes dramatically as a function of time), such as supernova explosions or quasars. The output is a time series of brightness observations associated with each object extracted from the images observed. For over five months, we worked closely with an astronomer on the project's data management team working at a supercomputing facility. The scientific goal is to identify a smaller set of potential candidates that may be astrophysical transients in order to study their properties in more detail.

Participant A1 was interested in *zenvisage* as he recognized how specific pattern queries could help scientists directly search for these rare objects. While an experienced astronomer who has examined many transient light curves can often distinguish an interesting transient object from

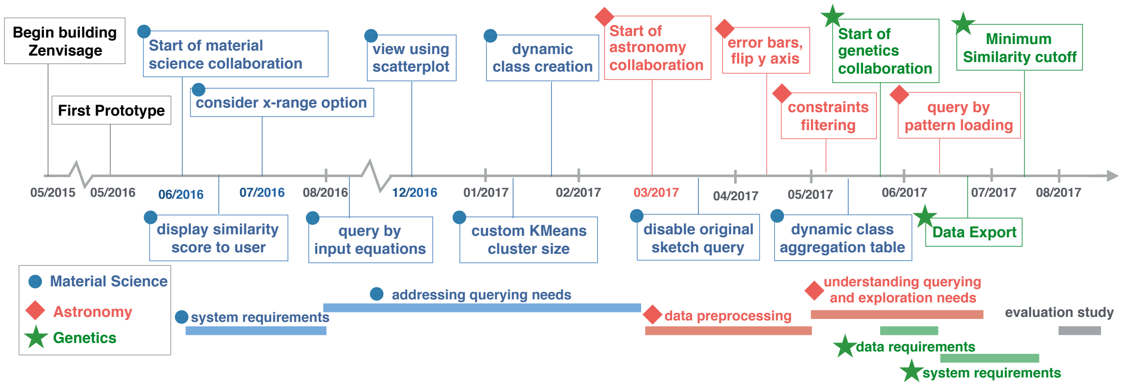

Figure 1: Participatory design timeline for the scientific use cases.

noise by sight, they must visually examine and iterate through large numbers of visualizations of candidate objects. Manual searching is time-consuming and error prone as the large majority of the objects are not astronomical transients.

Genetics

Gene expression is a common data type used in genomics and is obtained via microarray experiments. The data used in the participatory design sessions was the gene expression data over time for mouse stem cells aggregated over multiple experiments.. We worked with a graduate student and a PI at a research university over three months who were using gene expression data to better understand how genes are related to phenotypes expressed during early development [10, 28]. They were interested in using *zenvisage* to cluster gene expression data before conducting analysis with a downstream machine learning workflow.

To analyze the data, participant G1 loads the preprocessed data into a desktop application for visualizing and clustering gene expression data. Participant G1 sets several clustering and visualization parameters before executing the clustering algorithm, then overlaid time series for each cluster is displayed on the interface. G1 visually inspects that all the patterns in each cluster looks "clean" and checks that the number of outlier genes that do not fall into any of the clusters is low. If the number of outliers is high or the visualizations look unclean, she reruns the analysis by increasing the number of clusters. When the visualized clusters look "good enough", G1 exports the cluster patterns into a csv file to be used as features in their downstream regression tasks.

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

Material Science

We collaborated with material scientists at a research university who are working to identify solvents that can improve battery performance and stability. These scientists work with large datasets containing over 25 chemical properties for more than 280,000 different solvents obtained from simulations. We worked closely with a graduate students, a postdoctoral researcher, and a PI for over a year to design a sensible way of exploring their data using VQSs. Each row of their dataset represents a unique solvent, and consists of 25 different chemical attributes. They wanted to use *zenvisage* to identify solvents that not only have similar properties to known solvents but also are more favorable (e.g. cheaper or safer to manufacture), as well as to understand how changes in certain chemical attributes affects them.

Participant M1 starts his data exploration process by iteratively applying filters on a list of potential battery solvents using basic SQL queries. When the remaining list of the solvents is sufficiently small, he examines each solvent in more detail to weigh in the cost and availability to determine experimental feasibility. Participant M1 was interested in using a VQS as it was impossible for him to manually compare between hidden relationships (such as how changing one attribute affects another attribute) between large number of solvents manually.

5 STUDY FINDINGS

Themes Emerging from Participatory Design

We employed participatory design with our scientists to incorporate key features missing in our original VQS, and unaddressed in their existing workflows. We discovered three central themes encapsulating these features that are important to facilitate rapid hypothesis generation and insight discovery,

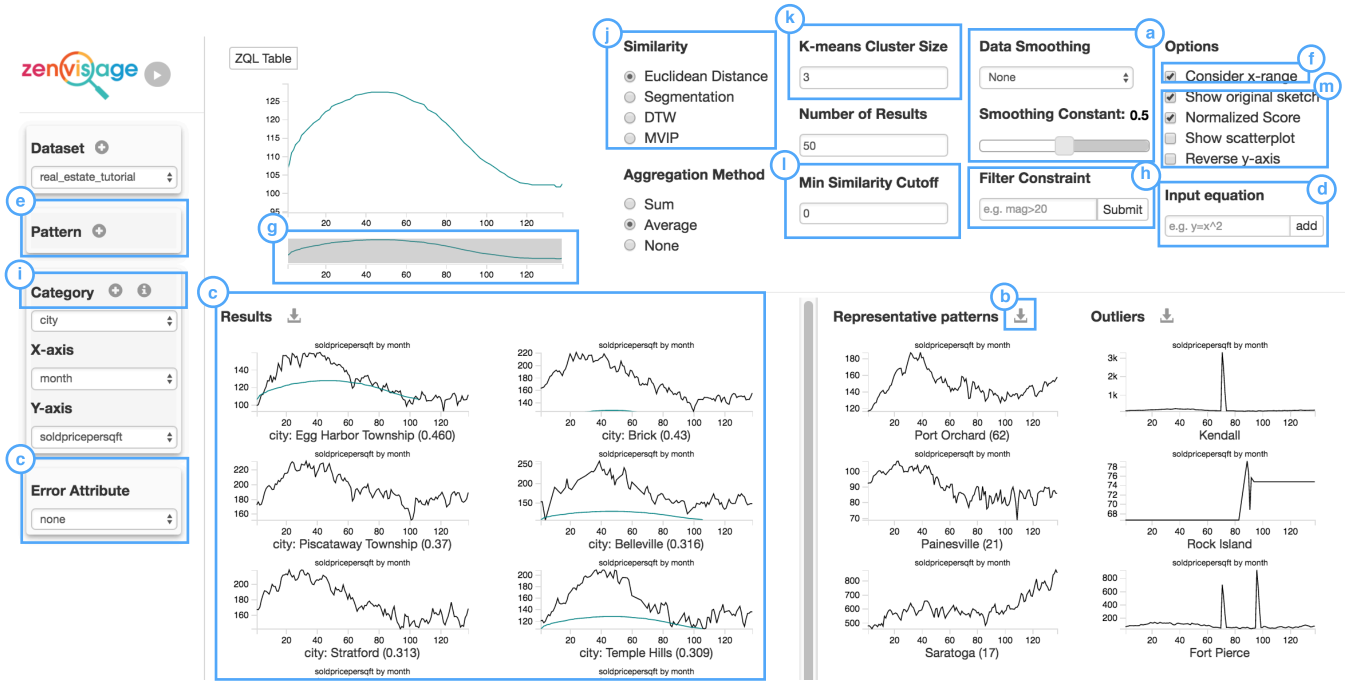

Figure 2: Our VQS after participatory design, which includes: the ability to preprocess via (a) interactive smoothing; (b, c) the ability to export data outputs ; querying functionalities via (d) equations and (e) patterns; query specification mechanisms including (f) x-range invariance, (g) x-range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

but are missing in prior VQSs. While some of our findings echo prior work on system-level taxonomies of visualization tasks [2, 11], we highlight how specific analytic tasks and interaction features could be used to enhance VQSs in particular.

Uninterrupted workflow

Our cognitive walkthroughs revealed that in many participants' existing workflows, they switched between parameter specification, code execution, and visualization comparisons. The non-interactive nature of these segmented workflows has been shown to incur a large cognitive barrier during exploratory data analysis [17]. In addition, since scientific research often takes place in a collaborative setting, this means that the data sense-making process could be delayed by weeks because the analysis-to-results phases needed to be rerun offline based on changes that were suggested during a meeting. Moreover, data-cleaning emerged as a common pain-point, echoing prior work [15, 16].

Integrative preprocessing through interactive smoothing: While *zenvisage* does not attempt to solve all of the preprocessing issues that we faced during participatory design, we identified data smoothing as a common data cleaning procedure that could benefit from a tight integration between pre-processing and visual analysis. Data smoothing is a denoising procedure that generates a smoothed pattern approximating key features of the visualized trend with less noise. Smoothing also raises an interesting trade-off between the smoothness of the curve and the quality of shape-matching for VQSs. If the visualization is over-smoothed, then shape matching would return results that only loosely resemble the query pattern. However, if no smoothing is applied, then the noise may dominate the overall trend, which could also lead to bad pattern matches. In addition, it is often hard to tell what the appropriate smoothing parameter should be applied simply by visualizing a small number of sampled visualization, as one would do in an offline analysis.

To address this issue, we developed an interface for users to interactively adjust the data smoothing algorithm and parameters on-the-fly to update the resulting visualizations accordingly (Figure 2a). This was applied to the material science and astronomy use cases, as both had noisy and dense observational data.

Increasing expressiveness of querying capabilities

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

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

Upload Pattern as Query: While the input equation is useful when simple analytical models exist, this may not be true for other domains. In these cases, users can upload a query pattern of a sequence of points (Figure 2e). This is useful for patterns generated from advanced computational models used for understanding scientific processes, usually as part of the downstream analysis of the exploratory workflow.

Consider/Ignore x-range: We improved query specification by allowing users to change how the shape-matching criterion is applied. For finding supernovae, A1 primarily cared about the existence of a peak above a certain amplitude with an appropriate width of the curve, rather than the exact time that the event occurred, leading them to use the consider x-range feature. G1 also expressed that she does not really know what is the "trigger point" of when the expression level of a gene will rise and it would be interesting to find all "rising" profiles independent of the change-point. We implemented an option to ignore the x-range in shape matching (Figure 2f) and a brushing mechanism that enables users to select the specific x-region they want to perform their shape matching on (Figure 2g).

Ability to dynamically facet through subsets

Past studies in taxonomies of visualization tasks have shown that it is important to design features that enable users to select relevant subsets of data in visual analytics[2, 11]. We designed two dynamic faceting features coupled with coordinated views that enabled users to specify subsets of data they are querying on and see immediate changes updated in the query, representative, and outlier results.

Filtering Constraints: Users with large datasets first used their domain knowledge to narrow down their search to a subset of data. This would increase their chances of finding an interesting pattern for a given query. To filter data, users could submit one or more SQL-like `WHERE` conditions as filter constraints in a text field (Figure 2h).

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

Scientists can utilize multiple properties to create custom classes, effectively slicing-and-dicing the data based on their needs. The information regarding the created classes is displayed in the dynamic class information table or as a tooltip over the aggregated visualizations, as shown in Figure 3.

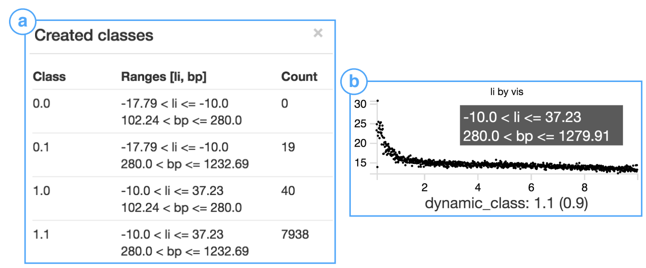

Figure 3: Example of dynamic classes. (a) Four different classes with different Lithium solvation energies (li) and boiling point (bp) attributes based on user-defined data ranges. (b) Users can hover over the visualizations for each dynamic class to see the corresponding attribute ranges for each class. The visualizations of dynamic classes are aggregate across all the visualizations that lie in that class based on the user-selected aggregation method.

Finer System-level Control and Understanding

During the participatory design exercise, we found that many of the features suggested by the participants indicated they wanted finer control of the system. Prior work in direct manipulation visual interfaces has suggested that finer-grained control enabled users to discover patterns and rapidly generate hypothesis based on visual feedback [4, 32].

Controlling VQS internals: In addition to query and dataset specifications, users also wanted the ability to modify the model parameters in *zenvisage*. Our findings echoed Chuang et al. [5], which showed that the ability to modify the model can facilitate interpretation and trust in model-driven visualizations, especially during early-stage exploration. These model parameter options include the ability to change the choice of similarity metrics (Figure 2j), the cluster size in the representative patterns (Figure 2k), setting a minimum similarity threshold for displaying the search results (Figure 2l), and the ability to tune the smoothing algorithm and parameter (Figure 2a).

Displaying interpretable explanations for VQS recommendations: Explanatory system outputs include displaying similarity scores of the outputs, the number of datapoints in each cluster, and overlaying the original query sketch on the return visualization for comparison (Figure 2m). We further provided display-related options for plotting modifications, including displaying error bars, and toggling between a scatterplot and line chart view, to help analysts better understand the visualizations.

Final Evaluation Study Results

We recorded audio, video screen captures, and click-stream logs of the participant's actions during the evaluation study. We analyzed the transcriptions of these recordings through open-coding and categorized every event in the user study. In addition, based on how each feature was used during the user study, we categorized the features into one of the three usage types:

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

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

The audio recordings and transcriptions of pre- and post-study interview questions are thematically encoded and summarized in Figure **??** and 4. Overall, we find that VQSs can enable rapid, fluid iteration, catalyzing new questions or insights; that different querying modalities in VQSs support different forms of exploration; and that expressive querying allowed participants to compose novel analysis patterns. In addition, we find that VQSs can be used for a range of tasks that go beyond just exploration; that participants used the outputs from VQSs in various ways; and that VQSs are most appropriate for certain types of datasets. For the remaining paper, we will focus on developing a process model and design guideline for insight formation in VQSs and divert our thematic analysis of how VQSs fit into the context of an analysis workflow to our technical report.

These observation inform our —– search-browse paradigm

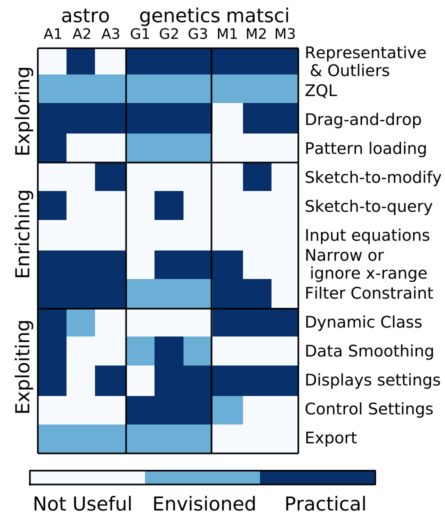

Figure 4: Heatmap of features categorized as practical usage (P), envisioned usage (E), and not useful (N). We find that participants preferred to query using bottom-up methods such as drag-and-drop over top-down approaches such as sketching or input equations. Participants found that data faceting via filter constraints and dynamic class creation were powerful ways to compare between subgroups or filtered subsets. The columns are arranged in the order of subject areas and the features are arranged in the order of the three foraging acts.

6 DESIGN GUIDELINES

In this section, we present design guidelines for key components of a visual query system, drawing from our participatory design experience, evaluation study, and literature review in this space. Moreover, we organize our findings in a process model for visual query systems.

Search-Browse Paradigm

- What does the act of browsing and searching mean in the context of VQSs - browse: viewing ranked result and any recommended results on the side, derived from the data and analysis context. - search: act of going from a user's in-the-head concept to an actionable query that could be executed through the VQSs, most work have focussed on sketch, we allow more than this. - The challenge of browsing and searching is well-known in information retrieval [26], browse alone is limited by how much a user can browse and process at once, search alone can be ambiguous without sufficient context from looking at example results.

To contextualize our study results with respect to prior work on how analysts make sense of data, we employ Pirolli and Card's [29] information foraging framework for domain-experts. Pirolli and Card's notional model distinguishes between information processing tasks that are *top-down* (from theory to data) and *bottom-up* (from data to theory). Recent

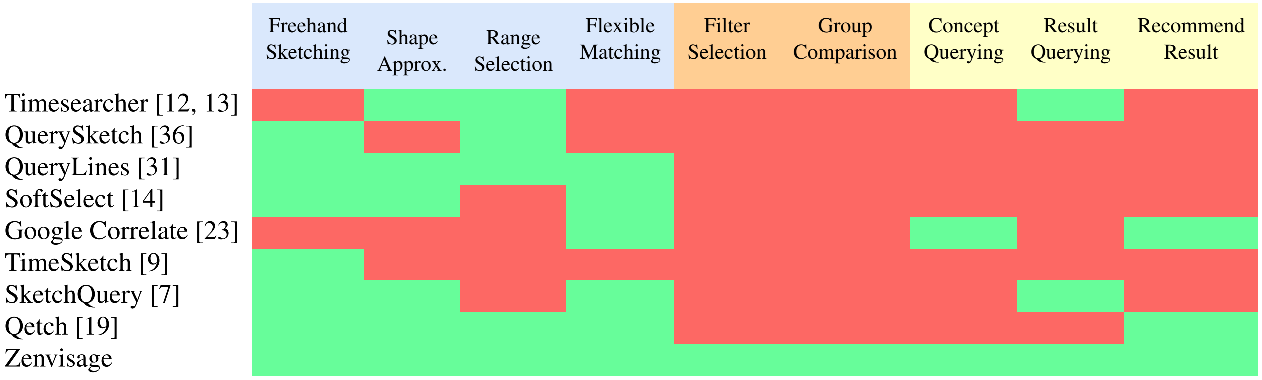

Figure 5

work have also used the top-down v.s. bottom-up framework in understanding visualization construction[20]. In the context of visualization querying, top-down approaches are query specification based on a user's preconceived notion of what to search for, whereas bottom-up approaches are queries that originate from the data (or equivalently, the visualization).

Pirolli and Card's notional model further characterizes the trade-offs between three central activities in the information foraging process: exploring, enriching, and exploiting [29]. *Exploring* involves gathering more information during the analysis. In the context of VQSs, exploring includes viewing representatives and outliers, incidental viewing of other visualizations in the ranked search results, and querying via drag-and-drop and pattern-loading. *Enriching* involves tasks that narrow down the space of analysis, such as filtering, dynamic class creation, query specification, and querying via input equations and sketching. *Exploiting* involves spending time inspecting the results in more detail, including interpreting each visualization in greater detail or making plotting changes that offer another perspective (smoothing, display, and interpretability settings). We organize the features that we have developed in *zenvisage* into these foraging acts, as shown in Figure 4.

We find that participants often create unexpected workflows that chain together multiple analysis steps, including interactions, controls, and queries in order to address a higher-level research question. We find that participants often construct a central workflow, which they then iterate on while adding additional variations. Their *central workflow often resembles one of the three foraging acts* that aligns with the type of research question and dataset they are interested in. The variations are based on intermixing their central workflow with the other two foraging acts.

As illustrated in Figure 5, our search-browse paradigm is motivated by the characteristic challenges and foraging acts each use cases pose on existing VQSs observed in our design study. For example, the genetics participants do not have a preconceived knowledge of what they want to search for in the dataset. They were mostly interested in *exploring* clusters to gain an overall sense what profiles exist in the dataset through representative trends and therefore queried mainly through drag-and-drop to jumpstart further queries. Point to need for D3 and D4. The variations to their main workflow include changing cluster sizes and display settings to offer them different perspectives on the dataset (*exploit*) and filtering on data attributes (*enriching*).

In the astronomy use case, the participants knew the patterns they are looking for, but the patterns are hard to specify and find. The main challenge for the VQS involves finer specification of sketched patterns, such as amplitude and width of the peak and noise level tolerance for defining a pattern match. Describe more in D1. The main workflow for the astronomers in our user study involves *enriching*, either through finer query specification or via filtering data subsets, to increase the probability that their queries would be more accurately matched with what they are looking for.

The main workflow for material scientists involves *exploiting*, since they spend the majority of their efforts performing "close-reading" of individual visualizations to understand the relationships between physical variables. The participants are able to identify interesting relationships between physical variables when they examine each closely, but they are not sure what patterns to look for to begin with. More in D2.

Top-down approaches

D1: Increasing Control and Flexibility.

D2: Searching with Context.

Bottom-up approaches

D3: The Need for Recommendation in VQSs.

D4: Closing the loop: query through browsing results.

7 CONCLUSION

REFERENCES

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

[2] Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. IEEE, 111–117. `https://doi.org/10.1109/INFOVIS.2005.24`

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

[4] B. Shneiderman. 2007. Creativity Support Tools: Accelerating Discovery and Innovation. *Commun. ACM* 50, 12 (2007), 20. `https://doi.org/10.1145/1323688.1323689`

[5] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. 2012. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 443–452. `https://doi.org/10.1145/2207676.2207738`

[6] Ciolfi et al. 2016. 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* (2016), 13–25. `https://doi.org/10.1145/2818048.2819967`

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

[8] Drlica Wagner et al. 2017. Dark Energy Survey Year 1 Results: Photometric Data Set for Cosmology. (2017). arXiv:1708.01531

[9] Philipp Eichmann and Emanuel Zgraggen. 2015. Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings. *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15* (2015), 28–37. `https://doi.org/10.1145/2678025.2701379`

[10] Brian S. Gloss, Bethany Signal, Seth W. Cheetham, Franziska Gruhl, Dominik C. Kaczorowski, Andrew C. Perkins, and Marcel E. Dinger. 2017. High resolution temporal transcriptomics of mouse embryoid body development reveals complex expression dynamics of coding and noncoding loci. *Scientific Reports* 7, 1 (2017), 6731. `https://doi.org/10.1038/s41598-017-06110-5`

[11] Jeffrey Heer and Ben Shneiderman. 2012. A taxonomy of tools that support the fluent and flexible use of visualizations. *Interactive Dynamics for Visual Analysis* 10 (2012), 1–26. `https://doi.org/10.1145/2133416.2146416`

[12] Harry Hochheiser and Ben Shneiderman. 2001. Interactive Exploration of Time Series Data. In *Discovery Science*, Klaus P. Jantke and Ayumi Shinohara (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 441–446.

[13] Harry Hochheiser and Ben Shneiderman. 2004. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization* 3, 1 (2004), 1–18.

[14] Christian Holz and Steven Feiner. 2009. Relaxed Selection Techniques for Querying Time-series Graphs. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 213–222. `https://doi.org/10.1145/1622176.1622217`

[15] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3363–3372. `https://doi.org/10.1145/1978942.1979444`

[16] S Kandel, A Paepcke, J M Hellerstein, and J Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE transactions on visualization and computer graphics* 18, 12 (2012), 2917–26. `https://doi.org/10.1109/TVCG.2012.219`

[17] Mary Beth Kery, Amber Horvath, and Brad A Myers. 2017. Variolite: Supporting Exploratory Programming by Data Scientists.. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1265–1276. `https://doi.org/10.1145/3025453.3025626`

[18] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. 2012. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics* 18, 9 (2012), 1520–1536. `https://doi.org/10.1109/TVCG.2011.279`

[19] Miro Mannino and Azza Abouzied. 2018. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. (2018), 1–12. `https://doi.org/10.1145/3173574.3173962`

[20] Gonzalo Gabriel Méndez, Uta Hinrichs, and Miguel Nacenta. 2017. Bottom-up vs . Top-down : Trade-offs in Efficiency , Understanding , Freedom and Creativity with InfoVis Tools. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2017). `https://doi.org/10.1145/3025453.3025942`

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

[22] Delbert C. Miller, Neil J. Salkind, and Delbert C. Miller. 2002. *Handbook of research design and social measurement*. SAGE.

[23] Mohebbi et al. 2011. Google correlate whitepaper. (2011).

[24] Michael J. Muller and Sarah Kuhn. 1993. Participatory Design. *Commun. ACM* 36, 6 (June 1993), 24–28. `https://doi.org/10.1145/153571.255960`

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

[26] Christopher Olston and Ed H. Chi. 2003. ScentTrails: Integrating Browsing and Searching on the Web. *ACM Transactions on Computer-Human Interaction* 10, 3 (2003), 177–197. `https://doi.org/10.1145/937549.937550`

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

https://doi.org/10.1145/1866029.1866038

[28] Pei Chen Peng and Saurabh Sinha. 2016. Quantitative modeling of gene expression using DNA shape features of binding sites. *Nucleic Acids Research* 44, 13 (2016), e120. https://doi.org/10.1093/nar/gkw446

[29] Peter Pirolli and Stuart Card. 2005. 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. 2–4.

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

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

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

[33] Ben Shneiderman and Catherine Plaisant. 2006. 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*. ACM, 1–7. https://doi.org/10.1145/1168149.1168158

[34] Tarique Siddiqui, John Lee, Albert Kim, Edward Xue, Chaoran Wang, Yuxuan Zou, Lijin Guo, Changfeng Liu, Xiaofo Yu, Karrie Karahalios, and Aditya Parameswaran. 2017. Fast-Forwarding to Desired Visualizations with zenvisage. (2017). https://doi.org/10.1145/1235

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

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