The Role of Visualization in Genomics Data Analysis Workflows: The Interviews

Sehi L'Yi
Harvard Medical School
sehi_lyi@hms.harvard.edu

Qianwen Wang
Harvard Medical School
qianwen_wang@hms.harvard.edu

Nils Gehlenborg
Harvard Medical School
nils@hms.harvard.edu

2023 IEEE Visualization and Visual Analytics (VIS) | 979-8-3503-2557-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/VIS54172.2023.00029

Abstract

The diversity of genome-mapped data and analysis tasks makes it challenging for a single visualization tool to fulfill all visualization needs. To design a visualization tool that supports various genomics workflows of users, it is critical to first gain insights into the diverse workflows and the limitations of existing genomics tools for supporting them. In this paper, we conducted semi-structured interviews (N=9) to understand the role of visualization in genomics data analysis workflows. Our main goals were to identify various genomics workflows, from data analysis to visual exploration and presentation, and to observe challenges that genomics analysts encounter in these workflows when using existing tools. Through the interviews, we found several unique characteristics of genomics workflows, such as the use of multiple visualization tools and many repetitive tasks, which can significantly affect the overall performance. Based on our findings, we discuss implications for designing effective visualization authoring tools that tightly support genomics workflows, such as supporting automation and reproducibility.

Index Terms: H.5.2 [Information Systems]: Information Interfaces and Presentation (e.g., HCI)—User Interfaces;

1 Introduction

Given the size and complexity of genome-mapped data [17], visualization has been playing a vital role in genomics workflows as evidenced by a noticeable number of visualization tools in the wild [16, 21]. However, the diversity of genomics data types (e.g., BED, BAM, VCF, and BigWig) and different analysis tasks makes it challenging for a single visualization tool to fulfill all visualization needs [17, 21]. This results in frequent switching between specialized visualization tools in analysis workflows, rendering the analysis process less efficient and effective due to frequent context switching.

There are several genomics visualization tools that are intended to be used for a broad range of use cases and data types. These include genome browsers [4, 13, 23]—template-based [19] visualization authoring tools based on graphical user interfaces (GUIs)—and genomics visualization libraries [8, 14, 17, 18, 33], such as Gosling [17] and ggBio [33] based on a visualization grammar [32]. However, they still lack several important features needed in diverse genomics workflows of different users, such as flexible customization in genome browsers and ease of using visualization libraries without programming experience by domain experts. To design and develop a genomics visualization tool that is tailored to various genomics data analysis workflows, we need a comprehensive understanding of existing workflows and the diverse needs of users. While several design studies identified important analysis tasks from biologists [20], we still lack formative studies that inform visualization designers and researchers to create effective visualization authoring tools for genomics data.

In this paper, we conducted semi-structured user interviews (N=9) to gain a clear and bigger picture of genomics data analysis workflows and the role of visualization in the workflows. Our main goals were to identify the entire workflow around genomics data analysis and also observe challenges in different stages, such as integration support, efficiency, and expressiveness. We found several unique characteristics of genomics workflows, such as the use of multiple visualization tools and many repetitive tasks, which can significantly affect the performance of user tasks. Based on our findings, we discuss implications for designing effective visualization tools that tightly support genomics workflows, such as supporting automation and reproducibility.

2 Related Work

Through literature reviews and interviews with domain experts, research has been conducted to analyze common visualization workflows, including studies focusing on general visualization practices [3, 25, 28], as well as the employment of visualization techniques within specific domains [5, 24, 30]. For example, researchers conducted user interviews to identify visualization workflows and inform the creation of visualization authoring tools. Liu et al. [15] conducted user interviews to identify the workflow of creating visualizations, such as three main frequent tasks (i.e., sketch, arrange, and bind), and discovered that people first think about overall graphics and then visual encoding. Similarly, Satyanarayan and Heer [26] identified a three-phase design process, i.e., exploration, drafting, and production, as well as challenges of the current tools (e.g., no support for non-linear narratives).

Several other studies focused on conducting larger-scale user interviews to understand in-depth visualization workflows. Crisan and Fiore-Gartland [5] conducted semi-structured interviews (N=29) to understand how automatic machine learning systems can help in data science work. Bako et al. [2] performed interviews with 15 university students and 15 professional designers to understand how visualization designers find and use data visualization examples. Kandel et al. [11] also conducted interviews with 35 data analysts from 25 organizations to characterize the process of data analysis in industry settings. Focusing on visualization novices, Grammel et al. [6] identified common visualization processes, as well as challenges of novices in the processes. While these studies analyzed visualization workflows in various contexts, there is a limited understanding of genomics visualization workflows that informs the design of genomics visualization authoring tools.

A relatively small number of studies focused on understanding genomics and, more broadly, biomedical data visualization workflows. Stitz et al. [29] identified five different workflows of ChIP-seq data analysis from a review study [22] and applied them in their provenance graph visualizations. Meyer et al. [20], focusing on pairwise genome comparison (i.e., synteny data), identified a series of key analysis questions through user interviews and literature reviews. However, we still lack the identification of unique characteristics of visualization workflows that take a wide range of genomics file formats and visualization tasks into account.

In this paper, we conducted interviews with genomics analysts in diverse positions and backgrounds to understand unique aspects of

DOI 10.1109/VIS54172.2023.00029

Table 1: **The background of nine interview participants.** The order of visualization tools reflects the frequency of use, where the first tool is the primary tool for the corresponding participant.

ID	Position	Area	Visualization Tools
P1	PhD Student	Genetic Variation	IGV [23], Keynote
P2	PhD Student	scATAC Analysis	IGV [23], PPT, AI
P3	Postdoc	Genetic Variation	Matplotlib [10], IGV [23]
P4	Postdoc	Structural Variation	IGV [23], ggPlot [31]
P5	Senior Scientist	Structural Variation	IGV [23], Circos [14]
P6	Software Engineer	Methylation	GViz [8], IGV [23]
P7	PhD Student	Genetic Variation	IGV [23]
P8	PhD Student	Structural Variation	Circos [14], IGV [23]
P9	Postdoc	Structural Variation	IGV [23], Circos [14]

genomics workflows and identify limitations of existing tools, which can inform the design of visualization authoring tools for genomics data.

3 The Interview

The main goals of our semi-structured user interviews were to (1) identify various workflows for analyzing and visualizing genomics data (from data processing to visual exploration and communication) and (2) understand the challenges of existing genomics tools for supporting these workflows.

3.1 Participants

We recruited a total of nine interview participants (Table 1) from universities and research institutes through recruitment emails and Slack messages. The participants were in various positions, including four PhD students, three postdocs, one senior scientist, and one software engineer. All participants reported that they frequently analyze and visualize genomics data, at least once a week. The participants commonly said that Integrative Genomics Viewer (IGV) [23] (standalone genome browser) is the most frequently used tool while many participants also reported using other tools, such as genomics tools (e.g., Circos [14] and GViz [8]) and general-purpose tools (e.g., Matplotlib [10] and ggplot2 [31]). Most participants reported having experience using programming languages (e.g., R and Python) except one participant (P8).

3.2 Procedure

The semi-structured remote interviews using Zoom were structured with three main sessions: (i) participants' backgrounds (approx. 10 min), (ii) workflows (30 min), and (iii) challenges (20 min) using existing tools. We asked participants to fill out a questionnaire before the interview to collect background information of participants, such as current roles, areas of study, and computational skills. During the second session, we tried to understand how participants analyze their data using existing genomics tools. Example questions for this session include *"How do you analyze your data using an existing tool?"*, *"How do you process data?"*, and *"How do you construct your visualization?"* In this session, an interviewer (the first author), together with participants, drew flow charts that accurately illustrate participants' workflows using Zoom's collaborative Whiteboard[1]. After the identification of workflows, we asked participants to fill out questionnaires to document their experience using visualization tools in their workflows. The questionnaires contained 7-point Likert scale questions that assess visualization tools in eight different aspects adopted from Amini et al.'s work [1], such as expressiveness, integration, and flexibility. Using the responses, as well as the flow charts, in the third session, we asked follow-up questions to identify challenges participants usually encountered at each stage of the workflows when using existing visualization tools. For example, if we found the participants strongly disagreed that certain aspects are supported in a visualization tool, we asked what made them think in that way (e.g., *"What made you think that IGV does not support flexible customization?"*). All participants were compensated with $25 Amazon Gift Cards. The entire interview per participant took about an hour. The semi-structured questions and questionnaire questions used for the interviews are available in the supplemental materials.

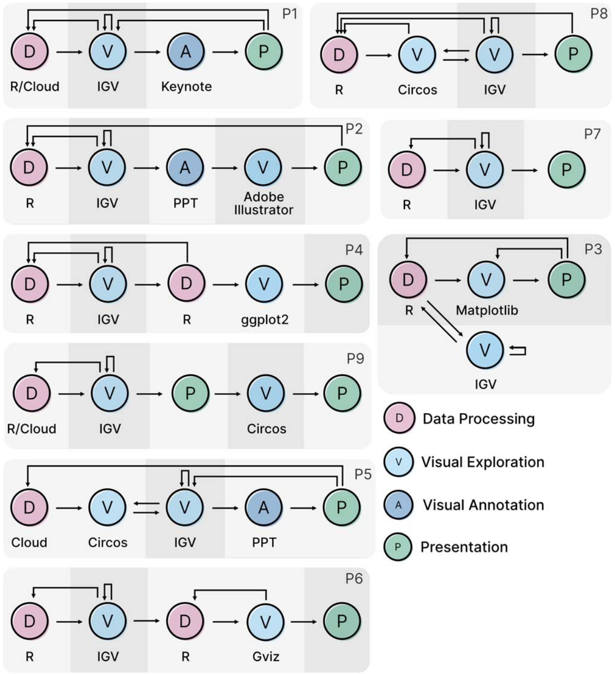

Figure 1: **The summary of workflows of all interview participants.** These flow charts are drawn collaboratively with participants during the interviews, reflecting their everyday workflows. Only the cyclic events that happen frequently are drawn as backward arrows between stages. The names of the tools used in individual stages are labeled right below circular nodes, where R refers to the programming language. The same background color between nearby nodes represents that corresponding stages are performed in the same environment by the participant. The participants are ordered in a way that participants with the same positions are closer to each other (e.g., P1, P2, P7, and P8 who are postdocs).

3.3 Analysis

After the interviews, we collected flow charts that we drew with participants, as well as transcript notes to analyze interview results. We identified diverse workflows as visually summarized in Fig. 1. Individual circular nodes in this figure represent one of the four main steps: (D) data preparation and processing, (V) visual exploration, (A) visual annotation (e.g., adding arrows and labels to existing visualizations for the purpose of presentation), and (P) presentation and communication. The tools that the participants used for each step, such as R programming languages and IGV [23], are labeled right under the node. Arrows represent frequent transitions between stages, where the self cycles in visualization stages refer to user interactions in interactive visualization tools.

Various challenges participants encountered at each stage of workflows are shown in Fig. 2, where black circle nodes represent types of challenges adopted from a previous study [1], where similar categories are grouped together for simplicity and clarity: (C) refers

[1] https://explore.zoom.us/en/products/online-whiteboard/

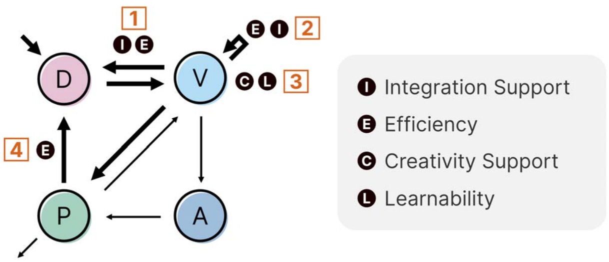

Figure 2: **Various challenges that participants encounter in their workflows.** The thicker arrows represent that the corresponding transitions between stages are frequently observed in our interviews. The black circle labels represent the challenges we identified.

to *creativity support*, *expressiveness*, and *flexibility*, (L) refers to *learnability*, *intuitiveness*, and *guidance*, (I) refers to *integration* (e.g., supporting a series of tasks and the transition between tools) and (E) refers to *efficiency*.

4 Findings

In this section, we describe our main findings, including unique characteristics of genomics data analysis workflows and limitations of existing genomics tools to support their workflows.

4.1 Use of Multiple (Visualization) Tools

All participants relied on multiple tools and environments in their workflows, as can be seen by the various stages labeled with different tools in Fig. 1. Interestingly, **almost 90% of participants used multiple visualization tools**; 8 out of 9 participants used two or three visualization (or graphic design) tools in their workflows. This observation underscores the absence of a single visualization tool that can fully satisfy the diverse user needs throughout the genomics analysis workflow.

We found distinct purposes that were satisfied by different visualization tools. A genome browser [23] was mainly used to confirm and validate their analysis which involves the least customization. Static visualization tools [10, 14, 31], on the other hand, were used to (i) complement the genome browser in visual exploration by providing different perspectives (e.g., overview [14] or summary [10]) and/or (ii) help to generate visualizations for presentation and publication. For example, three participants (P5, P8, P9) used multiple tools in combination to analyze the same data at different scales. P5 used static visualizations generated using Circos [14] to see an overview of large events (e.g., rearrangement of sequences across chromosomes) and used interactive tool [23] to see corresponding regions in a more detailed manner. However, using two tools together was *"clunky"* accordingly to P5 (Fig. 2-2). Circos showed the whole genome, and since it is a static visualization, P5 was not able to accurately infer the exact positions of important patterns, resulting in using the interactive tool to find the positions by trial and error (i.e., using manual navigation and comparing visualizations between Circos and IGV). Several other participants (5 out of 9) used additional tools for making presentation-purpose visualizations, such as matplotlib [10], ggplot2 [31], and Gviz [8].

Tendencies to use multiple visualization tools seem to be mainly due to the limited expressiveness (see Sect. 4.3) of the genome browser and its lack of support for data analysis environments, such as the R programming language. P4 said, although ggplot2 [31] does not natively support genome-mapped data, they use ggplot2 since it is *"tailored for workflows."*

Context switching also happened frequently between the visualization and data processing stages (Fig. 2-1). For example, P7 used the R programming language to perform data analysis (e.g., variant calling) and used an interactive visualization tool IGV [23] to explore the results and capture a screenshot for their publication. However, due to the large size and number of files to analyze (1,000 files of 300MB each), P7 usually stores and analyzes all files on a cloud computing service that does not support IGV and then transfers files to a local computer whenever they want to visualize their analysis results. P7 expressed that the transition between the two stages requires a *"mental hurdle."* Combined with the limited space of a local computer, P7 said they usually end up visualizing only a small number of all 1,000 files, which is not ideal.

4.2 Manual Repetitions and Iterations

We found that **all workflows are heavily iterative and involve many repetitive tasks** (i.e., cyclic arrows in Fig. 1), which can significantly impact the efficiency of analysis workflows. This seems to be an even more critical characteristic given that participants found the interactive visualization tool [23] to be *"lacking reproducibility and automation"* (P7).

We identified these patterns at multiple stages and different levels in the workflows. Within a visualization stage, participants usually zoom into multiple regions of interest repeatedly (i.e., self-cycle in Fig. 1 and Fig. 2-2) after visualizing their datasets in IGV [23] to confirm their hypothesis within the data. These happen for 5–7 genes for P2, 5–10 genes for P3, and all annotated regions for P5. Such tasks are usually heavily manual for most of the participants in a genome browser since they had to search for gene names (e.g., `MYC`) or exact chromosome regions (e.g., `chr8:128,746,972-128,755,021`) and/or click on zooming and panning buttons. P2 reported such interactions to be tedious, saying that these involve *"a lot of clicks."* Other frequent repetitions were replacing datasets from the visualizations and performing similar tasks (i.e., manual navigation to the same regions) (Fig. 2-1). Many participants reported having a large number of files for such repetitive tasks (e.g., thousands of files for P4, P7, and P9).

Making this even more painful, several participants reported that they needed to occasionally go all the way back to the initial stages (data processing) after communicating their analysis results with their teams (Fig. 2-4). Given the nature of interdisciplinary work, this seems inevitable in genomics data analysis. For example, P5 usually shares their findings with colleagues in different backgrounds, such as students, clinicians, and scientists, and had to occasionally reperform the entire tasks: *"We found that not all structural variants are there, so we had to rerun the analysis with different callers."*

4.3 Limited Flexibility in Genome Browsers

Participants complained about limited flexibility in customizing visualization in a genome browser [23] (Fig. 2-3). This issue is tightly connected to the template-based approach used in tool [19], which hard-code visualization types by genomics file types (e.g., bar charts for BigWig files [12]). P6 said *"There is not much you can do with the file. ... There is only a very small handful of ways to visualize it."* Also, several participants did not appreciate the aesthetic aspect of default visualization designs, stating that they are *"ugly"* (P2, P3, P8). The main reason for P3 to use matplotlib [10] over the genome browser was to create *"appealing"* visualizations. The limited flexibility also applies to the flexibility in transforming datasets: *"Ideally, I wanted to visualize the methylation directly on the individual reads. However, this was not possible in IGV. ... so, I processed the files myself." (P5).*

4.4 Barriers to Using Visualization Tools

We noticed diverse barriers to using visualization tools (Fig. 2-3). A matplotlib [10] user said *"I cannot imagine using it without examples and documentation"* (P3). While many participants appreciated the GUI, we also found several challenges in learning and using the genome browser [23]. For example, P9 mentioned that zooming in and out—which is a common interaction in many genome browsers—is *"not that straightforward."* P2 also stated that there is *"a lot of remembering"* involved. Another interesting aspect is that many participants do not seem to be aware of the *Regions* feature that P6 mentioned (i.e., a specialized batch script for automatic navigation), resulting in constant manual navigation. Another challenge stems from the complexity of genomics visualizations, making it difficult for junior researchers (P2, P7) to interpret visualization. Focusing on a default read alignment visualization in IGV [23], P2 said there is *"too much color you need to understand."* Similarly, P7 said, *"I was not sure how to interpret the reads. ... I was not very clear about colors."*

4.5 Additional Suggestions

Participants also suggested several additional features that could make visualization tools more useful for their workflows, including comparative visualizations. Given the nature of collaborative work, P3 emphasized that enabling collaborative visualization authoring would be useful. P2 said since they frequently share their insights from the visualization with their team, it would be good to provide features for better communication, such as easy video recording user interactions and sharing them with others, such as seeing detailed information with tooltips. P4 mentioned that one of the lacking features is to compare multiple samples at once.

5 Design Implications

Reflecting on findings from the interviews, we discuss several important aspects that are valuable to consider during the design of visualization authoring tools for genomics data. We discuss these not only considering the visualization tools used by interview participants but also other recent genomics visualization tools since these tools also lack the same aspects we found through interviews.

Support Automation and Reproducibility. The biggest pain points we found in genomics workflows are that they are highly repetitive and iterative, and we witness that many tasks in the workflows can be (and need to be) automated and streamlined to increase the efficiency and reproducibility of overall workflows. For example, given that the same visualization encodings are used multiple times for different datasets, the visualization options that are lastly used can be recorded and automatically applied to the new datasets. The provenance of interactions (such as exact locations that users viewed) can be recorded as in [7] so that similar analysis can be easily performed even when the data is updated in the future.

Support Visual Annotation. Communicating and presenting visualizations were one of the most frequent tasks for all participants in their collaborative research. However, existing genomics visualization tools do not support even simple annotations, such as adding arrows, rectangles, and textual notes. Support of visual annotations can reduce context switching between tools, can help communicate visualization insights with colleagues, and can be helpful to increase the reproducibility of analysis when combined with the support of interaction provenance (e.g., adding a note to an outlier on a track).

Support Seamless Tool Transitions. Given that many transitions between tools and environments happen, visualization tools need to support seamless transitions between them. For example, support exporting SVG images in a visualization tool can help the use of existing graphic design tools. Also, supporting standard genomics files as much as possible can be helpful. For example, P6 created standard genomics files (BED) to specify the regions of interest which helped them to use the file in multiple genomics visualization tools (e.g., displaying regions of interest in both IGV [23] and Gviz [8] using the same file).

Consider Multi-Modal Interactions. Participants provided inconsistent opinions about the ease of use for GUI-based approaches. For example, several participants with programming skills found that direct manipulation is more cumbersome than using visualization libraries. This indicates that supporting interactions of multiple modalities can be beneficial to fulfill a wide range of users at once.

Support Computational Notebooks. Ideally, data processing and visualization can be performed in a single environment. However, participants did not find useful genome browsers in R environments, making them use a standalone tool for visualizing processed data. Support of companion libraries for R and Python environments, such as JBrowseR [9] and Gos [18], can simplify users' analysis and visualization workflows, making the process more reproducible and efficient.

Support Diverse Data Sources. The location that stores and processes data files varied largely in our interviews, from a local computer to public/private cloud services. Since there are often many large-size files that people have to analyze, it would not be ideal to transfer these files to different locations for visualization tools. Therefore, visualization tools should support flexibility in loading datasets from local and remote locations, such as the support of Google Cloud Storage in IGV [23] that several participants appreciated.

Lower Data Transformation Burden. Dealing with data (e.g., processing, analyzing, and transferring) was commonly a laborious task for participants. The lack of flexibility in transforming data, such as merging and filtering, for the purpose of using visualization tools increases the barrier to using visualization tools even more. In addition to supporting diverse standard genomics file formats, it will be valuable for users to support frequently used data transformation functions, as in Vega-Lite [27] and Tableau.

6 Limitation

Our interview results do not fully reflect the diversity of users and visualization tools in the real world due to the limited sample size. Most participants in our interviews had computational skills, but there are many biologists and clinicians who analyze similar data without any experience with programming languages. Moreover, while we recruited participants from multiple institutions using public channels, many of the participants in our interviews were working on a similar analysis, i.e., structural variation. While the findings of this study can be applied immediately to visualization design, studying a broader set of workflows might lead to additional findings or would allow us to refine the proposed design implications.

7 Conclusion

The understanding of genomics data analysis and visualization workflows and the identification of challenges in performing workflows can inform the creation of effective and efficient visualization authoring tools. Toward this goal, we conducted semi-structured interviews with nine genomics data analysts, including four graduate students, three postdocs, one senior scientist, and one software engineer. Through the interviews, we were able to identify nine different workflows for analyzing genomics data with visualizations. We discussed several unique characteristics of workflows, such as heavily repetitive and iterative tasks, as well as limitations of existing genomics tools for supporting various workflows, such as the lack of automation. Our interviews provide potentially useful features that can be considered when designing future genomics data visualization tools.

Acknowledgments

This work was supported by the National Institutes of Health (R01HG011773). We thank all interview participants who provided in-depth insights about their workflows and genomics tools.

References

[1] F. Amini, M. Brehmer, G. Bolduan, C. Elmer, and B. Wiederkehr. Evaluating data-driven stories and storytelling tools. In *Data-driven storytelling*, pp. 249–286. AK Peters/CRC Press, 2018.

[2] H. K. Bako, X. Liu, L. Battle, and Z. Liu. Understanding how designers find and use data visualization examples. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1048–1058, 2022.

[3] M. Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[4] F. Chelaru, L. Smith, N. Goldstein, and H. C. Bravo. Epiviz: interactive visual analytics for functional genomics data. *Nature methods*, 11(9):938–940, 2014.

[5] A. Crisan and B. Fiore-Gartland. Fits and starts: Enterprise use of automl and the role of humans in the loop. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

[6] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010.

[7] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From visual exploration to storytelling and back again. In *Computer Graphics Forum*, vol. 35, pp. 491–500. Wiley Online Library, 2016.

[8] F. Hahne and R. Ivanek. Visualizing genomic data using gviz and bioconductor. *Statistical genomics: methods and protocols*, pp. 335–351, 2016.

[9] E. A. Hershberg, G. Stevens, C. Diesh, P. Xie, T. De Jesus Martinez, R. Buels, L. Stein, and I. Holmes. Jbrowser: an r interface to the jbrowse 2 genome browser. *Bioinformatics*, 37(21):3914–3915, 2021.

[10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55

[11] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE transactions on visualization and computer graphics*, 18(12):2917–2926, 2012.

[12] W. J. Kent, A. S. Zweig, G. Barber, A. S. Hinrichs, and D. Karolchik. Bigwig and bigbed: enabling browsing of large distributed datasets. *Bioinformatics*, 26(17):2204–2207, 2010.

[13] P. Kerpedjiev, N. Abdennur, F. Lekschas, C. McCallum, K. Dinkla, H. Strobelt, J. M. Luber, S. B. Ouellette, A. Azhir, N. Kumar, et al. Higlass: web-based visual exploration and analysis of genome interaction maps. *Genome biology*, 19(1):1–12, 2018.

[14] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009.

[15] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[16] S. L'Yi and N. Gehlenborg. Multi-view design patterns and responsive visualization for genomics data. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):559–569, 2022.

[17] S. LYi, Q. Wang, F. Lekschas, and N. Gehlenborg. Gosling: A grammar-based toolkit for scalable and interactive genomics data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):140–150, 2021.

[18] T. Manz, S. L'Yi, and N. Gehlenborg. Gos: a declarative library for interactive genomics visualization in python. *Bioinformatics*, 39(1):btad050, 2023.

[19] G. G. Méndez, U. Hinrichs, and M. A. Nacenta. Bottom-up vs. top-down: Trade-offs in efficiency, understanding, freedom and creativity with infovis tools. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 841–852, 2017.

[20] M. Meyer, T. Munzner, and H. Pfister. Mizbee: a multiscale synteny browser. *IEEE transactions on visualization and computer graphics*, 15(6):897–904, 2009.

[21] S. Nusrat, T. Harbig, and N. Gehlenborg. Tasks, techniques, and tools for genomic data visualization. In *Computer Graphics Forum*, vol. 38, pp. 781–805. Wiley Online Library, 2019.

[22] P. J. Park. Chip–seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.

[23] J. T. Robinson, H. Thorvaldsdóttir, W. Winckler, M. Guttman, E. S. Lander, G. Getz, and J. P. Mesirov. Integrative genomics viewer. *Nature biotechnology*, 29(1):24–26, 2011.

[24] D. Sacha, M. Kraus, D. A. Keim, and M. Chen. Vis4ml: An ontology for visual analytics assisted machine learning. *IEEE transactions on visualization and computer graphics*, 25(1):385–395, 2018.

[25] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014.

[26] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. In *Computer Graphics Forum*, vol. 33, pp. 361–370. Wiley Online Library, 2014.

[27] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016.

[28] 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, 2012.

[29] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg. Avocado: Visualization of workflow–derived data provenance for reproducible biomedical research. In *Computer Graphics Forum*, vol. 35, pp. 481–490. Wiley Online Library, 2016.

[30] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–12, 2019.

[31] H. Wickham. ggplot2. *Wiley interdisciplinary reviews: computational statistics*, 3(2):180–185, 2011.

[32] L. Wilkinson. *The grammar of graphics*. Springer, 2012.

[33] T. Yin, D. Cook, and M. Lawrence. ggbio: an r package for extending the grammar of graphics for genomic data. *Genome biology*, 13(8):1–14, 2012.