



Thus, a dedicated dataset of genomic-specific NL2VIS data is essential to enable NLIs to support exploratory analysis.

To provide the foundation for generative NLIs in genomics, we present GQVis, the first comprehensive dataset of natural language to visualization data for genomic applications (Figure 1). Our **primary contribution is a dataset of over 2.2 million data points**, which consists of 1.14 million single-queries, 628k query pairs, and 589k multi-step (3+) query chains, each built using Gosling’s interactive visualization grammar [12]. Our **secondary contribution is extending the DQVis pipeline** for greater robustness in applied domain settings [8]. The original framework packages data schema, natural language queries, and visualization specifications. We extend this framework to support genomic data formats and incorporate three additional components: 1) justifications describing the choices made in visualization design, 2) academic figure captions, and 3) AltGosling [21] alternate text descriptions for visual accessibility. These additions increase the granularity and applicability of the dataset, providing rich semantic context for visualizations and supplementary information to enable LLMs to reason more effectively about visual design choices.

## 2 BACKGROUND & RELATED WORK

Over the past two decades, advances in high-throughput sequencing technologies have generated an unprecedented volume of genomic data [18]. Visualization offers a means for exploration, discovery, and communication within this data, transforming complex numerical representations into interpretable images. Researchers have developed numerous visualization systems to facilitate this analysis of genomic data. These include IGV [23], UCSC Genome Browser [5], Ensembl [4], and Comparative Genome Viewer [17], each designed to address specific visual and analytical requirements. More recent work that provides multiscale navigation and interaction is Gosling, a grammar-based visualization library for genome track data [12]. Unlike static visualization grammars, Gosling supports fully interactive, declarative specifications, enabling users to dynamically adjust scales, filter data, and compose multiple coordinated views in real time. This flexibility allows researchers to reproduce established plot types and iteratively explore new hypotheses by reconfiguring and interacting with visualizations.

Outside of genomics, grammar-based visualization systems have been primarily developed in the context of general-purpose data visualization. For example, Vega-Lite is a declarative grammar that specifies visualizations in a JSON format [20]. By abstracting visual design into composable building blocks, Vega-Lite enables reproducibility, flexibility, and diverse visualization types. These general-purpose systems do not support unique visualization methods and file formats commonly used in genomics. In general, grammar-based systems are particularly useful for constructing NL2VIS datasets because they enable structured and constructive specifications, creating alignment between queries and their corresponding visualization components.

Despite great potential in natural language-driven visualization generation, there are no existing datasets for NL2VIS that apply to genomic data. Rather, prior work has demonstrated the feasibility of creating large data visualization repositories for general-purpose and biomedical data. For example, Ko et al. introduced a dataset built on the Vega-Lite grammar [6], which contains a wide range of visualization specifications paired with natural language queries. This method scraped Vega-Lite specifications, then employed an LLM to produce possible queries that would result in the image. Similarly, Luo, Tang, and Yi put forth nvBench, which transformed existing data that mapped natural language queries to SQL queries and instead generated grammar-based visuals [9].

More recently, the DQVis framework [8] proposed a systematic approach for generating such datasets by varying data attributes and chart configurations. This pipeline creates triples of data, queries,

and visualizations, which can be adapted to any grammar-based language of visualization. Our work acts as a proof of concept for extending the DQVis generation pipeline by adapting its use to build a genomic dataset. This implementation demonstrates that grammar-based methods of visualization, when combined with DQVis queries, data schema, and specifications, can produce diverse and meaningful NL2VIS datasets.

## 3 DATASET GENERATION

The dataset generation process consists of five major components: template generation, template expansion, multi-step query curation, paraphrasing, and quality review.

### 3.1 Template Generation

The goal of this step is to capture a range of queries that could possibly be posed for a dataset. Nusrat et al. [15] details seven abstract tasks “covering the most important tasks for genomic visualizations.” We designed abstract queries spanning these seven tasks across single- and multi-locus objectives and single- and multi-feature sets.

In DQVis, abstract queries are written with entity (**E**) and field (**F**) placeholders, in which entities refer to donors, patients, or data tables, while features correspond to attributes of an entity. However, this cannot capture the full complexity in the structure of genomic data. Consequently, we expand this placeholder vocabulary to include:

1. Sample (**S**): A given sample or donor. This can contain metadata attributes, such as cancer type, cell type, and tissue type.
2. Entity (**E**): A data type found in a sample, such as point mutation data, RNA-seq reads, or Hi-C data.
3. Locus (**L**): A physical location of a gene or genetic marker on a chromosome.

These placeholders allow us to create generalized query forms that can later be expanded to a number of diverse outputs. For example, instead of writing a dataset-specific query, “What structural variants are present on chromosome 1?”, the template query would ask “What <E> are present on <L>?”. Here, <E> will take the place of an entity, while <L> will take the place of a location.

Queries may include information on metadata-level, as indicated with `S.metadata-identifier` syntax. To represent a cancer type or a cell type, a query would use `S.cancer-type` or `S.cell-type`, respectively. These queries are paired with a Gosling specification that includes the same placeholders. For example, where the specification asks for a URL, the template will state `<E.url>`, indicating that the place will later be filled with the URL from the corresponding entity.

For each query-visualization pair, we also create a justification and a caption to provide additional context. The justification describes why the visualization was selected for the query and can include descriptions supporting the use of a circular or linear layout, view alignment, choice of plot type, and more. The caption represents a figure caption for the image, and similar to the queries and visuals, contains S, E, and L placeholders to be expanded for caption specificity.

### 3.2 Template Expansion

The goal of template expansion is to fill the placeholder values in each query and specification with concrete sample, entity, and location names. However, not all abstract queries are logically meaningful. For example, we cannot ask about creating a point plot of structural variants. Therefore, each query-visualization pair will contain constraints that limit which samples, entities, and locations

may be applied to the query. If we are asking a question with a corresponding bar graph as the output visualization, we would add the constraint that the E must be able to be visualized as a bar graph. These constraints can also apply to relationships. If we are asking a question about S1 versus S2, we must ensure that S1 is not the same as S2. These constraints ensure a valid output query is created.

After constraints are defined, abstract queries will then be filled with real data. These data come from schemas that describe the sample-level, entity-level, and location-level data for the corresponding datasets. For our schemas, we drew from 4DN [3], ENCODE [22], and Chromosome [13] to represent genomic data across structural, functional, and epigenetic applications.

The reification of abstract queries with a dataset schema is formulated as a constraint satisfaction problem. This locates all sample-entity-locus combinations within the dataset that meet the required constraints and provides a list of solutions that map the abstract features to real data features. Any placeholder names and data references will then be replaced with the concrete data, resulting in a meaningful query and corresponding Gosling specification, which can be converted into an interactive visualization. We also create alternative text from this specification with AltGosling [21].

### 3.3 Multi-step Query Curation

Multi-step query chains are sets of two to eight queries that represent a synthetic analysis sequence. For example, the first query may ask to show the data at SMAD4, while the follow-up could ask to compare this plot to the data at BRCA1. These chains can help train conversational models to update figures in accordance with user requests.

We first took all major start queries (i.e., the output of single-query template generation) and created a list of possible follow-up queries for each. These possible follow-ups fell into one of five transition types:

- **Layout:** altering the visual layout of a plot
- **Comparative addition:** stacking a new data view to compare with the initial visual
- **Overlay:** overlaying two visualizations
- **Location zoom:** focusing the visual on a new location
- **Data stratification:** stratifying data by type, such as by type of structural variant

We tracked chains as tuples of the start query, the follow-up query, and the transition type. The chain adopts a structure related to a linked list, wherein each specification inherits its initial visualization from the previous query and links forward as the next tuple’s starting query until the end of the chain. This terminates after a randomly-selected length has been reached, chaining 2–8 queries.

After these multi-step chains are generated, we create concrete queries and visualizations for each query step. Based on the given transition type between two queries, we adjust the output specification to match the new view. For example, suppose we have the initial query “What is the <E> data?” and the follow-up “Display <E> at <L>.” This type of transition is a location zoom, as we are changing the initial data view from covering the whole genome to covering the area around <L>. As a result, the specification will be adjusted according to the handling of a location zoom, creating a new visualization built on the context of the prior query (Figure 2).

### 3.4 Paraphrasing

Expanding query templates results in many queries of the same template format. However, these expanded templates do not capture the full diversity and syntax of real user queries. Paraphrasing

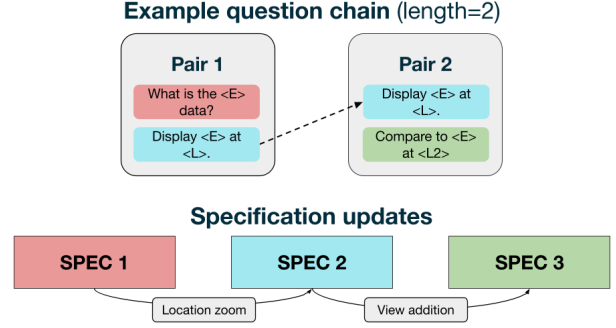


Figure 2: Multi-step generation pipeline. Chains are constructed from pairs of queries, in which the end query of a given pair corresponds to the start query of the next pair. The type of transition will determine how to update specs within the chain.

these concrete queries creates diversity in the query base, adding greater expressivity to the queries. Furthermore, integrating a range of query syntax enriches potential LLM learning from the dataset.

Based on the Ko et al. [7] framework, we employ GPT-4o to vary a query by *expertise* and *formality* on a score of 1–5, with higher scores expressing more technical and proper verbiage. Up to 25 paraphrased queries can be generated for each expanded template. Within the prompt template for the LLM, we also input relevant information about a query’s dataset schema, such as the entity and sample names, to enhance the LLM’s contextual understanding. Thus, a query phrased as “What is the frequency of structural variants at FBXW7?” is reworded as “What is the prevalence of structural variants (SVs) at the FBXW7 location?” and “How common are structural variants (SVs) around FBXW7?”, all corresponding to the same visualization.

### 3.5 Quality Review Software

The goal of reviewing is to ensure the quality and robustness of the generated dataset. Our review software, shown in (Figure 3), demonstrates queries and their corresponding Gosling visualizations with options for feedback. Should a given datum be below standards, researchers will have the option to elaborate on the issue and its significance. We plan to obtain opinions from domain-expert scientists across genomic research to assess the datasets to ensure alignment with researchers’ goals. Incorporating a review phase enables us to have a dataset that is both applicable to researchers’ needs and of high quality.

## 4 DATASET RESULTS

The initial resulting dataset consisted of 2.2 million data points describing genomic NL2VIS data. However, these data were strongly skewed to represent sample comparison queries, which covered over 80% of the dataset. To mitigate the bias for specific query types, we implemented data balancing measures to subsample from sample comparison and location comparison queries (Figure 4). Thus, the resulting single-query dataset consists of 1.14 million data points. Subsampling reduced the relative proportion of comparison queries, though they remain the dominant task types. The dataset is therefore not fully balanced, but the adjustment ensures improved representation of other query types without removing the natural distributional skew present in real-world tasks.

The dataset has extensive coverage and diversity of visualizations (Figure 5). We can view standard **structural** and **mutation** data through point, bar, and connectivity plots. Furthermore, epigenetic signals, such as **Hi-C**, **ATAC-seq**, and **ChIP-seq**, can be shown as a range of heatmaps, line plots, bar plots, and area plots

Dataset: CESC Sample 2

What are the different types of structural variant data?

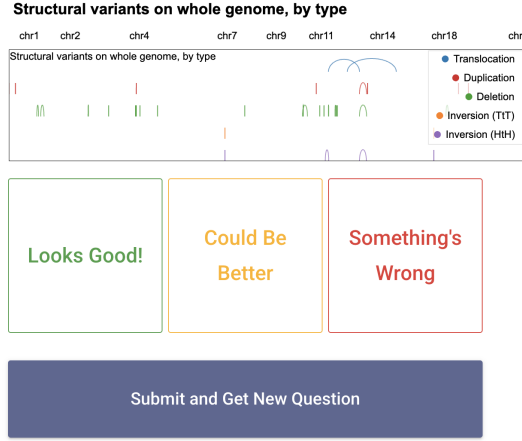


Figure 3: The review interface for assessing the quality of GQVis. Visuals and queries can be reviewed.

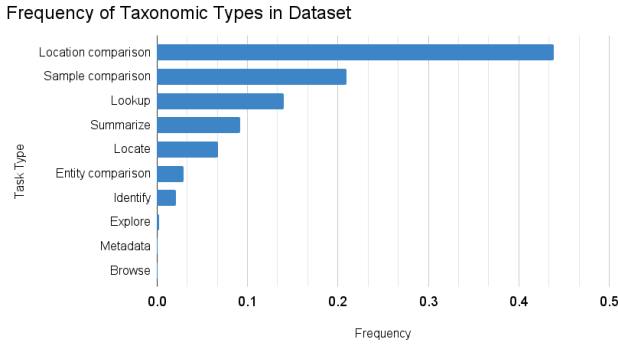


Figure 4: Relative frequency of query-visualization pairs by task taxonomy [15] after subsampling.

depending on the use case. These visualization types support comparison across entities (i.e., ChIP-seq versus ATAC-seq), samples (SV in sample 1 versus sample 2), and locations (Hi-C at chromosome 1 versus chromosome 2). This greatly increases the applicability of the data to investigative use. Moreover, any visualization can be paired with an ideogram for genome context information.

In addition to these single- and multi-view visualizations, we can also import complex visualizations. Chromosome is a collection of interactive multiscale visualizations for structural variation in human genomics. Each visual combines structural variants, indels, point mutations, a chromosome cytoband, and additional data to create an all-encompassing site for data exploration. These visualizations are directly ported into the GQVis dataset, enabling the creation of highly complex exploratory visualizations across scales.

## 5 CONCLUSION AND FUTURE WORK

This paper 1) generates a dataset of over 2.2 million data points linking natural language queries to data visualizations for genomic applications and 2) proposes a pipeline for generating a natural-language-to-visualization dataset that focuses specifically on genomic data visualizations. We identify two primary directions for future work.

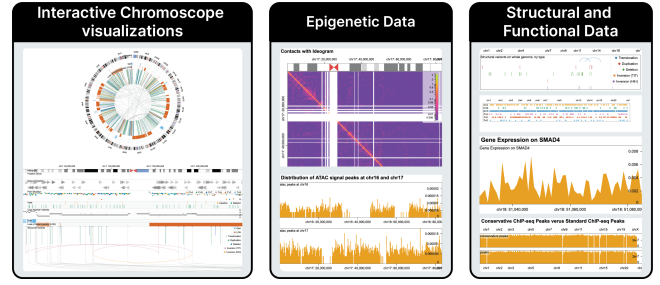


Figure 5: Diversity of the dataset. GQVis covers structural, functional, and epigenomic data, including Chromosome visuals.

First, we are actively developing a quality assessment framework for the generated dataset. Following the methodology established by DQVis, we are implementing a review interface that enables human evaluators to systematically assess the alignment between generated visualizations and corresponding natural language queries. Our evaluation approach encompasses both individual assessment at the query-visualization pair level and comprehensive robustness evaluation of the entire dataset, representing a work package that extends beyond the scope of this initial contribution.

Second, we plan to leverage this dataset for fine-tuning large language models specifically for natural-language-to-visualization (NL2VIS) tasks in the genomic domain. In particular, we aim to develop a domain-adapted model to generate accurate visualization specifications that respect genomic conventions. Beyond accuracy, we will examine the ability of these models to generalize to unseen query types, thereby establishing benchmarks for NL2VIS systems in genomics and informing the development of next-generation interactive analysis tools. Furthermore, the broad nature of our dataset beyond the standard query/visual system allows us to integrate the data into other tools, such as in visualization quality assessment, caption generation, or multimodal genomics visualization search engines [14]. An example application of the fine-tuned LLM is employing it in a visualization authoring tool for genomic data (e.g., Blace [11]). One of the biggest challenges that people in genomics confront when authoring visualizations is configuring coordinated multiple views [24], which is essential for visual exploration and analysis of genomic data [10]. With the fine-tuned LLM, visualization authoring tools can provide reusable templates of multi-view visualizations based on user prompts describing visualization needs, or recommend the coordinated interactions between pre-authored views.

In summary, this work establishes the first large-scale, genomic-specific NL2VIS dataset and demonstrates the feasibility of adapting grammar-based pipelines to complex genomic domains. By bridging natural language, visualization grammars, and genomic conventions, our approach provides both a resource and a methodology for advancing generative AI-based natural language interfaces. GQVis not only enables model training and evaluation, but also lays the foundation for more accessible, dynamic, and interpretable genomic analysis. As these systems mature, we anticipate that domain-adapted NLIs will lower barriers to exploratory visualization, accelerate hypothesis generation, and ultimately strengthen the integration of computational and biological research.

## ACKNOWLEDGMENTS

We thank the members of the HIDIVE lab for their feedback and guidance throughout the project. This work was enabled by the Dr. Susanne E. Churchill Summer Internship in Biomedical Informatics (SIBMI) and the Human BioMolecular Atlas Program (HuBMAP) Undergraduate Student Internship Program. This study was in part funded by NIH grants R01HG011773 and K99HG013348.



## REFERENCES

- [1] A. G. Bick, G. A. Metcalf, K. R. Mayo, L. Lichtenstein, S. Rura, R. J. Carroll, A. Musick, J. E. Linder, I. K. Jordan, S. D. Nagar, S. Sharma, R. Meller, M. Basford, E. Boerwinkle, M. S. Cicek, K. F. Doheny, E. E. Eichler, S. Gabriel, R. A. Gibbs, D. Glazer, P. A. Harris, G. P. Jarvik, A. Philippakis, H. L. Rehm, D. M. Roden, S. N. Thibodeau, S. Topper, A. L. Blegen, S. J. Wirkus, V. A. Wagner, J. G. Meyer, M. S. Cicek, D. M. Muzny, E. Venner, M. Z. Mawhinney, S. M. L. Griffith, E. Hsu, H. Ling, M. K. Adams, K. Walker, J. Hu, H. Dodapaneni, C. L. Kovar, M. Murugan, S. Dugan, Z. Khan, E. Boerwinkle, N. J. Lennon, C. Austin-Tse, E. Banks, M. Katzen, N. Gupta, E. Henricks, K. Larsson, S. McDonough, S. M. Harrison, C. Kachulis, M. S. Lebo, C. L. Neben, M. Steeves, A. Y. Zhou, J. D. Smith, C. D. Frazar, C. P. Davis, K. E. Patterson, M. M. Wheeler, S. McGee, C. M. Lockwood, B. H. Shirts, C. C. Pritchard, M. L. Murray, V. Vasta, D. Leistriz, M. A. Richardson, J. G. Buchan, A. Radhakrishnan, N. Krumm, B. W. Ehmen, S. Schwartz, M. M. T. Aster, K. Cibulskis, A. Haessly, R. Asch, A. Cremer, K. Degatano, A. Shergill, L. D. Gauthier, S. K. Lee, A. Hatcher, G. B. Grant, G. R. Brandt, M. Covarrubias, E. Banks, A. Able, A. E. Green, R. J. Carroll, J. Zhang, H. R. Condon, Y. Wang, M. K. Dillon, C. H. Albach, W. Baalawi, S. H. Choi, X. Wang, E. A. Rosenthal, A. H. Ramirez, S. Lim, S. Nambiar, B. Ozenberger, A. L. Wise, C. Lunt, G. S. Ginsburg, J. C. Denny, The All of Us Research Program Genomics Investigators, Manuscript Writing Group, All of Us Research Program Genomics Principal Investigators, M. Biobank, Genome Center: Baylor-Hopkins Clinical Genome Center, C. Genome Center: Broad, and Mass General Brigham Laboratory for Molecular Medicine, Genome Center: University of Washington, Data and Research Center, All of Us Research Demonstration Project Teams, and NIH All of Us Research Program Staff. Genomic data in the All of Us Research Program. *Nature*, 627(8003):340–346, Mar. 2024. Publisher: Nature Publishing Group. doi: 10.1038/s41586-023-06957-x 1
- [2] H. K. Brittain, R. Scott, and E. Thomas. The rise of the genome and personalised medicine. *Clinical Medicine*, 17(6):545–551, Dec. 2017. doi: 10.7861/clinmedicine.17-6-545 1
- [3] J. Dekker, A. S. Belmont, M. Guttman, V. O. Leshyk, J. T. Lis, S. Lomvardas, L. A. Mirny, C. C. O’shea, P. J. Park, B. Ren, et al. The 4d nucleome project. *Nature*, 549(7671):219–226, 2017. 3
- [4] P. W. Harrison, M. R. Amode, O. Austine-Orimoloye, A. G. Azov, M. Barba, I. Barnes, A. Becker, R. Bennett, A. Berry, J. Bhai, et al. Ensembl 2024. *Nucleic acids research*, 52(D1):D891–D899, 2024. 2
- [5] D. Karolchik, A. S. Hinrichs, and W. J. Kent. The UCSC Genome Browser. *Current protocols in bioinformatics / editorial board, Andreas D. Baxeianis ... [et al.]*, CHAPTER:Unit1.4, Dec. 2009. doi: 10.1002/0471250953.bi0104s28 2
- [6] H.-K. Ko, H. Jeon, G. Park, D. H. Kim, N. W. Kim, J. Kim, and J. Seo. A Vega-Lite Dataset and Natural Language Generation Pipeline with Large Language Models. 1, 2
- [7] H.-K. Ko, H. Jeon, G. Park, D. H. Kim, N. W. Kim, J. Kim, and J. Seo. Natural Language Dataset Generation Framework for Visualizations Powered by Large Language Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI ’24, pp. 1–22. Association for Computing Machinery, New York, NY, USA, May 2024. doi: 10.1145/3613904.3642943 3
- [8] D. Lange, P. Sui, S. Gao, M. Zitnik, and N. Gehlenborg. DQVis Dataset: Natural Language to Biomedical Visualization, May 2025. doi: 10.31219/osf.io/rqb7u\_v1 1, 2
- [9] Y. Luo, J. Tang, and G. Li. nvBench: A Large-Scale Synthesized Dataset for Cross-Domain Natural Language to Visualization Task, Dec. 2021. arXiv:2112.12926 [cs]. doi: 10.48550/arXiv.2112.12926 1, 2
- [10] 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. 4
- [11] S. L’Yi, A. van den Brandt, E. Adams, H. N. Nguyen, and N. Gehlenborg. Learnable and expressive visualization authoring through blended interfaces. *IEEE transactions on visualization and computer graphics*, 2024. 4
- [12] 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. 1, 2
- [13] S. L’Yi, D. Maziec, V. Stevens, T. Manz, A. Veit, M. Berselli, P. J. Park, D. Glodzik, and N. Gehlenborg. Chromoscope: interactive multiscale visualization for structural variation in human genomes. *Nature methods*, 20(12):1834–1835, Dec. 2023. doi: 10.1038/s41592-023-02056-x 3
- [14] H. N. Nguyen, S. L’Yi, T. C. Smits, S. Gao, M. Zitnik, and N. Gehlenborg. Multimodal retrieval of genomics data visualizations. 2025. doi: 10.31219/osf.io/zatw9\_v2 4
- [15] S. Nusrat, T. Harbig, and N. Gehlenborg. Tasks, Techniques, and Tools for Genomic Data Visualization. *Computer Graphics Forum*, 38(3):781–805, 2019. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13727>. doi: 10.1111/cgf.13727 2, 4
- [16] A. Pandey, S. L’Yi, Q. Wang, M. A. Borkin, and N. Gehlenborg. GenoREC: A Recommendation System for Interactive Genomics Data Visualization. *IEEE transactions on visualization and computer graphics*, 29(1):570–580, Jan. 2023. doi: 10.1109/TVCG.2022.3209407 1
- [17] S. H. Rangwala, D. V. Rudnev, V. V. Ananiev, D.-H. Oh, A. Asztalos, B. Benica, E. A. Borodin, N. Bouk, V. I. Evgeniev, V. K. Kodali, V. Lotov, E. Mozes, M. V. Omelchenko, S. Savkina, E. Sukharnikov, J. Virothaisakun, T. D. Murphy, K. D. Pruitt, and V. A. Schneider. The NCBI Comparative Genome Viewer (CGV) is an interactive visualization tool for the analysis of whole-genome eukaryotic alignments. *PLOS Biology*, 22(5):e3002405, May 2024. Publisher: Public Library of Science. doi: 10.1371/journal.pbio.3002405 2
- [18] B. J. Reon and A. Dutta. Biological Processes Discovered by High-Throughput Sequencing. *The American Journal of Pathology*, 186(4):722–732, Apr. 2016. doi: 10.1016/j.ajpath.2015.10.033 2
- [19] H. Sadasivan, A. Klauser, J. Hensch, Y. Turakhia, G. Singh, A. Zeni, S. Beecroft, S. Narayanasamy, J. Nivala, B. Robey, O. Mutlu, K. Denolf, and S. Sitaraman. The Genomic Computing Revolution: Defining the Next Decades of Accelerating Genomics. In *2024 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, Sept. 2024. ISSN: 2643-1971. doi: 10.1109/HPEC62836.2024.10938492 1
- [20] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. 2
- [21] T. C. Smits, S. L’Yi, A. P. Mar, and N. Gehlenborg. Altgosling: automatic generation of text descriptions for accessible genomics data visualization. *Bioinformatics*, 40(12):btac670, Nov. 2024. doi: 10.1093/bioinformatics/btac670 2, 3
- [22] The ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012. 3
- [23] H. Thorvaldsdóttir, J. T. Robinson, and J. P. Mesirov. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, 14(2):178–192, Mar. 2013. doi: 10.1093/bib/bbs017 2
- [24] A. van den Brandt, S. L’Yi, H. N. Nguyen, A. Vilanova, and N. Gehlenborg. Understanding visualization authoring techniques for genomics data in the context of personas and tasks. *IEEE transactions on visualization and computer graphics*, 2024. 4