GEViTRec: Data Reconnaissance Through Recommendation Using a Domain-Specific Visualization Prevalence Design Space

Anamaria Crisan, Shannah E. Fisher, Jennifer L. Gardy, and Tamara Munzner, *Senior Member, IEEE*

Abstract—Genomic Epidemiology (genEpi) is a branch of public health that uses many different data types including tabular, network, genomic, and geographic, to identify and contain outbreaks of deadly diseases. Due to the volume and variety of data, it is challenging for genEpi domain experts to conduct data reconnaissance; that is, have an overview of the data they have and make assessments toward its quality, completeness, and suitability. We present an algorithm for data reconnaissance through automatic visualization recommendation, GEViTRec. Our approach handles a broad variety of dataset types and automatically generates visually coherent combinations of charts, in contrast to existing systems that primarily focus on singleton visual encodings of tabular datasets. We automatically detect linkages across multiple input datasets by analyzing non-numeric attribute fields, creating a data source graph within which we analyze and rank paths. For each high-ranking path, we specify chart combinations with positional and color alignments between shared fields, using a gradual binding approach to transform initial partial specifications of singleton charts to complete specifications that are aligned and oriented consistently. A novel aspect of our approach is its combination of domain-agnostic elements with domain-specific information that is captured through a domain-specific visualization prevalence design space. Our implementation is applied to both synthetic data and real Ebola outbreak data. We compare GEViTRec's output to what previous visualization recommendation systems would generate, and to manually crafted visualizations used by practitioners. We conducted formative evaluations with ten genEpi experts to assess the relevance and interpretability of our results.

Code, Data, and Study Materials Availability: https://github.com/amcrisan/GEVitRec

Index Terms—Heterogeneous Data, Multiple Coordinated Views, Data Reconnaissance, Bioinformatics.

1 Introduction

Data reconnaissance is the process of exploring a group of datasets that are not yet understood by a specific person; we recently defined *data recon* and proposed an iterative four-stage process of acquire, view, assess, pursue as a conceptual framework to reason about it [1]. Although the visualization literature covers many systems designed for investigative exploration, where a specific existing dataset is transformed and analyzed in depth, the goal of very quickly understanding potential linkages between unfamiliar datasets remains difficult when using existing systems.

We posit that visualization recommendation systems have great promise for operationalizing the goal of data recon through a concrete algorithm that quickly and automatically computes reasonable visual encodings with minimal input from a user. A recommender system could speed up the *view* stage of the data recon process, in contrast to any kind of design or selection process for visual encoding that involves human judgement. The aim of data recon is to quickly assess whether the current collection of datasets is suitable for some intended analysis question, and if not to pursue additional data. After the recon process concludes when an appropriate collection of datasets has been acquired, a more lengthly analysis process with traditional investigative exploration visualization tools could occur.

Prior work takes a one-size-fits-all approach based solely on domain-agnostic perceptual effectiveness rankings, from the foundational APT system [2] to Tableau's ShowMe [3] to recent activity including Voyager [4] and Draco [5]. Such systems can help users rapidly understand their datasets by automatically suggesting suitable encodings based on data characteristics and the efficacy of different chart types. However, these systems primarily focus on encoding a single dataset of tabular data type. They fall short for two key requirements for the data recon use case: the ability to both find and show linkages between multiple datasets, and the ability to handle a broad variety of data types as input and a broad variety of chart types as output.

A novel feature of our approach is that we tailor the recommendation for a specific domain, to tame the combinatorial explosion of possibilities that arise from a broad variety of input data types and output chart types. We leverage the knowledge encapsulated in a recent *domain-specific visualization prevalence design space (VPDS)*, where visual design collections used by experts in a particular domain are both characterized and enumerated [6]. We use this domain-specific information in a few targeted stages of our recommender algorithm, in conjunction with many domain-agnostic decision procedures. The GEViT [6] VPDS

- *A. Crisan is with Tableau Research. Email: acrisan@tableau.com.*
- *S. Fisher and T. Munzner are with the University of British Columbia. Email: shannahelizabeth@gmail.com, tmm@cs.ubc.ca.*
- *J. Gardy is with the Gates Foundation. Email: jennifer.gardy@gatesfoundation.org*

Manuscript received September 15, 2020; revised 14 July 2021.

that we proposed in previous work arises from *genomic epidemiology (genEpi)*, a very appropriate exemplar domain for data recon with a diverse and heterogeneous set of input data types and output chart types; genEpi analysts often grapple with unfamiliar collections of datasets in situations where access is tightly controlled due to sensitive personally identifiable information.

The main contribution of our work is the design and implementation of `GEViTRec`, an end-to-end recommender algorithm to support the data recon process that computes potential linkages between datasets and automatically generates visually coherent combinations of charts to illustrate them. The visual coherence between charts is achieved through an automatic coordination of positional and color encoding channels of the individual charts. The resulting specification is rendered into a single static image of the visually coherent combination that allows the linkages between data sources to be immediately perceived, without requiring any kind of time-consuming interactive exploration [7]. Many stages of our proposed recommender algorithm feature domain-agnostic computations while some stages leverage domain-specific prevalence information. We validate our approach by comparing our results to existing genEpi visualization dashboards created by careful human curation. We also conducted an evaluative interview study with ten genEpi experts to verify the interpretability and utility of `GEViTRec`'s results.

2 Domain-Specific Prevalence for Genomic Epidemiology

Visualization has historically been a prominent component of epidemiological research and practice since John Snow's famous 1854 cholera map. More recently there has been an influx of new, heterogeneous, and multidimensional sources of data, including whole genome sequencing data and pulsed-field gel electrophoresis for DNA fingerprinting, that has birthed a new field: genomic epidemiology (genEpi) [8]. Genomic data can be difficult to integrate with other data sources, including tabular data from electronic health records, network data from contact tracing, and spatial data [9], [10]. The volume and variety of data at play in the genEpi domain make it an excellent exemplar for the utility of a domain-specific visualization prevalence design space in a proof-of-concept system. Moreover, there is a clear need for data recon in genEpi analysis contexts. The sensitivity of clinical health data leads to stringent access controls, so multiple rounds of acquire, view, assess, and pursue are often required to gather the requisite collection of datasets to answer any specific analysis question. Also, epidemiological analysis frequently takes place in an urgent context, particularly in pandemic times.

2.1 Genomic Epidemiology Visualization Typology

Our prior study characterized and enumerated a domain-specific data visualization collection assembled in such a way so as to be reflective of visualization designs currently used by genEpi experts [6]. It proposed a Genomic Epidemiology Visualization Typology (GEViT) that broke down how visualizations were constructed through chart types, enhancements, and combinations. The typology encompasses 25 unique chart types grouped within 8 categories (common statistical charts, color, relational, temporal, spatial, tree, genomic, and other), four types of chart combinations (positionally aligned, color aligned, small multiples, and unaligned), and two primary mechanisms of enhancements (adding or re-encoding marks). GEViT was developed using a corpus of approximately 18,000 research articles pertaining to genomic epidemiology that was representatively sampled to yield a set of 800 figures that informed the typology generation. Text mining techniques were applied to the titles and abstracts of all articles to derive a notion of the creation context for the sampled set of figures, so that connections between different data types and domain-specific attributes to data visualizations in the form of individual charts and chart combinations could be harvested. Most importantly, the representative sampling strategy also enabled the enumeration of these different visual design collections, providing a quantitative measure of their relevance and importance to genEpi. The GEViT collection also captures the year of use, providing longitudinal data about how and whether approaches to data visualization design might change over time.

This information exactly constitutes a domain-specific visualization prevalence design space (VPDS), although that specific term is introduced here and was not used that work.

2.2 Visualization Prevalence Design Spaces

The fundamental idea behind our approach is to inject domain relevance into a visualization recommendation algorithm based on an analysis of the visualization design space in use in that domain. The term *design space* is used broadly within the visualization research literature with many shades of meaning, and the GEViT approach that we leverage is one of many possible strategies for generating a visualization design space. The collection of visualizations assembled in the GEViT study was categorized along three cross-cutting axes to delineate a design space: chart types, chart combinations, and encoding enhancements.

Our approach has a similar spirit to the pioneering work by Card and Mackinlay to capture the structure of a visualization design space [11] by breaking down visual encoding choices along cross-cutting axes. A difference is that we generated a typology, whereas they defined visual vocabulary. Another interesting divergence is that Card and Mackinlay took a deductive approach, defining a vocabulary first and then using it to characterize a visualization design space; through GEViT we took an inductive approach starting with the analysis and characterization of existing visualizations and using that to construct our descriptive typology, and by extension the design space. Other examples of design spaces include SetVis [12] and TreeVis [13], focused on specific data types.

One very valuable property of the GEViT approach is that it captures visualizations made from many data types and with many different systems, crucially including manual post-processing adjustments made by users with image creation and processing tools. Design spaces derived from analyzing visualizations made with only a single system, such as Tableau Public [14] or ManyEyes [15], would not

capture such a broad diversity of visualization designs and their prevalence. Also, in contrast to these other design space examples, the construction of a design space via the sampling approach of GEViT allows us actually derive computable metrics from prevalence data, which we can use as inputs to visualization recommendation.

To differentiate our approach from past work, we define a *domain-specific design space* as the result of analyzing a collection of *visual encodings* that are produced by experts in some domain. We define a *domain-specific visualization prevalence design space*, or just **(VPDS)**, as one that both captures the full scope (within reason) of visual encodings used by some definable set of experts and includes a quantitative estimate for prevalence of different visual encoding strategies within the domain. We use this quantitative prevalence information in our algorithm to derive relevance scores for different visual encodings. We leverage the genEpi VPDS from the GEViT study in our proposed recommendation algorithm; the extension of our prior design space work to recommendation systems is a novel contribution of our present work, and it can transfer to domains beyond genEpi.

The specific implementation of the GEViT design space draws visualizations from research articles, which are an instance of expert-to-expert presentation. We posit that such a design space is nevertheless suitable for generating visualization recommendations. First, we can frame a recommendation algorithm as making choices of how to present data to experts to facilitate data recon. Second, even if data recon is considered closer to exploration than presentation, the overlap between design spaces for presentation and exploration is very substantial [16].

2.3 The Importance of Visual Cohesiveness

The quantification from the GEViT VPDS allowed us to observe the high prevalence of using shared position and color channels to encode common information. The design space axis of chart combinations categorizes these approaches as positional and color chart combinations (see Section 2.1). Instituting a visual cohesiveness across multiple chart types through the process of coordinating these encoding channels was a common practice, and we hypothesize it required significant manual labour.

The importance of ensuring visual coherence between multiple static charts, a scenario that represents a large portion of real-world use of data visualization, has been documented by multiple authors [6], [17], [18], [19]. However, the use and construction of static views has been understudied by the research community [20]. Instead, coordination has come to connote interactive techniques that link across chart types, which excludes other important methods for showing relationships between charts, such as layouts with shared positional axes. Although there are many ways that interaction can complement data visualizations [21], interactive approaches do have notable limitations such as technical and perceptual costs [7], [22]. For example, Battle et al. [23] found that specific types of interactions involved in reasoning about data can have high costs. Moreover, even when present, interaction capabilities are not always discovered or used [24]. While these limitations do not negate the benefits of interaction, they do suggest we take the opportunity to consider alternative and complementary modes for coordinating information between charts that do not rely primarily on interaction. Here, we use the terminology of *visually coherent chart combinations* to reflect the visual coordination of shared information between charts with respect to layout and consistency among visual channels, and to differentiate from the term *coordinated views* that may imply the use of interaction techniques.

Despite the heavy focus of the visualization research community on interaction as a means for coordinating shared information between multiple charts, prior research does suggest many benefits from coordinated layouts. For example, juxtaposed or superimposed charts have been deemed useful for comparison [17]. Kehre et al. [25] similarly demonstrated the importance of coordinated visual mapping accomplished via shared axes or color palettes on a single chart, a process they called *data fusion*, alongside interaction techniques for heterogeneous data. L'Yi et al. [18] also found that *visually linking* charts via juxtaposed layouts helped people orient themselves in a visualization even when systems had affordances for *interactive* linking of views. This prior work echoes findings by Qu et. al [19] who also demonstrated the importance of coordinating visual information between charts to keep multiple views consistent. While this research emphasizes the importance of visual cohesiveness, it does not address the challenges of creating visually coherent chart combinations across multiple data sources. This challenge is especially salient in genEpi, where static means of reporting findings, either through publications or through electronic health records [10] preclude reaping the benefits of interaction. Here, we take up this challenge by examining how to create visually coherent chart combinations, through leveraging the VPDS. We note that visual coherence and interaction are complementary approaches rather than a mutually exclusive dichotomy. We focus on the former, to advance the state of art for the automatic support of visual coherence through non-interactive chart-wise coordination; despite evidence of its potential utility, the problem of providing automatic support for it has been largely overlooked in previous work.

3 Related Work

We survey related work on visualisation recommendation, on integrating and visualizing heterogenous data, and on chart coordination and combinations.

3.1 Visualization Recommendation

Automated recommendations of visual encodings are a way to help users explore datasets, and have been pursued by both the research and practitioner communities. A number of visualization recommender systems use rule based approaches that generate recommendations with the combination of a user's specifications and the types of data attributes (i.e. numeric, categorical, and nominal) [2]. The ShowMe [3] and Voyager [4], [26] systems also rank visual encodings according to manually predetermined scores of graphical perception efficacy; highest ranked visualizations are prioritized and shown to the user, while lower ranked visualizations are accessible through interactions with the

data visualization systems. Draco takes this ranking approach further by learning the efficacy scores automatically from the results of graphical perception experiments [5].

Beyond perceptual ranking of visualizations, other tools have also explored leveraging the properties of the data to recommend visualizations. Zenvisage [27] allows users to quickly generate and curate a set of visualizations, and proposes the ZQL language to algebraically express visual patterns in hopes of making it easier to explore data. The ZQL language is complementary to the VizQL and COMPASS-QL languages the underlie the specification, used in the ShowMe and Voyager/Draco systems respectively.

SeeDB [28] proposes a mechanism for surfacing high-utility visualizations based upon user queries and the selection of a set of views with high variances; that is, they use a measure of deviation from a baseline to return views that represent a diverse view of the data. Our research is similarity interested in capturing a diversity of views, but unlike SeeDB we anchor our recommendations on diversity of data types as opposed to singleton view distributions. The VizDeck [29] system also leverages statistical properties of the data to recommend visualizations. Further, VizDeck uses the card metaphor to allow users to manually construct multiple views of the data. Foresight [30] treats visualization recommendation as a optimization over an insight search space; it builds upon earlier Rank-by-Feature [31] and AutoVis [32] approaches, which propose mechanisms for users to steer the ranking of automatically generated visualizations.

More recent systems, such as Data2Vis [33] and VizML [34], also attempt to learn the associations between different types of data and visual encodings. Both systems analyze datasets themselves, not just attribute types, and discover the kinds of associations that users generate "in the wild" from publicly available resources. These learned associations are then used to automatically generate data visualizations. Another class of prior systems attempt to develop a semantic understanding of data and use this knowledge in the recommendation processes, including SemViz [35] and Cammarano's schema matching technique [36]. Such techniques have not been expanded upon, likely due to the difficulties of developing and maintaining ontologies. A final class of systems attempts to infer user preferences via collaborative filtering [37].

These various approaches are not mutually exclusive and can be combined. While we take inspiration from these systems, all of them are limited to recommending visualizations for essentially a single source of tabular data. Our technique extends beyond their capabilities in supporting a wider variety of data types and visual encodings, and in supporting the automatic creation of visually coherent combinations of visual encodings.

3.2 Integrating and Visualizing Heterogeneous Data

Heterogeneous data sources can be challenging for analysts to work with because they mix different data types such as trees, tabular, and spatial data. Graph data structures have emerged as a primary means of integrating and visualizing these diverse data. Initial work by Cammarano et al. [36] demonstrated the utility of these techniques for semantic web data. More recent work by Kairam et al. [38] and Xie et al. [39] demonstrates the utility of this approach for other data types. The Refinery system proposed by Kairam et al. [38] goes further to demonstrate how graph structures can be used to support associative browsing that can help a user drill down to a data point of interest quickly. However, these approaches supporting heterogeneous data have not been applied to biomedical data, where systems are constrained to pre-existing views of the data. For example, Angelelli et al. [40] demonstrate how users can interactively explore heterogeneous data from patient records along with 3D spatial MRI images and Lineage [41] integrates both tree and tabular data. The primary limitation of such tools is only allowing users to integrate a limited set of data sources. The Domino system [42] straddles the bespoke visualization requirements of biomedical data with some of the graph-based integration requirements. From a single source dataset, Domino allows analysts to generate linked views of multiple subsets by allowing the user to generate 'subset blocks' and then linking these blocks according to the strength of their associations.

There are also many options for integrating and visualizing genomic epidemiology data specifically. GenEpi stakeholders can currently either develop their own solo or combined data visualizations using one of several charting libraries, or can use previously curated visualizations in publicly deployed dashboards and other interactive systems. We identified the charting libraries in common use: TreeViewer (stand-alone, [43]), Baltic (python, [44]), ape (R, [45]), ggplot (R, [46]), and ggtree (R, [47]). Since manually generating data visualizations is a time consuming and potentially wasteful activity in the midst of serious epidemiological crises, people have also developed interactive systems and dashboards with manually pre-curated sets of visualizations that are updated as outbreaks evolve. We have identified Nextstrain [48] and Microreact [49] as two such widely used and state of the art systems, and compare the results of `GEViTRec` to them.

While all of these tools support the integration and visualization of heterogeneous data, these systems still place a major burden on the analyst to provide systems with specifications for generating visualizations. We seek to lower this burden by integrating and visualizing heterogeneous data sources with minimal specifications from users.

3.3 Chart Coordination and Combination

Coordinating shared information amongst different charts is essential for visualization of many data types including biomedical data [25], [50]. Coordinated visual encodings are beneficial for highlighting shared attributes across a collection of datasets. Substantial previous work is devoted to interactive methods for linking views through shared information, particularly for juxtaposed views with linked highlighting [16], [51]. Visualization authoring systems including Improvise [52], Lyra [53], Data Illustrator [54], and Charticulator [55] and visualization libraries like D3 [56], ggplot [46], and Vega [57] (which underpins both Draco and Voyager) are able to support visually coherent combinations of charts, including small multiples, aligned color palettes, and unaligned combinations. It is possible to generate and manually coordinate single charts with commercial applications such as Tableau, PowerBI, or Excel.

Despite the flexibility and popularity of manual coordination of static charts for visual coherence, and prior work on grammars for facilitating visually coherent combinations such as HiVE [58] and ATOM [59], constructing effective combinations automatically remains a challenging proposition in practice [19]. No prior visualization recommendation systems attempts to do so; `GEViTRec` treats visually coherent combinations as first class citizens with full support.

To facilitate a consistent interface for combining visually coherent charts, our recommendation algorithm includes a specification syntax inspired by the GEViT [6] typology.

4 Recommendation Algorithm

Algorithm 1 GEViTRec (H, D, A, T, F_{users})

Input: Datasets H, Design Space D, Alignable Positional Combinations (A), Chart Templates T, User Specified Fields F_{users}

1: (F,M) ← `explodeFields`(H)
2: G ← `generateDataSourceGraph`(F)
3: R_p ← `rankPaths`(G, D)
4: V ← `generateVisSpecs`($R_p, M, F, F_{users}, A, T$)
5: `layoutAndRender`(V)

Our `GEViTRec` recommendation algorithm, summarized in Figure 1 and Algorithm 1, has six stages. In the first stage, the algorithm extracts attribute fields from input datasets, a procedure we refer to as *exploding fields*. In the second stage, the algorithm analyzes all of the exploded fields to detect which ones are shared between datasets, constituting links between them. The algorithm uses these detected shared fields to generate a data source graph encapsulating all connections between the input data. In the third stage, the algorithm analyzes and ranks paths within this data source graph according to their potential relevance to domain experts. The relevance criteria that we propose include broad coverage of input datasets and of different data types, and information from the domain-specific visualization prevalence design space to prioritize visual encodings commonly used in the domain. The fourth stage of the algorithm automatically generates initial partial programmatic specifications for singleton charts. Viable combinations of charts are generated in the fifth stage by modifying the specifications so that spatial positions or color palettes are aligned to link information according to shared fields. In the sixth stage, charts are arranged into a grid layout consistent with the computed alignments, and their completed specifications are rendered into boxes of pixels.

4.1 Input Data

Our proposed `GEViTRec` recommender algorithm has four required inputs (H, D, T, and A) and an optional fifth one (F). Two are provided by users. The required input from users is the (frequently) heterogeneous collection of datasets they wish to visualize together (H). Our proof of concept implementation supports the following data types for H: tabular, tree, genomic, images, spatial (polygons), or network data. Our recommender algorithm will still function if the user provides only a single dataset, or if the

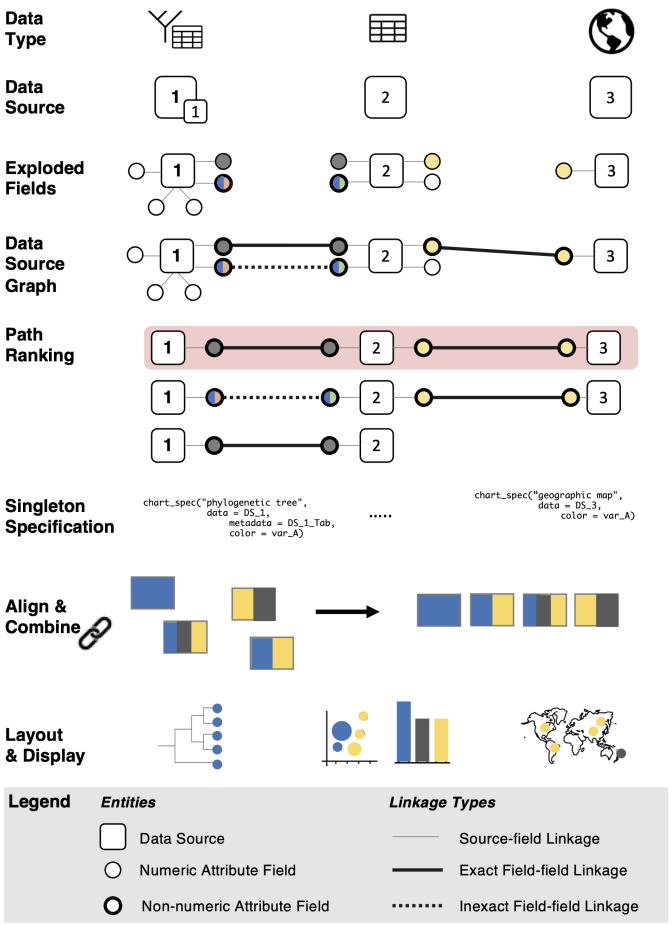

Fig. 1. **`GEViTRec` Overview.** The algorithm is illustrated with sources of three data types: #1 tree with associated tabular data, #2 tabular, and #3 spatial. The exploded attribute fields within these data sources are classified into numeric and non-numeric field types. The similarity of categories between pairs of non-numeric attribute fields is computed with the Jaccard index to establish exact and inexact linkages between data sources. The data sources, their attribute fields, and the linkages between their fields are used to generate a data source graph. The paths of the data source graph that link pairs of data sources are enumerated and ranked according to their link strength, diversity, and total relevance. For each path, in order of rank, partial specifications are generated for individual charts. They are modified to express linkages through aligned positional axes or color palettes, then arranged into a grid layout and the specifications are rendered into displayable pixels.

datasets they wish to visualize are independent (i.e. have no shared fields). Users have the option to specify a set of fields (F_{users}) that should appear in the final visual encodings, in addition to the automatically detected shared fields that provide linkages between the input datasets.

The other three inputs are not supplied by the user; they are precomputed datasets that we have built into our proof-of-concept implementation. One is a VPDS (D) that is a quantified and aggregated summary of different visual encodings and how frequently they are used, as described in Section 2.2. Another is a set of predefined template specifications (T) for all of the visual encoding types that are built into our proof-of-concept implementation, as discussed in Section 4.5, allowing `GEViTRec` to automatically generate visualizations without any further guidance from users. The third is manual analysis results A, namely the viability matrix for feasible spatial alignments of charts and the list of positionally immutable charts, described in Section 4.6.

4.2 Exploding Fields from Data Sources

For each input dataset in H, all of its fields are *exploded*; that is, they are extracted for easy comparison regardless of the original data type of the source. For each input data set, H_i, its fields are denoted $F = (f_1, f_2, ..., f_n)$. For each field in F, `GEViTRec` also determines whether it is numeric or non-numeric, and the cardinality of the non-numeric fields.

The automatic extraction of fields requires analysis of how attribute fields are stored in each supported data type. For tabular data types, these fields are simply the attribute columns. For other data types, these fields can be stored in different ways: for example, fields in tree data types may be stored within a plain text flat file that also describes the tree's structure. The attribute fields of non-tabular data types could be identifiers that link to an associated tabular dataset, or may contain important data as a complex concatenated string of data that needs to be parsed.

4.3 Generating the Data Source Graph

Our algorithm conducts a pairwise analysis of all non-numeric exploded fields between data sources to identify potential linkages between data sources. It computes the set similarity for all pairwise comparisons of non-numeric fields between data sources using the Jaccard Index. For a pair of attribute fields A and B, the Jaccard index is $J(A,B) = A \bigcap B / A \bigcup B$, where $J(A,B) \in [0,1]$. When $J(A,B) = 1$ there is an an exact match between all unique categories in the two fields, $J(A,B) = 0$ indicates no matches, and $0 < J(A,B) < 1$ is an inexact match indicating that some but not all values are in common.

The Jaccard index is used to derive a data source graph G of data sources and their associated attribute fields, shown in Figure 1. `GEViTRec` visualizes the resulting data source graph using a hub and spoke model (Figure 1, Figure 4). Rectangular hubs correspond to a data source (□) and circular spokes correspond to attribute fields (○). The strength of the connection between data sources, via field-field linkages, is shown using thick solid lines for exact matches and dashed lines for inexact matches. Thin solid lines denote linkages between sources and fields.

We limit the linkage check to non-numeric fields because numeric fields lack context to be robustly joined using this set similarity approach: they tend to have so many unique values that matching them to other fields is not straightforward. Extending `GEViTRec` to handle linkages between numeric fields robustly would be possible and interesting future work, but is beyond the scope of this initial research.

One immediately obvious limitation of this approach to generating the data source graph is that messy or noisy data may require users to undertake a clean-up process. Consider two datasets with a field for viral sample IDs that would be expected to have an exact match. If there are typographic errors or missed entries, the Jaccard Index would be sensitive to these issues: rather than an exact match, the linkage would be computed with a lower value (< 1). However, this situation still provides a useful insight to surface to the expert. The hub and spoke data source graph visualization allows experts to assess the strength of connections and do a targeted investigation into expected connections that failed to materialize. Moreover, since the Jaccard Index is a continuous value, experts can quickly triage their investigations into data quality by, for example, prioritizing those that have lower Jaccard Index values. Even with messy data, our recommendation algorithm will still produce and prioritize visualizations; performance will simply degrade in that fewer visually coherent combinations of charts will be generated.

4.4 Ranking Graph Paths

After generating a data source graph, the next stage of the algorithm generates and ranks paths between paired groups of data sources. Since data sources may not all be connected to each other, `GEViTRec` generates paths within individual connected components in the graph. A single connected component of the data source graph could contain anywhere between a single dataset and all input data sources, so the number of paths per component can vary from 1 to beyond $\binom{N_H}{2}$, where N_H is the total number of data sources.

Algorithm 2 rankPaths(G,D)

Input: Data Source Graph G, Design Space D
Output: List of Ranked Paths, R_p

```
rankedPaths R_p ← new Array
for each component C in G do
    C_p = enumeratePaths(G_c)   ▷ Dijkstra's algorithm
    for each path P in C_p do
        P_s ← calculatePathStrength(P)
        P_d ← calculatePathDiversity(P)
        P_vr ← calculatePathVisRelevance(P)
        add P_s, P_d, P_vr as a row to R_p
R_p ← columnNormalize(R_p)
pathScore ← score(R_p)
sort R_p by pathScore
return sorted R_p
```

We rank paths in each component of the data source graph with a **relevance rank** that incorporates three metrics that we developed: the strength of the connections between data sources (**link strength**), the diversity of data types (**data type diversity**), and the cumulative relevance of visual encodings that could be generated from those data types (**visual encoding relevance**). The visual encoding relevance metric is how we incorporate domain-specific visualization practices into the path ranking criteria, specifically the genEpi visualization prevalence design space available from the GEViT study [6].

We summarize our approach to path ranking in Algorithm 2. The **link strength (P_s)** of a path is the normalized sum of edge weights (e_i):

$$P_s = \frac{\sum_{i=1}^{N_e} e_i}{N_e} \tag{1}$$

where N_e is the total number of edges on that path and e_i is the Jaccard index of the i^{th} edge, and finally $P_s \in [0,1]$. A value of $P_s = 1$ indicates that data sources are linked entirely by exact matches, whereas $P_s = 0$ indicates that there is only one data set in that path.

Data type diversity (P_d) is calculated by summing the number of unique data types in a path. Each data source is classified as belonging to one of the supported data types;

in the implementation, this calculation is done at runtime when the user loads the input data into GEViTRec. A diversity value of 1 indicates that every data source appearing on the path is the same data type, for example tabular data. The maximum possible diversity value of N_H, the total number of input sources, would indicate that each data source is a different data type. The objective of the diversity metric is to include a number of different data sources, but also to prioritize visualizing data across a variety of data types.

Finally, we compute the **visual encoding relevance (P_{vr})** of a path by summing the relevance scores of the unique visual encodings that can be produced from the different data sources along the path. We have pre-computed a scaled visual encoding relevance (R') using the results of our prior analysis of visualization practices in genomic epidemiology [6]. The R' value is derived from a quantification for visual encoding (V) usage within a domain-specific visualization prevalence design space ($VPDS$), where $R' \in [1, 10]$. We compute R as follows:

$$R = \sum_{y}^{Y} \sum_{n=1}^{N_y} V_{i,n} \times w_y \tag{2}$$

where V_i is an individual visual encoding in the form of a chart (i.e. scatter chart, map, phylogenetic tree). A simple relevance metric score would just quantify how often some visual encoding (V_i) is used in some design space ($VPDS$); for the prior GEViT study [6], this number captures how many papers a chart type appears in.

We sought to give greater weight to more recent visual encodings, reasoning that best practices were likely to evolve over time. Thus, we create a penalty term (w_y) that down-weights counts from older years, a possible computation since the GEViT VPDS includes information about the year of use Y. This approach sums the occurrence of individual visual encodings, while giving greater weight to visual encodings that domain experts used most recently.

For consistency across evolving design spaces, we rescale the visual encoding relevance (R) as follows:

$$R' = \left(\frac{V_i}{max(R)} \right) \times 10 \tag{3}$$

The above equation identifies the visual encoding (V_i) that is used most frequently in this design space (R). All the visual encodings in the design space are scaled as a fraction of R. We include a scaling factor of 10 for interpretability and ease of computing; other scaling factors could have been used. The scaled relevance score ranges between 1 (least common) and 10 (most common). It is designed to produce non-linear quantitative values to emphasize the relative importance of different types of visual encodings, not simply an ordering of the elements.

For example, a phylogenetic tree is the most used visual encoding in genEpi and the next most common visual encoding is a bar chart. Our scaling produces scores of 10 and 4 respectively for these charts, rather than scores like 10 and 9, to capture the relative importance of a phylogenetic tree compared to all other encodings.

A) Scatter Chart Template

```
chart_spec(chart_type = "scatter",
           data = data_obj(NA,"table"),
           x=var_obj(NA,"quant|qual",dataSource=NA,TRUE),
           y=var_obj(NA,"quant|qual",dataSource=NA,TRUE),
           color = var_obj(NA,"qual-12",dataSource=NA,FALSE),
           shape = var_obj(NA,"qual-6",dataSource=NA,FALSE))
```

B) Phylogenetic Tree Chart Template

```
chart_spec(chart_type = "phylogenetic tree",
           data = data_obj(NA,"phyloTree"),
           metadata = data_obj(NA,"table"),
           color = var_obj(NA,"qual-12",dataSource=NA,FALSE),
           shape = var_obj(NA,"qual-6",dataSource=NA,FALSE))
```

Fig. 2. Chart Templates. These initial partial specifications for chart templates are pre-defined and internal to GEViTRec, not exposed to the user. Examples: (a) Scatter chart. (b) Phylogenetic tree.

We calculate the total relevance of a path by calculating P_{vr} as follows:

$$P_{vr} = \sum_{1}^{I_p} max(R') \tag{4}$$

where I_p is the total number of datasets in the path and $1 \leq I_p \leq N_H$. In some instances, a data type may map to multiple different possible encodings, for example with tabular data that can map to various statistical chart types (bar chart, scatter chart, etc.). In this case our algorithm only considers the highest value encoding ($max(R')$) that a dataset maps to. Other data types only link to one visual encoding, for example, spatial polygons can only be drawn as a map; in this case, there is only a single value of R'. Consequently, $P_{vr} \in [1, max(R'_{VPDS}) \times I_p]$, where R_{VPDS} is the visual encoding with the highest value in the entire design space. As a concrete example, a phylogenetic tree is the most important visual encoding in our existing design space and would have $R' = 10$. If the user supplies five input datasets, each of which are phylogenetic trees and are all connected to each other via a common sample ID, then the value of $P_{vr} = R' \times 5 = 10 \times 5 = 50$. The P_{vr} metric enables our technique to prioritize data types likely to produce visual encodings users may care most about, from the domain-specific VPDS information.

As a final step, we rank normalize the values of P_s,P_d,P_{vr}, such that P_s, P_d, $P_{vr} \in [1, P]$, where 1 indicates the highest ranked and P, the number of paths, is the lowest ranked value. The final importance score for a path is established by summing the normalized values of path strength, diversity, and encoding relevance: $pathScore = P_s + P_d + P_{vr}$, where $pathScore \in [3, 3P]$.

4.5 Generating Singleton Specifications

Beginning with the highest ranked path ($pathScore = 3$) GEViTRec generates programmatic specifications for coherent and combined visual encodings in two passes, as described in Algorithm 3. The first pass generates a specification for each singleton charts using a pre-defined set of chart templates, one for each supported chart type. At run time, chart templates are converted to declarative specifications that facilitate layout and rendering of charts.

It would have been possible to serve the same purpose with a bespoke query language such as CompassQL, as used

Algorithm 3 generateVisSpecs($R_p, M, F', F_{user}, A, T$)

Input: rankedPaths R_p, field metadata M, Fields F, User Specified Fields F_{users}, Encoding Templates T
Output: array of chart combinations V_C

$V_C \leftarrow$ new array
for each path P$\in R_p$ **do**
  $(F_p, M_p) \leftarrow (F, M) \in (P \subset F = field)$ ▷ Filtering
  sortDescending(F_p, by degree) ▷ Sorting
  $V \leftarrow$ `generateSingleChartSpecs`(F_p, F_{user}, T)
  $V_c \leftarrow$ `generateCombinationSpecs`(V, A)
  $V_C \leftarrow [V_C, V_c]$
return V_C

function GENERATESINGLECHARTSPECS(F_p, F_{user}, T)
  $F' \leftarrow [F_{user}, F_p]$ ▷ Sorted order by importance
  $V \leftarrow$ new array
  for each chart C in T **do**
    for each encoding slot C_e in C **do**
      try: $C_e \leftarrow f$ where $f \in F'$
    if $(\forall C_e \in C) \neq$ `NULL` **then**
      $V \leftarrow [V, C]$
  sort V by chart relevance
  return V

function GENERATECOMBINATIONSPEC(V, $N = 5$)
  $V' \leftarrow [V_1 ... V_N]$
  $V'_{spatial} \leftarrow$ `whichPositionallyAlign`(V')
  $V'_{color} \leftarrow$ `whichColorAlign`(V')
  return [$V'_{spatial}$,V'_{color}]

by Draco [5] and Voyager [4], [26], or the similar query language used by ShowMe [3]. An important limitation of these systems is that they are built on database-style queries for tabular data. Extending these systems to non-tabular data would not be trivial and is beyond the scope of our work. It would be possible to use a grammar, as ggplot does [46]. In fact, a number of libraries for non-tabular data, for example ggtree [47], build on the grammar of graphics in a very extensible way. We leverage ggplot and its underlying grammar by implementing chart templates (Figure 2) that allow us to specify and also evaluate the viability of generating a visualization. Our application of templates shares a similar spirit to Parameterized Declarative Templates, recently proposed by McNutt et al. [60], which enable the reuse and refinement of chart types with lower overhead; our implementation is comparatively lighter weight and leverages ggplot rather than implementing its own grammar. Our use of templates also includes a greater vocabulary for data roles and types than their proposal.

A chart template T contains slots of the chart type, data source (and if applicable metadata), and visual encoding channels (x, y, color, shape). Data and encoding slots contain additional parameters used to verify the suitability of some dataset or field to generate the encoding. Data slots contain parameters for the data source name (NA if unassigned) and data type (table, tree, network, spatial, image). Encoding slots contain parameters for field name (NA if unassigned), field type constraints (numeric or non-numeric), field data source, and whether that field is required to generate the encoding or optional. Data source name, field name and field data source are dynamically assigned at run time. Two example templates are shown in Figure 2.

The constraints on a visual encoding slot vary by the channel type, following established perceptual guidelines [2], [3], [5], [61]. The positional (x,y) encodings slot can be filled by either numeric fields or non-numeric fields with high cardinality. Color encoding slots are constrained to non-numeric fields that have fewer than 12 categories. Size encoding slots are limited to numeric fields.

At run time, `GEViTRec` sorts fields (nodes) in a path according to their degree of connectivity. Field nodes with a degree of 2 or more connect multiple datasets and are given higher priority than non-connecting fields when assigned to slots in chart templates. The user has the option to also specify fields that should appear in the final visual display and these are given the highest priority of all. `GEViTRec` attempts to assign fields in order of importance to appropriate visual encoding slots of chart templates. If a field does not meet the constraints for a encoding slot, our algorithm tries to fit it to another encoding slot or leaves the field unassigned. Only charts that have all of their required encoding slots assigned to a field move forward to the next pass where combination specifications are generated.

4.6 Aligning and Combining Charts

In the next stage, `GEViTRec` combines singleton charts together. Each path will result in a combination of multiple charts that have been explicitly visually coordinated to show linkages across multiple data sources, where the shared fields appear in multiple charts to express the linkage visually. At a minimum, a single chart is produced per each component of the graph.

Our recommender algorithm is intended to leverage existing software layers designed to render charts, such as D3 [56], Vega [57], or ggplot [46]. However, the problem of leveraging existing software when automatically coordinating static charts is not trivial. Once a chart is rendered into a small box of pixels, no aspects of its visual encoding such the colors or spatial position of marks can be updated or changed. We solve this coordination problem with a declarative approach that we call **gradual binding**: only partial specifications are initially generated, which are then modified in discrete stages. After a specification is finalized and complete, it is passed along to a charting library for rendering. This gradual approach allows `GEViTRec` to use extremely minimal initial specifications for the singleton charts and for the type of combination. The algorithm then automatically derives additional specifications that enforce the necessary consistency within each chart to instantiate the requested combination, so that all of the rendered boxes of pixels can simply be concatenated together.

The three chart combinations supported in `GEViTRec` are positional alignments, where two or more charts share a common field in the positional encoding slot; color alignments, where the same field is assigned to a color encoding slot for two or more charts; and unaligned combinations, where there is no visual linkage or shared fields. These combinations are not mutually exclusive: all three may

simultaneously occur. The GEViT study [6] also identified small multiples as a common combination within genEpi. Our algorithm does not support that combination type, to keep the scope of GEViTRec tractable; handling coordination seamlessly between all three of these combination types is an open research challenge that we leave for future work. We use the term alignment here to refer to the general idea of aligning ranges, when building visualization specifications from paths in the data source graphs, that by definition share common domains. The mechanisms of alignment are different between color and positional alignments.

Color aligned combinations will automatically apply a common color palette for shared attribute fields across multiple chart types. Color alignment is only facilitated between charts that have a common field assigned to their color encoding. At run time, GEViTRec recognizes this commonality and enforces a common color palette across the individual charts. Although the automatic coordination of color between views is supported in many existing systems for small multiples, most previous systems do not support this capability across multiple and diverse chart types.

For combinations facilitated by **positional alignments**, GEViTRec must ensure that charts have a common positional axis (x or y) along either the horizontal or vertical direction. The two considerations are whether it is feasible to positionally align charts, and if so how to orient them.

Determining a shared axis of alignment is straightforward when combining charts that all use Cartesian coordinates, with a shared field either the x or y axis, as is the case with many common statistical charts. However, domains with a diversity of chart types, including genEpi, require more extensive consideration in finding a shared axis. The solution we propose in GEViTRec is to manually analyze the full set of charts used in the domain design space (through discussions between the authors), to determine viable combinations where it may be possible to establish a shared axis. The result of our analysis is a viability matrix, shown in Figure 3, that is used by the GEViTRec algorithm to automatically determine whether positional alignments are viable. We assigned each combination into one of three categories: possible and straightforward to programmatically support, possible but not easy to programmatically support, and not possible. We made these determinations of compatibility based upon the types of axes (numeric and non-numeric) and coordinate systems of the chart. For example, a histogram contains two axes (x and y) that both have numeric data, and thus is theoretically compatible with other charts types that have numeric axes. Our current GEViTRec implementation only supports possible and programmatically feasible positional combinations.

The square matrix in Figure 3 addresses the 20 chart types handled in GEViTRec out of the 25 contained in the GEViT design space. Only 17 rows and columns are present because we collapsed several structurally identical chart types into the same rows: Tree represents both phylogenetic and dendrogram trees; Geographical Map includes choropleth maps; and Tables includes genomic alignments. We omit the 4 chart types of images, networks, pie charts, and Venn diagrams: we determine that these charts are unlikely be part of positionally aligned combinations because they lack a horizontal or vertical axis to facilitate such combinations. For example, a pie chart uses a radial coordinate system, whereas nearly all genomic charts use Cartesian coordinates. For network diagrams, the specific coordinates of the points often have no spatial meaning and can occlude one another if projected either horizontally or vertically; Venn diagrams have similar problems. The pixel space within images is sufficiently unrelatable to other genEpi data that it is typically not informative to positionally align other charts to them (with the exception of pulse field gel electrophoresis images and a few other specialized cases).

While some charts are straightforward to reorient, other chart types should not have their positional coordinates altered because it would inappropriately distort the information they contain. We again propose a solution based on manual analysis of the charts used in a domain design space, and then using those results programmatically within our recommendation algorithm. We analyzed all of the chart types that occur in the GEViT design space and identified the following 3 types as positionally immutable: trees, geographic maps, and images (with some exceptions). We refer to the positionally immutable charts as *lead charts* and the others as *support charts*. The GEViTRec algorithm is constrained to generate specifications with only one or zero lead charts; if no charts in the specification are positionally immutable, a lead chart is randomly selected. All of the support chart specifications are modified to rotate each one to share a common axis with the lead chart. A common scale is then automatically generated and applied to all charts based upon the scale of the lead chart.

GEViTRec may also generate unaligned combinations, where there are no shared fields between visual encodings. This situation can arise when two datasets are linked by a common domain (for example, a data ID), but this variable is not included in the visual encoding. The purpose of aligned combinations is to get different 'snapshots' of the same data, similar to Voyager or ShowMe, that are still grounded in data source relationships. Our prior study found this combination to be less common, so these paths are more likely to have low ranks than high ranks.

The gradual binding process concludes when complete chart specifications have been derived that fully adhere to the indicated combination types. Any stylistic defaults built into the underlying charting libraries will be inherited, unless they have been explicitly overridden.

To summarize, the algorithm can propose combinations containing a variable number of charts with a mix of positionally aligned, color aligned, and unaligned chart pairs. It takes a best-effort approach to alignment, choosing from a combinatorially large space of possibilities. Positional alignment is favored and the algorithm attempts to rotate charts to create positional alignment when feasible, but there is no requirement to use positional alignment for all charts. It falls back to simpler proposals that use color alignments for a potentially wider range of charts compatible with that approach, or even unaligned pairs as needed. In cases with multiple possibilities for combinations, priorities from the guiding VPDS could dictate a ranking of one combination as higher than another. When there are still multiple equally ranked possibilities, the algorithm resolves remaining ties by ranking simpler combinations with fewer charts ahead of those with a higher total number of charts.

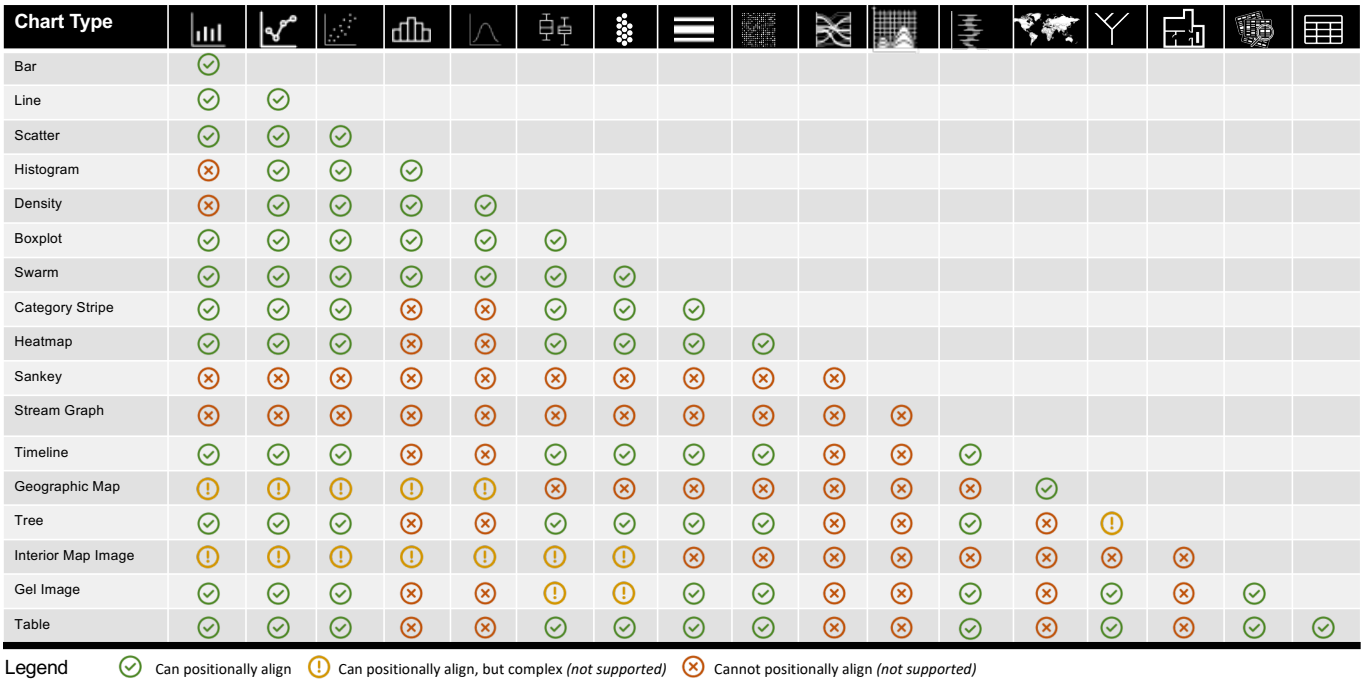

Fig. 3. **Compatible chart types for positional alignments.**

4.7 Arranging and Rendering Boxes of Pixels

In the final algorithm stage, arranging and rendering the charts is straightforward because the finalized specifications emerging from the previous stage have harmonized alignments and orientations, as the outcome of the gradual binding. The algorithm assumes access to a charting library that will take a complete specification and return a rendered box of pixels. The only further requirement is to arrange the boxes of pixels that represent rendered individual charts into a grid. By default, `GEViTRec` creates a 2×3 grid; that configuration is dynamically modifiable at run time. The arrangement depends upon the types of combinations that are feasible and any user specifications to modify the number of charts that compose a view. The final output of the recommender algorithm is a single large box of pixels, showing the complete combination of static visualizations with linked information content.

5 Results

We implemented `GEViTRec` as an end-to-end proof of concept system to investigate multiple datasets, compare to alternative tools, and finally to conduct an evaluation with experts. Next, we use both real and synthetic genEpi data to assess `GEViTRec`'s performance. In this section, we briefly describe these datasets and show one example chart combination for each. Due to limitations of space, we provide an overview of the implementation; a detailed walk through vignettes is in the supplemental materials.

5.1 Real Ebola Outbreak Dataset

The real genEpi dataset collection is publicly available data from the 2014-2016 Ebola outbreak [62]. It includes a phylogenetic tree for roughly 1610 Ebola virus genomic samples, one from each affected person. The spatial dataset has case counts for the affected nations. A tabular dataset contains additional information for each sample. The results shown in Figure 4 were generated in approximately 30 seconds using a 2017 MacBook Pro.

Figure 4C shows the highest ranked view created by `GEViTRec`, a 2×3 grid of 5 charts. The top row shows a positionally aligned combination of the phylogenetic tree, scatter chart, and heatmap. The phylogenetic tree was chosen by the algorithm to be the lead chart, so the vertical ordering of the phylogenetic tree leaves (representing affected people) is shared with the scatter chart encoding the country of the case, and with the heatmap that encodes the temporal progression of the outbreak. This positional alignment is explicitly indicated by the `combo_axis_var` label stating that the charts share a common y-axis and can be read together horizontally across the row. The bottom row has two charts showing case counts, one a choropleth map that contains fine-grained information about cases at the region level within each country, and the other a bar chart aggregating cases into the country level. The geographic map in the bottom row is color aligned with the phylogenetic tree in the top row, where the shared field is `country`. Figure 4A shows the extremely concise `GEViTRec` code required to generate this view.

5.2 Synthetic Simple Dataset

The small synthetic dataset that we created as a simple example to demonstrate the capabilities of our system features four fabricated datasets with 13 samples in each: tabular data, a phylogenetic tree, genomic data, and a pulse field gel electrophoresis (PFGE) image. The resulting 22 combinations, some of which are shown in Figure 5, were generated in approximately 17 seconds, again by a 2017 MacBook Pro.

The fourth-ranked view generated by `GEViTRec` is shown in Figure 5A, alongside a diagram that we manually created to illustrate the types of visually coherent combinations in that screenshot in Figure 5B. The three positionally

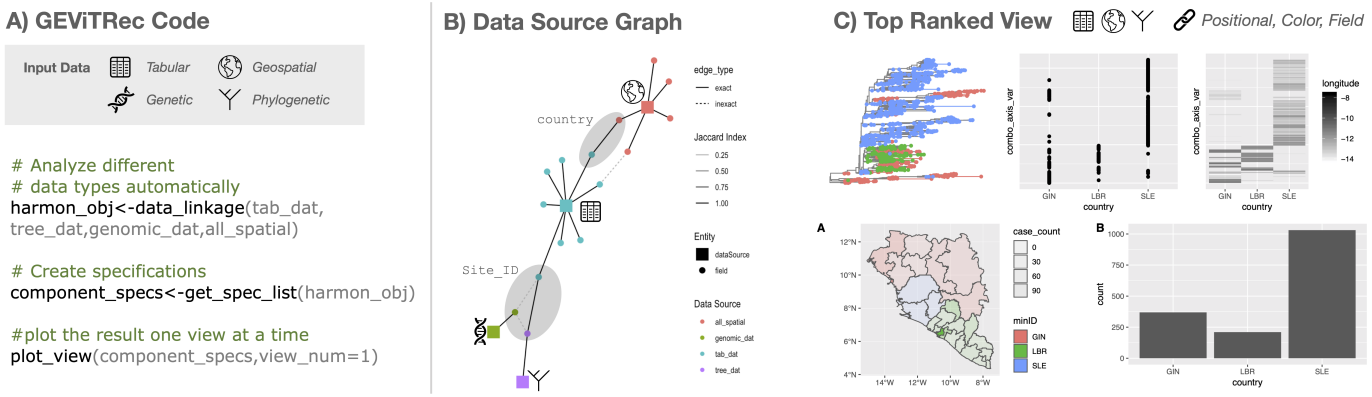

Fig. 4. **GEViTRec results: real Ebola outbreak data.** A) The GEViTRec code required to generate this view. B) The data source graph generated by GEViTRec. C) The highest ranked view generated by GEViTRec contains five charts of different types, featuring a positionally aligned combination across the three top-row charts and color alignment between the tree on the top row and the map on the bottom.

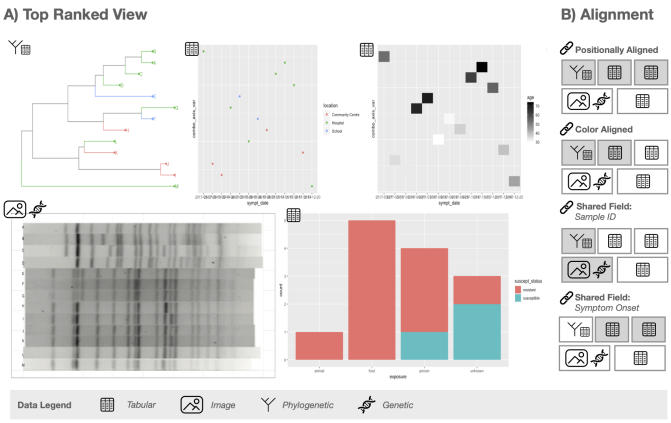

Fig. 5. **GEViTRec results: synthetic genEpi data.** A) The fourth ranked view generated by GEViTRec for simple dataset with 13 samples. B) Diagram depicting the types of alignments between the five charts.

aligned charts in the top row all share the same vertical ordering for samples, dictated by the phylogenetic tree. The left tree and the middle scatter chart are color aligned through the location field, and the middle scatter chart and the right heatmap showing peoples' ages have a shared field, the date of symptom onset, which has been chosen as the horizontal axis for both. The sample ID shared field appears in both the tree on the top left and the PFGE image on the bottom left, but these two charts are not positionally aligned: the vertical ordering is different. Neither trees nor gel images can be reordered, so GEViTRec did not attempt to align them positionally. Notably, the stacked bar chart on the right showing exposure vectors and susceptibility status does not share fields with any of the other charts. They occur within the tabular dataset, which does have shared fields with the other datasets, so it is within the same connected component of the data source graph. This example shows how an unaligned combination can usefully provide additional information that may be of interest during data recon, which should not be limited to only charts that can be aligned with each other.

6 Comparison to Existing Tools

We compare the visual encodings generated by GEViTRec to those from the Voyager [26], Draco, [5] and ShowMe [3] visualization recommender systems, as well as to two state-of-the-art genEpi tools that were manually curated by humans, Nextstrain [48] and Microreact [49].

6.1 Comparison to ShowMe, Voyager, and Draco

Although earlier recommendation systems such as ShowMe [3], Voyager [26], and Draco [5] are not designed to handle heterogeneous data types and are not optimized for the genEpi domain, we compare to them as the closest previous work in automatic creation of visual encodings.

Each of these systems is capable of making one or more recommendations for visual encodings provided a singular tabular dataset as input. Any genEpi data that could be transformed into a single tabular dataset, including preprocessing to combine multiple tables into one, would thus be viable for these tools. In contrast, GEViTRec is explicitly designed to handle multiple datasets and automatically creates a graph representation of these inputs, thus removing the need for the user to combine the data themselves. This graph representation of data is powerful not only for internal computation, but also for illustrating to the user the connections between their datasets in support of data recon and to help them verify data quality.

6.1.1 Overall Differences

Once datasets are loaded, all of these systems make fairly rapid recommendations through different mechanisms. ShowMe [3], Voyager [26], and Draco [5] all use some notion of graphical efficacy to rank and prioritize visualization recommendations. ShowMe and Voyager use a set of manually curated weights and a rule based system that is held fixed for all datasets. Draco uses a combination of user defined constraints and learned metrics of efficacy to prioritize visualization recommendations. To instantiate the Draco approach of learning metrics from perceptual experiments, one would need to conduct many additional experiments on visual encodings relevant to genomic epidemiology, such as phylogenetic trees, genomic maps, and even images, to make full use of such an inference engine. The user must also programmatically specify constraints.

Considering chart types, both Voyager [26] and Draco [5] principally recommend common statistical charts such as scatter plots, bar charts, and histograms. ShowMe [3] additionally recommends maps. None of these systems can make

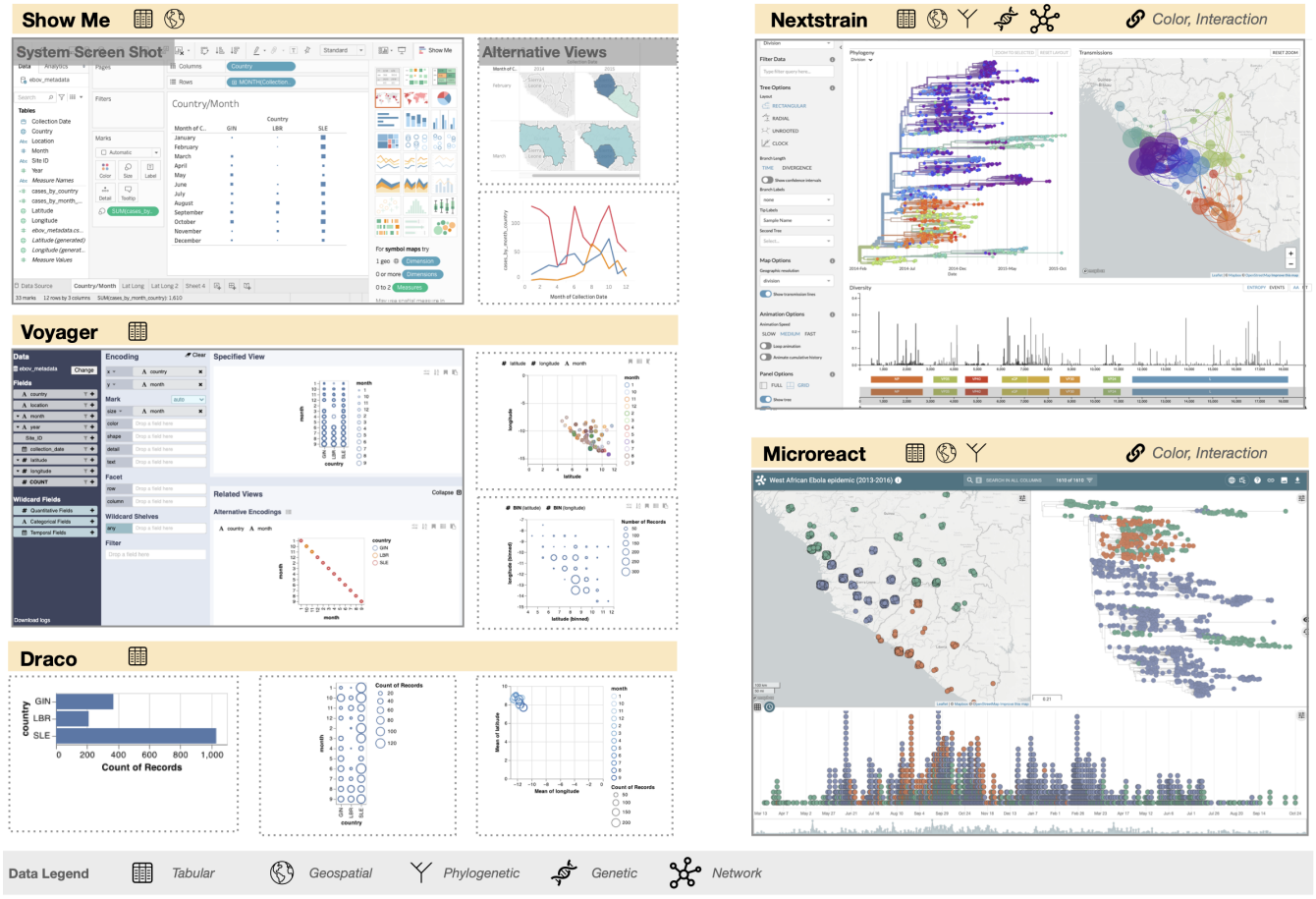

Fig. 6. We summarize the visualization recommendations of ShowMe, Voyager, and Draco alongside the human curated dashboards of Nextstrain and Microreact. Where applicable, we show a screen shot of the whole system (solid border outline) and a subset of alternative views (dashed border outline) that are generated by the system. For ShowMe and Voyager, the UI also contains the specification in the form of data columns dragged to encoding positions. The specifications for Draco are in the Supplemental Materials. Adjacent to the name of each system we also summarize the types of data sources that each system supports as well as the way that visual coherence and, if applicable, interaction is used to link information between charts. Both Nextstrain and Microreact have rudimentary linked highlighting, applicable to only a single point at a time through a click operation. Finally, Nextstrain and Microreact have pre-defined specifications of the visualizations that are created by the tool designers and exist as a fixed set of visualization templates in the interface that are populated by fields of the input data; they do not have alternative views. The directly comparable GEViTRec results are shown in Figure 4.

recommendations that include the trees, images, and tables that are routinely visualized in the genEpi domain.

Considering chart combinations, ShowMe [3] only generates a single visual encoding per Tableau worksheet. Although Tableau workbooks can have many worksheets, combining visual encodings from sheets into a dashboard requires manual action from the user. Voyager [26] produces more than one visual encoding and they may be linked through shared x or y axes or color. Users can pick and pin visual encodings they wish to review in greater detail later. However, these combinations are not deliberately created with the intent of coordination, but are instead an artifact of surfacing as many alternative views of the data as is reasonable. Draco [5] can also programmatically produce multiple visually coherent charts, but it is left up to user to facilitate the coordination process by explicitly defining a set of constraints. The ability of GEViTRec to automatically produce visually coherent chart combinations avoids the manual post-processing a user must undertake to combine and coordinate singleton charts with the other systems.

6.1.2 Comparison on Ebola Outbreak Data

To demonstrate the differences between these systems in action, we elicited visualization recommendations from ShowMe, Voyager, and Draco. As input we provide eight columns of tabular data from the Ebola Outbreak dataset. We limit our comparison to the tabular data as neither ShowMe, Voyager, or Draco support phylogenetic data as input and only ShowMe can take spatial data inputs; GEViTRec can take tabular, phylogenetic, and spatial data inputs. Figure 6 shows a subset of the results, focusing on the main findings of visualizing combinations of the country, month, latitude, and longitude fields. We choose these fields for their data diversity (nominal, ordinal, and numeric) and also for the special interpretation of latitude and longitude as geographic data. We also take additional consideration of whether these systems automatically generate univariate summaries of the data. The full analysis is in Supplemental Sec. S7.

In comparison to GEViTRec these tools have both limitations and advantages. The primary limitation, as noted in Section 6.1.1, is the inability to support non-tabular data.

A second limitation is that all of these systems require the specification of both a data set and some field specifications, for example dragging pills to channels as is done with ShowMe and Voyager, or writing programmatic code as is done in Draco. An exception is Voyager's automatic univariate summaries. By comparison, GEViTRec only requires that the user specify the data sources they wish to visualize; the user may also specify fields of interest to include in the outputs, but these are not strictly required.

A third limitation is that these systems do not suggest visually coherent chart combinations. The premise of our research is that visual coherent combinations are important. While ShowMe, Voyager, and Draco primarily produce one visualization at a time, these systems do support the generation of alternative data visualizations. For example, ShowMe can suggest alternative chart types and even chart specifications and uses a red outline in the suggestion panel to highlight a highly ranked chart (see Figure 6). Voyager also generates alternative specifications automatically, and will produce univariate summaries of each field in the dataset without any specifications from the user. Draco allows the user to explore a potentially vast combinatorial space of visualizations by allowing users to specify a set of constraints. Users can create dashboards using ShowMe's recommendations, or they can pin results from Voyager's alternative suggestions; they may also specify constraints from multiple chart types as well. ShowMe also supports multiple linked views through interaction. The functionality to explore alternatives and compose multiple views approximates what GEViTRec does, but in these other systems any visual coherence is a side effect of deliberate choices the user has made rather than being produced automatically. Achieving any visual coherence through manual processes is not trivial and requires a time commitment from the user. Finally, the recommendations of these systems are based on perceptual efficacy and do not encompass alternative modes of ranking visualizations such as the domain-targeted VPDS. For tabular data alone, ShowMe, Voyager, and Draco may outperform GEViTRec, which extends the state of the art in visualizing multiple data sources of different data types.

6.2 Comparison to Nextstrain and Microreact

We now compare the results of GEViTRec to Nextstrain [48] and Microreact [49], two human-curated visualization dashboards that have been developed by genEpi domain experts. While neither of these systems were explicitly created for data reconnaissance, the overlap between the presentation and data recon design spaces means that they do offer some reconnaissance affordances. In contrast to ShowMe, Voyager, and Draco, the Nextstrain and Microreact systems are able to support non-tabular input types, notably, genomic, phylogenetic, and spatial data inputs. In Figure 6 we indicate the different input types supported by all of these systems. GEViTRec also supports these non-tabular data types as well as image data. Furthermore, Nextstrain and Microreact use visualization templates that are fully specified by the domain experts. ShowMe, Voyager, and Draco use a query language approach that theoretically allows them to flexibly generate a variety of visualizations, but these systems have not yet been demonstrated on the types of visualizations that support genEpi analyses. Since query language approaches are highly optimized for databases, it is not obvious how they might extend to data that cannot be easily represented in a tabular form. While the chart template approach is less flexible than using a query language, it enables the domain experts to visualize data from multiple sources more easily. With GEViTRec we sought to strike a balance, using templates to enable the visualization of multiple data sources while introducing flexibility in the creation of visually coherent combinations of their corresponding chart types.

Comparing the results of GEViTRec (Figure 4) to Nextstrain and Microreact, we observe that all of these systems feature a phylogenetic tree and a map. GEViTRec assigns high priorities to these over other chart types based upon the distribution of these chart types in the VPDS. Nextstrain also shows a genomic map, while Microreact shows a timeline. While GEViTRec adaptively responds to input datasets, Nextstrain does not and relies on a specific analysis stack; Microreact supports only two datatypes (tabular and tree) that can be loaded via the interface. Neither can show any other visual encodings. Nextstrain also shows a limited hand-curated set of field attributes regardless of the dataset, while Microreact allows user to select some fields to visualize. Nextstrain and Microreact show color aligned combinations, but these fields must be selected by the user or are hard coded by the developers. Compared to Nextstrain and Microreact, our current algorithm is parsimonious in its use of color channels and does not prioritize redundant visual encodings. GEViTRec automatically chooses attributes to encode based upon a combination of data linkage and user specification, and these will vary according to the input data. Also, data need not be loaded on external servers for GEViTRec to generate visualizations. Some of GEViTRec's design choices are less effective than Nextstrain and Microreact, especially for encoding temporal data. While GEViTRec does not reproduce all aspects of the human curated visual design, its automatically generated recommendations are much closer to these systems than the results of ShowMe, Voyager, or Draco.

7 Evaluation with GenEpi Experts

We have conducted a qualitative study with genomic epidemiology experts to evaluate the usefulness, interpretability, and actionability of GEViTRec's visualization recommendations in a data recon setting. We remind the reader that all study materials including anonymized study data are available in the Supplemental Materials.

7.1 Study Procedures

Study participants were walked through GEViTRec's visualization recommendation procedures through a chauffeured demonstration from the study administrator using the two datasets described in Section 5. The demonstration is documented in Supplemental Sec. S3. Chauffeured demonstrations showcasing both datasets were carried out using the RStudio IDE and run within an R Notebook environment. Participants were shown the top five views generated by GEViTRec for each dataset and were asked to provide a more detailed assessment and interpretation of

the Ebola dataset views specifically. Participants could also interrupt the study administrator at any point in time to ask for clarification of either the algorithm's procedures or the data. Following the demonstration, participants were asked to complete an online questionnaire.

7.2 Data Collection and Analysis

We recruited ten genEpi domain experts to participate in our study. We verified that these individuals had a background and expertise in genEpi by looking at their affiliations and, if applicable, their publication record. With the exception of two individuals, the participants were not known to the authors prior, and those who were known to us were not current collaborators on any active project. One participant was a graduate student in genomic epidemiology and all others were professionals who either consulted or worked within regional or national public health agencies. We did not have an exclusion criterion on the basis of geography, but all of our participants were from either Canada, the United States, and the United Kingdom. We collected data through an online questionnaire, administrator session notes, and audio session recordings; results from all three sources were analyzed together.

Domain experts had varying degrees of exposure to genomic data, but all participants routinely analyzed heterogeneous data sources to conduct epidemiological investigations. The majority of experts identified as bioinformaticians and/or surveillance analysts. All experts had some exposure to the R language for statistical analysis and visualization (5 beginner, 4 intermediate, and 1 expert level of self-reported proficiency). Nearly all participants frequently developed data visualizations to understand and communicate their results. When visualizing data, the majority of participants used R, Excel, Google Docs, or Tableau on a daily or monthly basis.

7.3 Findings and Interpretation

Supplemental Sec. S4 contains all questionnaire results, Sec. S5 the study administrator notes, and Sec. S6 the transcripts of the session recordings. Here, we summarize participants' assessment of result relevance and interpretability.

■ **Usability Assessment of GEViTRec**

Participants were asked to assess their perceived usability of `GEViTRec` from the chauffeured demonstration.

Overall, participants had a positive impression. Participants all strongly or somewhat agreed they could quickly find a useful data visualization from the set of suggestions provided by the system and that it was much more useful for the system to automatically come up with visualizations than for them to come up with visualizations themselves. Participants indicated that `GEViTRec` visualized the data in ways they would not have thought of, but also indicated that `GEViTRec` did not show some of the visualizations they had expected. Their satisfaction with `GEViTRec`'s results seemed to correlate with whether the participant felt they had a particular research question in mind: *"I might have some very specific question that I want to answer. As opposed to when I am the researcher thinking about what is genomic epi of this disease. I might have much broader question which are address by this earlier image"*. Participants strongly agreed that `GEViTRec` produced relevant visualizations that help them understand the data.

The data source graph is valuable for understanding data. Participants also strongly or somewhat agreed that it was understandable how the data connected together. In their discussion with the study administrator and their textual responses, they emphasized how the data source graph was an especially useful way to understand their data: *"it's really nice to see that visualization of how all of these different datasets can connect together"*. Participants indicated that they would like the data source graph to have more interactivity and would use it more centrally in their analyses. Their suggestions open the way to many potential future directions as `GEViTRec` continue to evolve, to combine the power of automatically computed visual coherence of static views as we present here and the power of interactivity as has been heavily explored in previous work.

GEViTRec is fast and simple. Participants were impressed with the speed that data were linked and visualized. They also indicated that the limited amount of coding was manageable, but many felt that a dedicated user interface would have simplified it further and would increase its impact. Several participants also validated our claim that generating an R-based tool was effective because of growing support and infrastructure in their organizations for R as an analysis framework. As such, `GEViTRec` could fit easily with their existing workflows. Several participants asked about the data quality required to generate the data visualizations they were shown in through the demonstration. They were informed that `GEViTRec` acts as a data viewer and will visualize whatever quality of data is given. We suggested to participants this quality-agnostic approach of `GEViTRec` was a beneficial feature for data recon, especially when combined with the data source graph visualization, because it could help participants triage and wrangle their data within the R environment. Participants agreed that this `GEViTRec` usage would also fit within their workflows.

■ **Interpretability and Relevance of GEViTRec Output**

We assessed the relevance and interpretability of `GEViTRec` results by asking participants to interpret them on the Ebola dataset for the top ranked view.

Output and ordering is relevant and interpretable. All participants agreed that both the views generated by `GEViTRec` and the order they were presented in were useful and would help them with their current analysis. One participant stated that *"these are all extremely useful visualizations. And having essentially a menu to choose from is fantastic"*. The interpretability of the views was further by validated by participants' ability to dive into the interpretation of the Ebola outbreak data and begin answering questions of *"when, how much, and where"* the outbreak occurred. Some participants commented on the redundancy of information between the chart types saying that the views could more effectively communicate the diversity of data types and attributes. However, some participants stated that the redundancy of information encoding in the charts was effective because it allowed them to *"tease out different aspects of it [data] that you might be interested in"*.

Addressing issues of data scale. The majority of participants indicated the Ebola data was more difficult to interpret

than the synthetic data, a surprise to us due to the well-publicized nature of that outbreak. However, participants indicated the amount of data and variety of chart types together was overwhelming: *"It's not as easy to process as the sample [synthetic] data was, which you could take one look at it and say, I totally understand everything that's going on. This one would take me a few more minutes of just sitting and thinking and needing to digest a little bit."* Data aggregation, as Draco [5] or Voyager [4] does, is one potential solution, and GEViTRec does so for certain data and chart types such as bar charts. However, aggregating procedures for more complex data types, like phylogenetic trees, are not trivial; domain expertise is currently required. Below, we discuss how layout, effective text summaries, and improved chart coordination may help address these issues.

■ **Comparison to Nextstrain and Microreact**
Participants were shown the Ebola outbreak dataset on the Nextstrain [48] and Microreact [49] applications and asked to discuss their perceived differences between human generated (Nextstrain, Microreact) and machine generated (GEViTRec) views.

Overall, participants felt that the Ebola data was overwhelming, regardless of the data visualization tool, and that they needed more help understanding how to approach and consume the information. However, participants appreciated that Nextstrain and Microreact had interactive components that allowed them zoom in and filter the data. Future work to allow this ability for GEViTRec would be straightforward with R/Shiny integration. Several participants liked Nextstrain's animation of disease transmission. They also felt that the human curated visualizations were more aesthetically pleasing and more carefully coordinated, but those from GEViTRec contained information that the human generated views did not. Participants also felt that both Nextstrain and Microreact were suited for very specific tasks, and not helpful if you wanted to see a different view of the data or had a different research questions :*"phylogeography is not very interesting to what I am doing, so instantly one-third of that [...] visualization is not that useful"*. Participants indicated that it was important for data visualization tools to adapt to different datasets as GEViTRec does.

■ **Layout and Summarizing of Results**
Participants liked that different chart types were linked by positional or color alignment, or by having common attribute fields on the axes. However, they felt that data could be presented more effectively to help them consume the information. Several asked for text summaries that describe the data and the type of visualization, a finding echoing our prior study on generating a report design for genomic epidemiology data [10]. Participants also thought it could be helpful to lay out data in a 'scrollytelling' vertical layout, and one suggested that information could be further prioritized to show the simplest data first and more complex data (and their attendant visualizations) later. Participants also indicated that different charts should be sized according to their data density and relative importance for a more justifiable data-ink ratio. Optimizing the aspect ratio and size of each chart relative to information content would be another useful area for future research.

■ **Value for Data Reconnaissance**
GEViTRec was designed to provide users with a quick first glance of the data so that they can assess its value or the need to pursue additional datasets. Participants agreed that GEViTRec fulfilled this intended data recon purpose: *"This tool is extremely useful for data exploration, particularly where there are incomplete, or highly varied types of data that must be integrated and displayed. Very useful for public health surveillance and initial review of data."*

The boundary between data recon and standard data analysis is certainly not sharp, and GEViTRec may sometimes be versatile enough to suffice for preliminary analysis. Participants repeatedly stated they saw GEViTRec as a useful hypothesis generation mechanism, helping them to see multiple views of their data that they had not considered. Many participants differentiated between GEViTRec functionality and the more targeted operation of creating a specific data visualization to answer a specific question; essentially GEViTRec was seen as less effective for hypothesis verification. Hypothesis generation more closely aligns with the goals of data recon than hypothesis verification, so it would be fruitful to explore this particular tension in follow-on work. Ultimately, participants found value in being able to both generate a specific visualization and see alternative visualizations: *"when you're discovering things, then maybe it is good to have things that are generated without human input"*.

■ **Further customization to participant background**
All participants wanted to use GEViTRec on their own data and analyses. When queried about more specific datasets they would apply GEViTRec to, all participants identified data with a large number of samples, sourced from different data types, and with large numbers of diverse attribute fields. However, several participants wanted further optimization of the relevance ranking to include participant's background. Interestingly, participants also stated that they still wanted to see the multiple views of the data to preserve the hypothesis generation and exploratory abilities that they perceived GEViTRec afforded: *" relevance is very specific for that [genEpi] person. [...] having that [research question] as a criteria [...] would be very helpful to target your visualizations. [...] there's the counter point that it's sometimes helpful to see things that you are not asking the question for. So that the top ranked view was more tailored to them, but that lower ranked views needed not necessarily be."*

8 DISCUSSION

Heterogeneous and multidimensional data are already the norm in many domains and stakeholders are increasingly expected to use these complex data to derive informed actions [63]. In our own collaborations we have seen stakeholders struggling to understand landscapes of heterogeneous data. We believe that these challenges are not limited to genEpi, but extend more broadly to other data science applications. In prior work [1], we developed the terminology of data recon and a conceptual framework of how to tackle it to capture the challenges of these experts and to delineate their unmet needs. That framework identified a four-phase iterative cycle for data recon (acquire, view, assess, pursue) and emphasized the importance of experts being able to rapidly see visual encodings that provide

an overview of their data. While we had identified visualization recommender systems as a viable solution to the challenges of data reconnaissance, we also identified the limitations of previous systems that lessened their utility to experts in genEpi and other domains with heterogeneous data landscapes.

The algorithm that we present here is a significant advance over the previous state of the art that addresses the challenges of data recon in complex and diverse data landscapes, accelerating the *view* stage of the data recon process. We demonstrate that the `GEViTRec` proof-of-concept implementation provides rapid overviews across a wide variety of data types and produces actionable insights that experts can interpret and integrate into their existing workflows. Our evaluation provided evidence that our approach of using visually coherent combinations of static charts was a viable approach to data recon, supporting very fast overview comprehension without requiring the time investment of interactivity.

8.1 Domains Beyond GenEpi

The `GEViTRec` algorithm has many domain-agnostic elements, but also incorporates information from a domain-specific visualization prevalence design space (VPDS). While our specific implementation was in the genEpi domain, we intended this approach to be transferrable to other domains as well. The future work needed to fully validate our claim of domain independence would be to deploy an implementation in another domain and assess its utility. However, one current bottleneck in doing so would be the manual effort currently required to generate a VPDS. While the GEViT method for doing so requires a mix of automatic computation and human effort [6], a robust method that is fully automatic would address that problem. We do not solve the technical problem of automatically generating a VPDS in this work. Instead, what we demonstrate here is how a design space can be used to inject domain expertise into recommendation systems. We hope that our work makes headway on this "chicken and egg" problem, motivating further work on automating the construction of a VPDS by showing the practical benefits that can be obtained from having one available. Moreover, our current approach also requires further manual analysis of VPDS results, such as the chart combination viability matrix, which would also benefit from automation. A more prosaic barrier to extending this work to other domains is the amount of tedious and time-consuming work required to integrate domain-specific software packages that visually encode new data types into a software system. Our current `GEViTRec` implementation is a proof-of-concept that such an end-to-end system, where the input of multiple heterogeneous data sources results in the automatically generated output of domain-relevant visually coherent combinations of charts, is viable and that further automation would be a worthwhile investment to extend it to other domains.

8.2 The Relevance of Prevalence

Our algorithm is built to prioritize chart types according to a relevance metric that is built on domain-specific prevalence, namely examining the visual design collections commonly created by domain experts, in contrast to the many existing recommender systems use graphical perceptual effectiveness to rank visual encodings. We point out the concern that relying on perceptual effectiveness as the sole ranking mechanism may not be sustainable at scale because of the large number of studies that would be necessary even to assess single charts, in light of the full range of visual encodings possible for a heterogeneous array of data types. Moreover, it is even more challenging to fully assess the perceptual implications of combinations of charts, because they introduce perceptual questions that are challenging to isolate in an experiment. Using information from a domain-specific VPDS does not preclude incorporating efficacy judgements. It would be fruitful future work to examine the trade-offs between perceptual effectiveness and domain prevalence, and how to combine them. If perceptual experiments provide adequate coverage for any specific visualization prevalence design space, then it would be possible to penalize relevant visualizations that are not perceptually effective.

One potential objection to defining relevance according to domain-specific prevalence is that domain experts lack a full awareness of the breadth of possible visualization designs and the trade-offs between them. Although we agree that individual domain experts may not be fully aware of how to visualize their data, we have observed that the collective strategies of a large group of experts can reveal a complex combinatorial design space. We found that using this domain-specific prevalence information in a recommender system leads to generating visual encodings that individual experts find intriguing and informative, even when they would not have generated them manually. Our approach represents only one of many possibilities; future visualization recommender systems may contain several different ranking metrics that may be optimized through some combination of machine learning and user input.

8.3 Future Work

In the process of developing and implementing `GEViTRec` we identified several areas of future work. We can continue to improve how `GEViTRec` lays out chart combinations, the aesthetic design of the charts themselves, and further tailor visualization output to an expert's context. We did not prioritize full optimization of display aesthetics and of every possible design nuance in our proof-of-concept implementation. Some types of chart configurations pose a number of research challenges on their own. For example, automatically resolving visual coherence for some of the not-supported positional alignments in Figure 3 would not be trivial. Another area for future exploration is supporting more constraints on the specifications of visualization charts. For example, a hard constraint that a certain field must be used for a specific encoding slot could be enforced in a similar spirit to Draco's use of hard and soft constraints [5], but tailored more precisely to the use case of creating visually coherent combinations. The addition of narrative elements, such as text, or staged displays of the data, could help to alleviate some challenges of interpreting visualizations containing a lot of data. In the longer term, there remain several interesting and challenging issues in generating visually coherent combinations, including integrating more complex types of combinations and

incorporating interactivity. By implementing an end-to-end system we surfaced several limitations that were not fully addressed by our present research. For example, resolving conflicts between potential chart combinations is handled in a straightforward way by our algorithm, and more sophisticated approaches could be explored. We intend for the detailed description of our technique, its use case, and the `GEViTRec` proof of concept implementation to inform future research objectives.

9 Conclusion

We have presented a novel algorithm for data reconnaissance through visualization recommendation, `GEViTRec`, that automatically generates visually coherent combinations of charts from multiple heterogeneous datasets. Our approach includes many domain-agnostic components, but also incorporates information about the prevalence of visualization usage within a specific target domain, in our case genomic epidemiology. We created a proof-of-concept implementation to demonstrate its capabilities using both real and synthetic data, and conducted a thorough qualitative evaluation with ten domain experts to validate our claims that our method quickly produces relevant and interpretable views of data.

Acknowledgments

We thank Madison Elliott, Steve Kasica, Zipeng Liu, Michael Oppermann, and Ben Shneiderman for their thoughtful comments and feedback. We also acknowledge and thank our study participants for their time and insights.

References

[1] A. Crisan and T. Munzner, "Uncovering data landscapes through data reconnaissance and task wrangling," in *Proc. IEEE Conf. VIS (Short Papers)*, 2019.

[2] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. Graphics*, vol. 5, no. 2, pp. 110–141, Apr 1986.

[3] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show Me: Automatic Presentation for Visual Analysis," *IEEE Trans. Visualization & Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.

[4] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager 2: Augmenting visual analysis with partial view specifications," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2017, pp. 2648–2659.

[5] D. Moritz, C. Wang, G. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, "Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco," *IEEE Trans. Visualization & Computer Graphics*, vol. 25, no. 1, pp. 438–448, Jan 2019.

[6] A. Crisan, J. L. Gardy, and T. Munzner, "A Systematic Method for Surveying Data Visualizations and a Resulting Genomic Epidemiology Visualization Typology: GEViT," *Bioinformatics*, vol. 35, no. 10, pp. 1668–1676, 09 2018.

[7] H. Lam, "A framework of interaction costs in information visualization," *IEEE Trans. Visualization & Computer Graphics*, vol. 14, no. 6, pp. 1149–1156, 2008.

[8] J. L. Gardy and N. J. Loman, "Towards a Genomics-informed, Real-time, Global Pathogen Surveillance system," *Nature Reviews Genetics*, vol. 19, no. 1, pp. 9–20, 2018.

[9] A. Crisan, J. L. Gardy, and T. Munzner, "On regulatory and organizational constraints in visualization design and evaluation," *Proc. Workshop Beyond Time and Errors: Novel Evaluation Methods for Visualization*, vol. 1, pp. 1–9, 2016.

[10] A. Crisan, G. McKee, T. Munzner, and J. L. Gardy, "Evidence-based design and evaluation of a whole genome sequencing clinical report for the reference microbiology laboratory," *PeerJ*, vol. 6, p. e4218, Jan. 2018.

[11] S. K. Card and J. Mackinlay, "The structure of the information visualization design space," in *Proc. IEEE Symp. Information Visualization (InfoVis)*, 1997, p. 92.

[12] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing sets and set-typed data: State-of-the-art and future challenges," in *EuroVis State of the Art Report*. The Eurographics Association, 2014.

[13] H.-J. Schulz, "Treevis.net: A Tree Visualization Reference," *IEEE Computer Graphics & Applications*, vol. 31, no. 6, pp. 11–15, Nov 2011.

[14] K. Morton, M. Balazinska, D. Grossman, R. Kosara, and J. Mackinlay, "Public data and visualizations: How are Many Eyes and Tableau Public used for collaborative analytics?" *ACM SIGMOD Rec.*, vol. 43, no. 2, pp. 17–22, 2014.

[15] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, "ManyEyes: a site for visualization at internet scale," *IEEE Trans. Visualization & Computer Graphics*, vol. 13, no. 6, pp. 1121–1128, 2007.

[16] T. Munzner, *Visualization Analysis and Design*, ser. A.K. Peters Visualization Series. Boca Raton, Florida: CRC Press, 2015.

[17] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 289–309, 2011.

[18] S. L'Yi, J. Jo, and J. Seo, "Comparative layouts revisited: Design space, guidelines, and future directions," *IEEE Trans. Visualization & Computer Graphics*, pp. 1–1, 2020.

[19] Z. Qu and J. Hullman, "Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring," *IEEE Trans. Visualization & Computer Graphics*, vol. 24, no. 1, pp. 468–477, 2018.

[20] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" *IEEE Trans. Visualization & Computer Graphics*, vol. 25, no. 1, pp. 682–692, Jan 2019.

[21] E. Dimara and C. Perin, "What is interaction for data visualization?" *IEEE Trans. Visualization & Computer Graphics*, vol. 26, no. 1, pp. 119–129, 2020.

[22] J. J. van Wijk, "Views on visualization," *IEEE Trans. Visualization & Computer Graphics*, vol. 12, no. 4, pp. 421–432, 2006.

[23] L. Battle and J. Heer, "Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau," *Computer Graphics Forum*, vol. 38, no. 3, pp. 145–159, 2019.

[24] J. Boy, L. Eveillard, F. Detienne, and J. Fekete, "Suggested interactivity: Seeking perceived affordances for information visualization," *IEEE Trans. Visualization & Computer Graphics*, vol. 22, no. 1, pp. 639–648, 2016.

[25] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE Trans. Visualization & Computer Graphics*, vol. 19, no. 3, pp. 495–513, 2013.

[26] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, D. Howe, and J. Heer, "Voyager: Exploratory analysis via faceted browsing of visualization recommendations," *IEEE Trans. Visualization & Computer Graphics*, vol. 22, no. 1, pp. 649–658, Jan 2016.

[27] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, "Effortless data exploration with Zenvisage: An expressive and interactive visual analytics system," *Proc. VLDB Endow.*, vol. 10, no. 4, p. 457468, Nov. 2016.

[28] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis, "SeeDB: Efficient data-driven visualization recommendations to support visual analytics," *Proc. VLDB Endow.*, vol. 8, no. 13, p. 21822193, Sep. 2015.

[29] A. Key, B. Howe, D. Perry, and C. Aragon, "VizDeck: Self-organizing dashboards for visual analytics," in *Proc. ACM Intl. Conf. Management of Data (SIGMOD 2012)*, 2012, p. 681684.

[30] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, "Foresight: Rapid data exploration through guideposts," in *Proc. IEEE Workshop Data Systems for Interactive Analysis (DSIA)*, 2017.

[31] J. Seo and B. Shneiderman, "A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections," in *Proc. IEEE Symp. Information Visualization*, 2004, pp. 65–72.

[32] G. Wills and L. Wilkinson, "AutoVis: Automatic visualization," *Information Visualization*, vol. 9, no. 1, p. 4769, 2010.
[33] V. Dibia and Ç. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Computer Graphics & Applications*, vol. 39, no. 5, pp. 33–46, 2019.
[34] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2019, pp. 128:1–128:12.
[35] O. Gilson, N. Silva, P. W. Grant, and M. Chen, "From web data to visualization via ontology mapping," in *Proc. Eurographics/IEEE Conf. Visualization (EuroVis)*, 2008, pp. 959–966.
[36] M. Cammarano, X. Dong, B. Chan, J. Klingner, J. Talbot, A. Halevy, and P. Hanrahan, "Visualization of heterogeneous data," *IEEE Trans. Visualization & Computer Graphics*, vol. 13, pp. 1200–1207, 2007.
[37] B. Mutlu, E. Veas, and C. Trattner, "Vizrec: Recommending personalized visualizations," *ACM Trans. Interact. Intell. Syst.*, vol. 6, no. 4, pp. 31:1–31:39, Nov. 2016.
[38] S. Kairam, N. H. Riche, S. Drucker, R. Fernandez, and J. Heer, "Refinery: Visual exploration of large, heterogeneous networks through associative browsing," *Comput. Graph. Forum*, vol. 34, no. 3, p. 301310, 2015.
[39] C. Xie, W. Zhong, W. Xu, and K. Mueller, "Visual analytics of heterogeneous data using hypergraph learning," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 1, 2018.
[40] P. Angelelli, S. Oeltze, J. Hasz, C. Turkay, E. Hodneland, A. Lundervold, A. J. Lundervold, B. Preim, and H. Hauser, "Interactive visual analysis of heterogeneous cohort-study data," *IEEE Computer Graphics and Applications*, vol. 34, no. 5, pp. 70–82, 2014.
[41] C. Nobre, N. Gehlenborg, H. Coon, and A. Lex, "Lineage: Visualizing multivariate clinical data in genealogy graphs," *IEEE Trans. Visualization & Computer Graphics*, vol. 25, no. 3, pp. 1543–1558, 2019.
[42] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit, "Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets," *IEEE Trans. Visualization & Computer Graphics*, vol. 20, no. 12, pp. 2023–2032, 2014.
[43] J. Huerta-Cepas, F. Serra, and P. Bork, "ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data," *Molecular Biology and Evolution*, vol. 33, no. 6, pp. 1635–1638, June 2016.
[44] G. Dudas, "Baltic," https://github.com/evogytis/baltic, 2019, accessed: 2019-03-25.
[45] E. Paradis and K. Schliep, "ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R," *Bioinformatics*, vol. 35, pp. 526–528, 2018.
[46] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. New York, NY: Springer-Verlag, 2016.
[47] G. Yu, D. K. Smith, H. Zhu, Y. Guan, and T. T.-Y. Lam, "ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data," *Methods in Ecology and Evolution*, vol. 8, no. 1, pp. 28–36, 2017.
[48] J. Hadfield, C. Megill, S. M. Bell, J. Huddleston, B. Potter, C. Callender, P. Sagulenko, T. Bedford, and R. A. Neher, "Nextstrain: Real-time tracking of pathogen evolution," *Bioinformatics*, vol. 34, no. 23, pp. 4121–4123, Dec 2018.
[49] S. Argimón, K. Abudahab, R. J. E. Goater, A. Fedosejev, J. Bhai, C. Glasner, E. J. Feil, M. T. G. Holden, C. A. Yeats, H. Grundmann, B. G. Spratt, and D. M. Aanensen, "Microreact: Visualizing and Sharing Data for Genomic Epidemiology and Phylogeography," *Microbial Genomics*, vol. 2, 2016.
[50] A. Cruz, J. P. Arrais, and P. Machado, "Interactive and coordinated visualization approaches for biological data analysis," *Briefings in Bioinformatics*, vol. 20, no. 4, pp. 1513–1523, 03 2018.
[51] J. C. Roberts, "State of the art: Coordinated multiple views in exploratory visualization," in *Intl. Conf. Coordinated and Multiple Views in Exploratory Visualization (CMV)*, July 2007, pp. 61–71.
[52] C. Weaver, "Building highly-coordinated visualizations in improvise," in *IEEE Symp. Information Visualization*, Oct 2004, pp. 159–166.
[53] A. Satyanarayan and J. Heer, "Lyra: An interactive visualization design environment," *Computer Graphics Forum (Proc. EuroVis)*, 2014.
[54] 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 *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2018, pp. 123:1–123:13.
[55] D. Ren, B. Lee, and M. Brehmer, "Charticulator: Interactive construction of bespoke chart layouts," *IEEE Trans. Visualization & Computer Graphics*, vol. 25, no. 1, pp. 789–799, 2019.
[56] J. Heer and M. Bostock, "Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2010, pp. 203–212.
[57] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Trans. Visualization & Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2017.
[58] A. Slingsby, J. Dykes, and J. Wood, "Configuring hierarchical layouts to address research questions," *IEEE Trans. Visualization & Computer Graphics*, vol. 15, no. 6, pp. 977–984, Nov 2009.
[59] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist, "Atom: A grammar for unit visualizations," *IEEE Trans. Visualization & Computer Graphics*, vol. 24, no. 12, pp. 3032–3043, 2018.
[60] A. M. McNutt, "Integrated visualization editing via parameterized declarative templates," Feb 2021. [Online]. Available: osf.io/cture
[61] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *Journ. American Statistical Association*, vol. 79, no. 387, pp. 531–554, 1984.
[62] G. Dudas, L. M. . M. A. Carvalho, P. Lemey, and A. Rambaut, "Virus Genomes Reveal Factors that Spread and Sustained the Ebola Epidemic," *Nature*, vol. 544, p. 309, Apr 2017.
[63] T. Hey, S. Tansley, and K. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009. [Online]. Available: https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/

Anamaria Crisan is a Research Scientist at Tableau. She completed her PhD in Computer Science in 2019 at the University of British Columbia. Her research explores visualization through the data analysis life cycle with an emphasis on biomedical applications. Her work on this project was supported by a Vanier Award.

Shannah E. Fisher conducted this work as an undergraduate student in Computer Science and Cell and Developmental Biology at the University of British Columbia. She is now with the University of Saskatchewan.

Jennifer L. Gardy is the deputy director, Surveillance, Data, and Epidemiology for the Malaria team at the Bill & Melinda Gates Foundation. Before that, she spent ten years at the BC Centre for Disease Control and the University of British Columbia's School of Population and Public Health, where she held the Canada Research Chair in Public Health Genomics. Her research focused on the use of genomics as a tool to understand pathogen transmission.

Tamara Munzner is a professor at the University of British Columbia, and holds a PhD from Stanford. She has co-chaired InfoVis and EuroVis, and received the IEEE VGTC Visualization Technical Achievement Award. She has worked on visualization projects in a broad range of application domains from genomics to journalism. Her book Visualization Analysis and Design appeared in 2014, and she co-edits the AK Peters Visualization book series with CRC Press.