Lineage: Visualizing Multivariate Clinical Data in Genealogy Graphs

Carolina Nobre, Nils Gehlenborg, Hilary Coon, and Alexander Lex

Abstract—The majority of diseases that are a significant challenge for public and individual heath are caused by a combination of hereditary and environmental factors. In this paper we introduce Lineage, a novel visual analysis tool designed to support domain experts that study such multifactorial diseases in the context of genealogies. Incorporating familial relationships between cases can provide insights on shared genomic variants that could be implicated in diseases, but also into shared environmental exposures. We introduce a data and task abstraction and argue that the problem of analyzing such diseases based on genealogical, clinical, and genetic data can be mapped to a multivariate graph visualization problem. The main contribution of our design study is a novel visual representation for tree-like, multivariate graphs, which we apply to genealogies and clinical data about the individuals in these families. We introduce data-driven aggregation methods to scale to multiple families . By designing the genealogy graph layout to align with a tabular view , we are able to incorporate extensive, multivariate attributes in the analysis of the genealogy without cluttering the graph. We validate our designs using an illustrative example based on real-world data, and report on feedback from domain experts.

Index Terms—Multivariate networks, biology visualization, genealogies, hereditary genetics, multifactorial diseases.

1 Introduction

Studying ancestry and familial relationships, i.e., genealogies, is both a past-time enjoyed by amateurs and a research area for professionals [41]. It is hence not surprising that there are a wealth of tools to record and visualize genealogies. Yet, most of these tools focus on analyzing family structures for historical purposes and only a few target a clinical use case of analyzing genealogies in the context of complex, hereditary diseases. Geneticists, on the other hand, have long used genealogical graphs to study how a genetic disease manifests itself in families. They use drawing conventions and standardized symbols to show both, the family structure and the phenotype, i.e., the observable characteristics of an individual [6], [7]. These charts can provide insights about the heritability and segregation patterns of genetic diseases. In their current form, however, they are predominantly useful for Mendelian diseases, or genetic diseases caused by a small number of mutations. Complex diseases such as cancer, autism, diabetes, obesity, and psychiatric conditions such as depression or suicide are known to have hereditary components that are regulated by a multitude of genes, each having a modest effect on risk, and also to depend strongly on environmental conditions and chance. When studying these conditions in a population, it is imperative to simultaneously consider genetic similarities, shared characteristics of the phenotype, and environmental conditions. Also, for these polygenic conditions one needs to consider significantly larger populations to reason about hereditary relationships and pursue discovery of genetic risk mutations.

Current medical or historical genealogy [1] visualization tools are ill equipped to help researchers find patterns in these large, highly multivariate graphs of families and their rich medical histories. In this paper, we present a novel genealogy visualization tool that we have developed in collaboration with psychiatrists and geneticists studying the genetic underpinnings and the environmental factors of suicide and autism. We use data from the Utah Population database [2], a uniquely rich resource for population based analysis of hereditary diseases.

We contribute a novel technique to visualize large, tree-like graphs (rooted, directed graphs that have some cycles but are predominantly in tree form) associated with rich numerical, categorical and textual attributes. Our approach leverages the tree-like structure of the graphs to produce a linearized layout which enables the direct association of the nodes with rich attributes in a tightly integrated tabular visualization. We address the issue of scalability by introducing novel forms of degree-of-interest based aggregation that preserve the structure of the graph, and if desired also provide an overview of the attributes of aggregated individuals. While we demonstrate our technique in the context of genealogical data, we argue that it can equally be applied to other multivariate trees or tree-like graphs.

We also contribute a detailed characterization of the domain problems and of the domain data as they are encountered when analyzing large, clinical genealogies and a set of task and data abstractions derived from these characterizations. Finally, we contribute the open source Lineage visualization tool (http://lineage.caleydoapps.org), shown in Figure 1, which implements the technique; and describe multiple design decisions tailored towards genealogical data visualization.

Lineage is in the process of being adopted by our collaborators, and has undergone iterative design refinements. We have also demonstrated it to other research groups working with genealogical and genetic data and have encountered overwhelming enthusiasm. We validate this work in an illustrative usage scenario and through qualitative user feedback from domain experts.

- *Carolina Nobre and Alexander Lex are with the University of Utah. E-mail: {cnobre, alex}@sci.utah.edu*
- *Hilary Coon is with the University of Utah. E-Mail: hilary.coon@utah.edu*
- *Nils Gehlenborg is with Harvard Medical School. E-mail: nils@hms.harvard.edu.*

1. The terms genealogy and pedigree can be used interchangeably in this context. However, for simplicity, we will always use genealogy.

2. https://healthcare.utah.edu/huntsmancancerinstitute/research/updb/

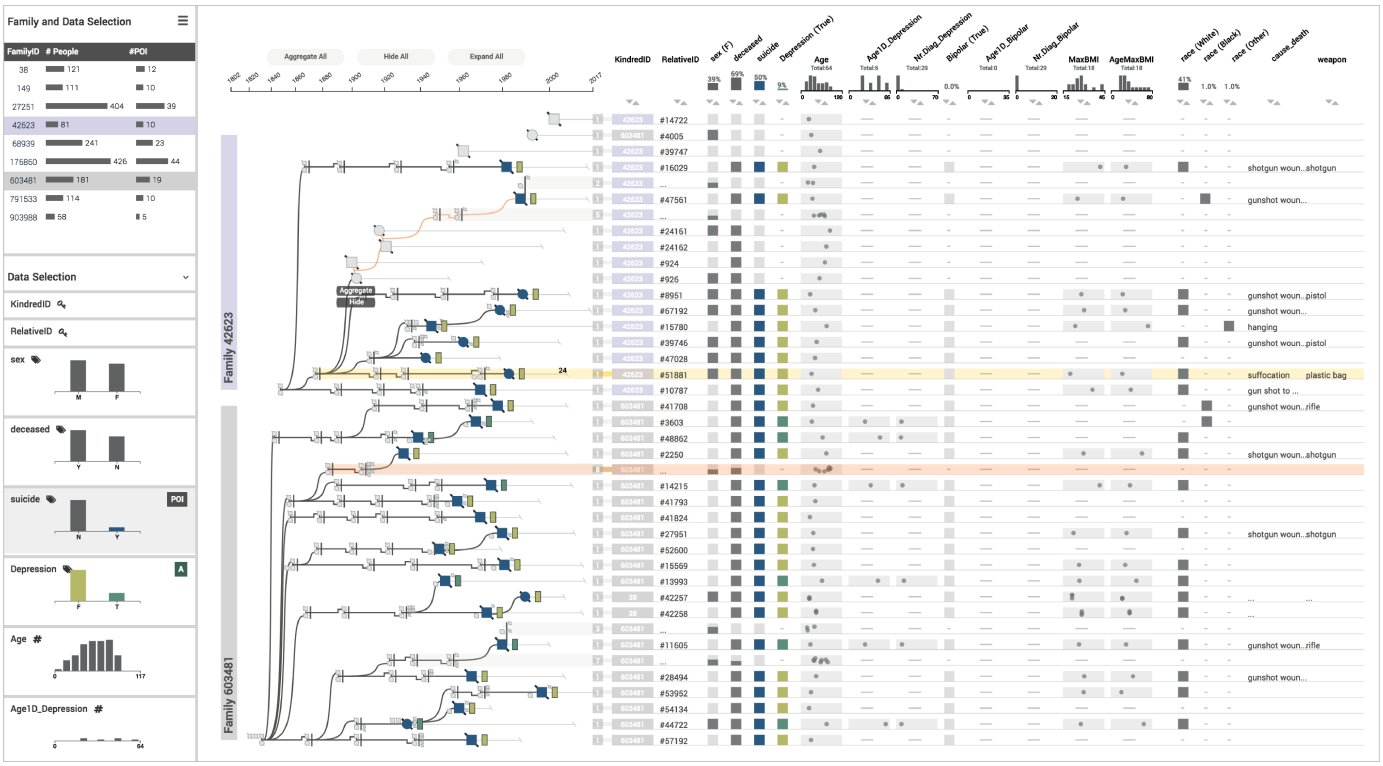


Fig. 1. Lineage visualizing the genealogy of two families with increased numbers of suicides. The genealogy view shows the family relationships in a linear tree layout, where each node corresponds to a row in the associated table. Suicide cases are highlighted in blue, a glyph next to the nodes indicates whether individuals were diagnosed with depression. Some branches are aggregated. The table shows detailed attributes about individuals, or, when branches are aggregated, for groups of individuals.

2 Domain Background and Data

Our collaborators study the genetic underpinnings and the environmental factors influencing psychiatric conditions, such as autism and suicide, using detailed genealogical, clinical, and genetic data. In this paper, we will focus on suicide, yet our methods are easily transferable to other complex, multifactorial conditions and diseases. Suicide is a high impact application, as it is one of the leading causes of life-years lost [59], and the 10th most common cause of death in the United States [43]. Suicide is believed to be caused by a complex combination of risk factors, including environmental stressors, but also genetic vulnerability. Aggregated data across multiple large studies has produced heritability estimates of completed suicide of 45% [39], [47]. Genetic risk factors for suicide are complex and can be classified into multiple subtypes. These subtypes often are characterized by co-occurring psychiatric conditions (comorbidities), and/or combined risk of psychiatric diagnosis. For example, genetic risk for schizophrenia is also associated with risk for suicide [52].

Our collaborators have compiled a unique dataset of suicide cases, including DNA and clinical profiles on 4,017 cases. These cases are linked to the Utah Population Database (UPDB), which provides genealogical data. Genealogies describe the familial relationships of individuals across multiple generations.

Figure 2 shows two genealogies using the standardized drawing conventions [6], [7]. Females are drawn as circles, males as squares. Couples are connected by an edge, children connect to this edge using orthogonally routed links. The vertical position of nodes is given by their generation. A phenotype of interest is marked by a filled-in node.

When studying family relationships, a common approach is to draw family trees considering the ancestry of an individual. Figure 2(a), for example, shows the family of the woman marked in black. The genealogy includes her two siblings and traces her family tree up for two generations to include their parents, uncles and aunts, and grandparents.

In contrast, our collaborators are interested in understanding genetic relationships between individuals afflicted with a condition and hence care about individuals who share genetic variants. They select families for study that have a statistically increased rate of a condition. These family trees are constructed by tracing cases back to a “founder”, as illustrated in Figure 2(b). The underlying hypothesis is that the founder has genetic risk variants which they passed on to their descendants. Within the genealogy, the likelihood of genetic homogeneity is increased, and is more easily detected through the repeated occurrence of the genetic risk variant in the familial cases. Note that this genealogy only contains individuals that are descendants of one founder and their spouse, with the exception of spouses of descendants. Also, the dataset only contains individuals with direct links to a case; i.e., siblings, descendants, and direct ancestors are included, whereas, for example, uncles/aunts and cousins are not.

The dataset our collaborators have compiled contains about 19,000 suicide cases, including 4,017 recent cases with detailed data, backed by family structures made up of 118,000 individuals from 551 families. Suicide is frequently associated with psychiatric comorbidities, i.e., co-occurring chronic conditions, such as depression, bipolar disorder, substance abuse, PTSD or schizophrenia [52]. Also, non-psychiatric conditions such as asthma [26] may play a role in some cases. Environmental factors, such as socioeconomic

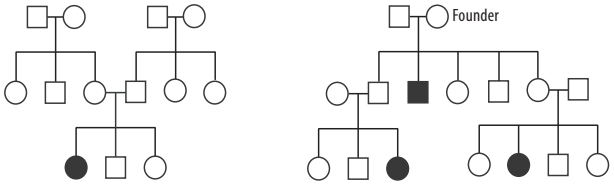


Fig. 2. Two genealogies using standardized symbols focusing on different aspects of the family structure. Females are shown as circles, males as squares. Individuals with a phenotype of interest are filled-in in black. (a) A genealogy showing the family of the female in black, including siblings, parents, uncles/aunts, and grandparents. (b) A genealogy based on a founder, tracing down generations to include the families of individuals with a phenotype of interest (black).

status, pollution, and seasonality are also known to be factors in suicide [2]. To capture this information, our datasets includes demographic variables such as gender, race, age at death, method of death, family demographics (marriage, divorce, number of siblings/children), and place of residence at the time of death. It includes records of other diagnoses captured as codes from the International Classification of Diseases (ICD) systems, the frequency with which these diagnoses were made, and the time of the first diagnosis.

To summarize, we have many graphs, each describing a family, with individuals as nodes and family relationships as edges. Since the graphs are constructed by tracing ancestry to a founder, they are predominantly tree-like, but do include cycles, for example, when two cousins have offspring. In addition, we have attributes on the individuals/nodes in the graphs of various data types, including numerical, categorical, temporal, geographic, and textual data. These attributes are often sparse, as only about 10% of individuals in the dataset have committed suicide, and our detailed records extend to only about 2% (4,017) of individuals across all families. These detailed records capture about 3,000 dimensions that contain demographic information and information about the manner of death, but predominantly contain comorbidities in the form of disease codes, time of the diagnosis, and the frequency of the diagnosis. These dimensions are themselves often sparse, as, among other reasons, a colloquial diagnosis such as “depression” can be recorded using one of about 30 ICD codes.

3 Domain Goals and Tasks

This project is rooted in a collaboration with faculty, clinicians, analysts, and graduate students in the Department of Psychiatry at the University of Utah. In total, six domain experts were involved in the process. We loosely followed the design study methodology by Sedlmair et al. [49]. Our “discover” phase consisted of multiple meetings with individual collaborators and with the whole group as a team, studying the domain literature and the tools they currently use. We also ran a creativity workshop, specifically the “wishful thinking” component described by Goodwin et al. [19], involving all of the collaborators. In the workshop, we asked participants to think about the analysis of suicide data and then discuss in small groups and take notes on post-its about what it is they would like to *know*, *see*, and *do*. This idea generation phase was followed by a phase where the teams had to prioritize their insights, and then finally give the whole team an overview of their key ideas. We recorded the workshop, and transcribed both the audio and the post-its. We then coded the artifacts and three themes emerged: they described details about the *data*, the *factors involved in suicide*, and the *analysis tasks*. The insights on the data and the factors involved in suicide are described in the previous section.

The overarching goal of our collaborators is to gain a better understanding of the determining or associated factors of suicide. They classify these into comorbidities, demographic, genetic, and environmental factors. Specifically, they are interested in identifying and defining detailed phenotypes associated with suicide and the degree to which these phenotypes are familial. By finding people that are similar to each other in a relevant way, our collaborators hope to reason about genetic homogeneity, i.e., shared genetic factors contributing to suicide. They currently rely only on familial structure as a proxy for genetic homogeneity. However, they recognize that this is limited both as too broad — it is possible that they should only consider a part of a family — and as too narrow — people outside a family that have a similar phenotype could also have a similar genotype. Robust and detailed phenotypes are of course also interesting by themselves, as they, for example, can be used as part of a risk assessment in a clinical context.

It is important to note that the contextual knowledge of a researcher is immensely beneficial to the task of classifying a phenotype. For example, a diagnosis of depression is weighted differently if it is diagnosed dozens of times and was first diagnosed early in a patient’s life. Similarly, a suicide case at a young age in a rural community is unlikely to have a detailed medical history. Hence, such a case could potentially be similar to others, even if certain phenotypes are not recorded, if other factors, such as a close familial relationship indicate it.

We identified the following domain tasks as the most important aspects in the workflows of our collaborators:

T1 Select families of interest. The analysts want to select a family either by browsing, or by selecting a specific family based on prior knowledge, or in a data driven way. An example of data-driven selection of families is to find families with high rates of suicide, or to find families with individuals where suicide co-occurs with bipolar disorder.

T2 Analyze individual case. Our collaborators need to investigate the context of a case. For example, a potential genetic component contributing to suicide is judged differently if the person had many psychiatric comorbidities and committed suicide at a young age, compared to a late-life suicide of a person with a terminal disease.

T3 Compare cases. This task encompasses comparing individuals and identifying shared attributes to characterize a potentially meaningful shared phenotype. It also pertains to analyzing how the individuals are related, which can indicate the likelihood of shared genetic traits. Insights on shared environmental factors can be gleaned from both the family structure or the attributes. For example, siblings are likely to be exposed to the same environment in their childhood, whereas cousins might not. Similarly, two people living in the same area are potentially of similar socioeconomic status.

T4 Judge prevalence and clusters of phenotype. The families in our dataset are selected for an increased number of suicides, but these numbers vary greatly between families, and also between branches of a single family. Judging how common a phenotype is in a family or a part of the family is helpful in identifying subsets of interest for further study.

T5 **Compare families.** Once an interesting observation has been made in one family, our collaborators want to be able to investigate whether similar cases also appear in other families. For example, when an association of asthma with suicide is discovered, it is important to know whether this is isolated in one family, or occurs in multiple families and/or individuals.

T6 **Quality control.** While not an analysis task per-se, our collaborators also need to judge the quality of the data and report errors back to the central database. A common data error we have seen, for example, are disconnected components or detached nodes, which are caused by missing information about an individual's mother and/or father.

Most of these domain tasks rely both on studying the topology of the network, i.e., the family relationships, and on investigating the attributes associated with the individuals. For example, the "compare cases" task (**T3**) relies on both the graph and the attributes to, for example, reject an outlier in an otherwise well-defined phenotype within a family, if that outlier is only distantly related to other cases.

4 Related Work

We focus our discussion of previous work on specialized genealogy visualization tools and on multivariate network visualization, as genealogies are highly multivariate graphs. With regards to multivariate network visualization approaches, we also restrict our discussion to explicit layouts (i.e., node link layouts), as implicit layouts (such as SunBursts and treemaps) are ill suited to visualize attributes at all levels of the hierarchy; and matrices are not an ideal choice for genealogies as (a) the nodes are only sparsely connected, hence wasting a lot of space, and (b) matrices are ill suited for path tracing, which is a common task of our collaborators.

4.1 Multivariate Networks

A multivariate network is a graph where the nodes and/or the edges are associated with potentially rich attributes [27]. Many graph visualization techniques are optimized for either topology or attribute based tasks [56], yet in many applications topology and attributes have to be judged in concert [45]. When analyzing genealogies, for example, our collaborators want to understand how two people with a similar phenotype are related, requiring them to first identify the phenotypes using the attributes, and then judge their relatedness using the topology of the genealogy.

Partl et al. [45] classify four basic approaches to visualize multivariate networks for explicit graph layouts: (1) on-node mapping, i.e., visualizing the attributes by changing a visual channel of the node mark or by embedding a small visualization in the node; (2) small multiples, i.e., showing the same graph multiple times and visualizing a different attribute on top of each of the small networks, (3) separate, linked views for the graph and the attributes, and (4) adapting the graph layout to better fit the needs of attribute visualization.

These approaches have different strengths and weaknesses with respect to the tasks they enable. Lee et al. [32] distinguish, among others, topology based tasks, i.e., tasks that are related to the network's connectivity, and attribute based tasks, i.e., tasks that are related to the attributes associated to the nodes.

While **on-node mapping** excels at simultaneously supporting topology and attribute based tasks, it does so only for very limited numbers of attributes, as the node size limits how many attributes can be encoded. Also, on-node visualizations are typically not aligned and have distractors between them, which makes accurate comparison difficult [11]. Gehlenborg et al. [17] review multiple systems that use on-node mapping for biological networks. An example for slightly more complex visualizations embedded on nodes is the Network Lens [25]. The work by van den Elzen and van Wijk [56] is a special case of an on-node mapping approach: instead of mapping data directly onto nodes in the networks, they aggregate nodes into super nodes, show the relationships between the supernodes, and visualize the attributes of these nodes in small, embedded visualizations.

Small multiples are also commonly used to visualize attributes on top of graphs. Barsky et al. [5] and Lex et al. [34], for example, use small multiples to show gene expression data on top of biological networks. Using small multiples for multivariate networks, however, has the disadvantage that the individual networks have to be rendered in less space, limiting their readability and/or the size of the graph for which they are useful for.

Separate, linked views excel at visualizing the attributes and the graph individually, but do not support the integration of both well. Systems that use this approach [33], [50] rely on linking and brushing to associate a node with the representation of its attribute, which requires interaction to reveal relationships between topology and attributes.

The fourth approach to multivariate graph visualization is to adapt the layout of the network so that the nodes can be easily associated with an effective attribute visualization. This is taken to the extreme in GraphDice [8], where nodes are positioned in a series of scatterplots purely based on attribute values. Gentler approaches are various linearization strategies, where graphs are laid out such that associated attributes can be visualized in efficient tabular layouts, overcoming the drawbacks of completely separated linked views. Typically, trade-offs between optimizing for the readability of the topology and the linear layout have to be made. Meyer et al. [40] manually linearize a complete network and render attributes next to the linear layout. While this is an efficient approach, the complexity of the networks for which this is feasible is limited, and topological structures can be hard to see. Partl et al. [45] use interaction to extract paths from a network, linearize these paths, and associate the nodes in the paths with rows in a tabular visualization. This, however, requires interaction and works only for selected subsets of the graph. The recently published Pathfinder system [44] uses path queries on networks and presents the resulting paths in a linear, ranked list, juxtaposed with rich attribute data. This approach, however, is only sensible for tasks related to paths.

Our work falls into the category of adapting the layout by linearization. We leverage the fact that the genealogical graphs our collaborators are interested in are tree-like and linearize the positioning of the nodes in the tree. We use this tree to juxtapose scalable and perceptually efficient visualizations of the attributes.

There are many approaches that do this for the leaves in a tree, such as juxtaposing dendrograms with a heatmap [13] or visualizing the attributes of the leaves of an evolutionary tree [30], [31]. While these approaches allow for multi-attribute visualization of the leaves of large trees, we are not aware of prior tree linearization approaches that also visualize attributes for intermediate nodes, either in aggregated form or for each node individually.

Also related to our approach are tree tables, as they can be found in file browsers, where the tree represents the structure of folder and files, and attributes such as the file size are shown. Tree tables

generally do not provide aggregation functionality — a branch can either be collapsed or expanded, but cannot be aggregated.

Finally, tree visualization techniques such as tree maps [24] or sun burst [53] are well suited to visualize one or two attributes of nodes in trees (using size and color), but do not scale to more attributes.

4.2 Genealogy Visualization

Genealogical charts, as shown in Figure 2, are widely used in genetic counseling and the literature on genetic diseases. While they are well suited to visualize a single phenotype of interest, they are not suitable to map a complex phenotype to the node. Our collaborators currently use Progeny [48], a commercial genealogy drawing tool that closely follows the standard for visualizing genealogies [6], [7] (see the supplementary material for an example figure created with Progeny). While Progeny is well suited to draw these standard genealogies for use in presentations, it is ill-suited for exploratory tasks, mainly because of its inability to efficiently encode attributes in the graph.

Interactive genealogy visualization tools that are designed to analyze disease clusters and to see disease propagation within families include PedVizApi [14], CraneFoot [42], Haploview [4], PediMap [57], and HaploPainter [54]. HaploPainter [54] visualizes genealogies and genetic recombination events below the individuals' nodes. While it shares the approach of showing metadata as rows associated with nodes with Lineage, it does not take a linearization approach to make values of different generations easy to compare, it does not aggregate the network, and it does not visualize different types of attributes. McGuffin and Balakrishnan [38], describe layout algorithms for complicated genealogical trees, but also introduce aggregation methods for sub-trees, which we adopt.

Among tools that don't use the standard genealogical drawing conventions are Fan Charts [12], which uses the SunBurst technique to visualize genealogical trees, and the work by Mazeikla et al. [37], which employs a force directed layout that considers similar phenotypes as additional attracting forces. Tuttle et al. [55] use an H-tree layout for scalable genealogy visualization, with the founder at the center and successive generations radiating out based on a fractal pattern. Ball [3] employs the idea to not represent generations as discrete units but use time to position the nodes, and also to draw a person's life span. Kim et al. introduced TimeNets [29] a technique also focused on the temporal aspects of a genealogy. While it is well suited to observe temporal changes in relationships between individuals, relationships between generations are harder to trace.

GenealogyVis [35] is a recent tool for visualizing genealogies to study historic data. While it visualizes multivariate attributes, it addresses different needs — those of historians — and uses different approaches. Unlike in Lineage, attributes of individuals are not shown, rather the focus is on demographic trends in (parts of) the network. Supplementary views, such as scatterplots and maps allow historians to study migration patterns, for example.

Genealogy visualization tools for animal genealogies face a different set of challenges compared to those for human genealogies, as the number of descendants sired by individual animals can be large, and complex interbreeding is common. Consequently, tree-based approaches are not well suited for these genealogies. Examples include CoVE [10], and VIPER [46]. VIPER introduces a sandwich view, that CoVE also adopts. The sandwich view scales well to many descendants of an individual, but only explicitly encodes the relationships between parents and their children. More distant relationships can be revealed through highlighting. Helium [51] is a visualization techniques for plant genealogies, which commonly have complex crossing. It uses color coding and scaling of nodes to encode up to two attributes.

GeneaQuilts [9] is a matrix based technique where each row constitutes a person and each column a nuclear family. In early stages of our design process we considered using a GeneaQuilt instead of our node link design, since GeneaQuilts produces a linearization of the graph that would be suitable to associate attributes. We ultimately decided against it because our design for aggregation is more suitable for node-link diagrams.

A different approach to analyzing relatedness is to calculate "kinship coefficients" between individuals, i.e., to calculate path-based metrics for relatedness and visualize them in a matrix [28]. While this is scalable, it is not suitable for reasoning about all patterns of inheritance.

A related tool that is concerned with visualizing phenotypes of patient cohorts is PhenoStacks by Glueck et al. [18]. PhenoStacks uses a similar tabular approach as we do for our table.

5 Visualizing a Multivariate Graph

The tasks our collaborators need to address rely heavily on both the familial information contained in the genealogy graph, i.e., the topology, and the myriad of attributes associated with individuals (see Section 3). Of the strategies for linearization introduced in Section 4.1, only the linearization method enables an integrated analysis of topology and attribute at the scale of attributes we are interested in. However, none of the described linearization methods are suitable for the data and tasks of our collaborators. Here, we introduce a linearization method for tree-like graphs. We define tree-like graphs as rooted, directed graphs that contain cycles. The purpose of the linearization is to associated the nodes with rows in a table visualization.

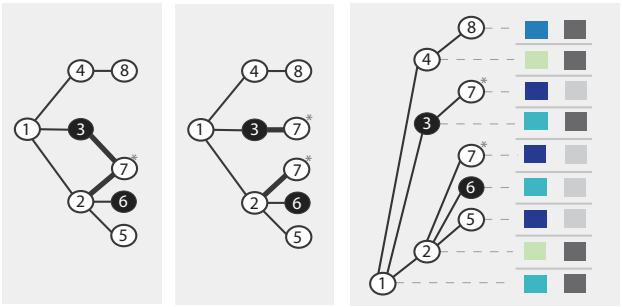
A consequence of the linearization strategy is that the layout is not as compact as other, common layouts are. To address this issue, we also introduce degree-of-interest based aggregation strategies that integrate seamlessly with the linearized graph.

We illustrate this concept here using general, tree-like graphs, for now ignoring specific properties of genealogies. We later show in Section 6 that this approach extends to genealogies (where each person has two roots — their parents) with minor modifications, and also elaborate on design decisions we made that are specific to our data and application area.

5.1 Linearization Approach

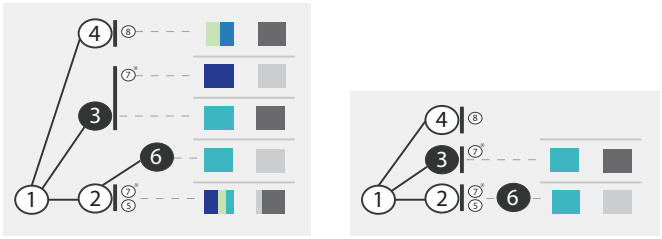
5.1.1 De-Cycling

In a first step, we remove cycles from the directed graph, transforming it into a tree, by duplicating the node that completes a cycle, similar to the approach by Mäkinen et al. [42]. If the duplicated node has children, we attach all children to one instance, while the other instance remains a leaf. Figure 3(a) shows a tree-like graph with one cycle, Figure 3(b) shows the resulting tree, where node 7 is duplicated. While this duplication strategy works for general directed graphs, it is most useful for directed graphs with a defined root and few cycles, as in these cases most of the topology is retained, and the number of additional nodes are negligible with respect to scalability.



(a) (b) (c)

Fig. 3. De-cycling and linearization. (a) A directed, rooted graph with one cycle ending in node 7. (b) We remove the cycle by duplicating the last node in the cycle (node 7). (c) The tree is linearized so that each node is assigned a distinct row. Leaves are rendered above their parents. This row-based, linear layout enables an unambiguous, position-based association with a table visualizing attributes.



(a) (b)

Fig. 4. Aggregation approaches demonstrated using the tree in Figure 3(c). A black fill indicates a node-of interest. (a) Attribute-preserving aggregation. Each node of interest (shown in black) is in a separate row. Branches without nodes of interest are aggregated into one row, yet all attributes are preserved in the aggregate representations in the table. Notice how the two children of node 2 that are not affected are shown using an implicit encoding, which we refer to as a "kid grid". (b) Attribute-hiding aggregation. The branches leading to nodes of interest are hidden behind them. Only nodes of interest and branches with no nodes of interest have a row of their own. Only the nodes of interest are represented in the table.

5.1.2 Linearization

In most tree layouts [1], associating the nodes with rows in a table by position is impossible. The tree in Figure 3(b) is compact, yet would require, for example, curved links to associate the nodes with a table row. To make this association between nodes and rows of a table intuitive, we use a linearization strategy that assigns every node a distinct vertical position (i.e., a "row"). The position of the node alone thus unambiguously associates the node with a row in a table (see Figure 3(c)). Note that while we assume a left-to-right tree layout here, a top-to-bottom layout would work equally well for associating a tree with table columns.

Linearized tree layouts are based on tree traversal strategies. While various strategies, such as breath-first (level-order), or in-order depth-first-search are possible, we found that pre-order depth-first search works well for our purposes, as it results in a crossing-free layout and keeps leaves in subsequent rows.

Following the in-order strategy, we recursively place the descendants of a given node directly above them. Note that a top-down strategy would be equally possible. We assume that an order of leaves can be defined, e.g., based on the attributes. If not, using a random order is possible.

Figure 3(c) illustrates the results of this algorithm when applied to the tree in Figure 3(b) and also shows how to easily associate a table with the tree. Note that the duplicate node also is duplicated in the associated table.

5.2 Aggregation

While linearizing the tree allows for a direct, position-based association of the nodes and their attributes, the resulting layout uses more space than a compact layout. However, due to their hierarchical structure, trees are well suited for aggregation. Degree of interest (DOI) functions [15] have been widely applied to trees. In our design, we use the generalized idea of degree of interest functions by Furnas [15], [16].

We let analysts define a degree of interest function based on the attributes of the nodes, which we call the **phenotype of interest (POI)**. Nodes that have the POI are referred to as **nodes of interest**. In contrast to the original formulation of a degree of interest, our POI function is binary (i.e., a node is either of interest or not) and does not consider a distance to a selected node. An example for a POI is "committed suicide", which would make all nodes representing individuals that committed suicide to be considered of interest, or "has a maximum BMI of higher than 30", which would consider all obese individuals to be of interest. Naturally, POIs that are compound of multiple attributes (high BMI and suicide) are possible.

Based on this degree of interest function, we introduce two different approaches to aggregation that vary in how they trade-off compactness and preserving the attributes of the nodes: (1) attribute-preserving aggregation, and (2) attribute-hiding aggregation. These aggregation approaches can, of course, not only be applied to the whole tree, but to selected sub-trees, and can also be combined.

5.2.1 Attribute-Preserving Aggregation

Here we introduce an aggregation strategy for linearized layouts that preserves both the structure of the tree and the attributes of all the nodes. Nodes of interest are assigned a row of their own, while other nodes are aggregated into a single row. Figure 4(a) shows an example of this strategy applied to the tree shown in Figure 3(c). This emphasizes the nodes of interest, while preserving both, the structure of the graph and the attributes of the other nodes.

Our algorithm recursively follows a (sub)tree down a branch by assigning a new row to each inner branch. Inner branches are branches that don't end in a leaf after the first edge, i.e., an edge that directly connects to a leaf is not an inner branch. If no node of interest is encountered, it continues to the leaves, placing all nodes of the branch in the same row. Multiple leaves that are not of interest are placed in a "kid grid", an implicit ISOTYPE encoding of the leaves as small nodes to the right of their parent. ISOTYPE uses simple pictographic elements to convey information [22]. We chose this approach for the representation of kid grids over alternative designs such as a numeric labels or bar charts since it is scalable and the relationship between the small ISOTYPE nodes and the large nodes in the tree is obvious. An example is visible in the bottom branch in Figure 4(a), where nodes 1, 2, 5, and 7 are on the same row, and the leaves (5, 7) are in a kid grid. If a node of interest is encountered, we distinguish two cases. If the node of interest has children that are leaves and that are not nodes of interest themselves, they are added to a kid grid, which is placed in the next row (see node of interest 3 and its descendant

(node 7) that is placed in a kid grid in Figure 4(a)). If the node has children that are inner nodes, the algorithm is applied recursively.

The result of this algorithm is a layout that has N rows, where N is the sum of:

- the number of nodes of interest,
- the number of inner branches that do not end in nodes of interest (case for node 4 in Figure 4(a)),
- the number of nodes of interest that have children that are leaves (case for the child of node 3 in Figure 4(a)).

The result in the associated table visualization is that each node of interest has a separate row, and the aggregated branches are represented in aggregated rows. In practice, we use visual encodings for aggregates and individual rows that can faithfully represent the data but are also comparable. For details on the table design, see Section 6.2.

5.2.2 *Attribute-Hiding Aggregation*

This form of aggregation also preserves the complete structure of the tree, but does not preserve attributes of nodes that are not of interest. The results, illustrated in Figure 4(b), is a scalable approach that can be used to address tasks that are only concerned with the attributes of the nodes of interest and their connectivity, but not with the attributes of the other nodes.

The main difference compared to the attribute-preserving aggregation is that nodes of interest are not assigned to a new row when they are encountered. The algorithm again recursively follows a (sub)tree down a branch by assigning a new row to each inner branch. If no nodes of interest are encountered while traversing the branch, the leaves are placed in a kid grid. If a node of interest is encountered, the next step depends on whether it has children that are inner nodes or not. For the node's children that are leaves, a kid grid is used, but no new row is started. For all other branches, the algorithm is applied recursively.

The resulting layout has M rows, where M is the sum of:

- the number of inner branches,
- the number of nodes of interest that have at least one child that is an inner node.

Here, only nodes of interest and inner branches that do not end in a node of interest are assigned their own row. For consistency, we do not represent branches that do not end in a node of interest in the table.

6 Lineage Design

Here we describe the design decisions that are specific to the use case of visualizing genealogies and that we realized in the Lineage prototype. To address the tasks of our collaborators, Lineage provides four views, shown in Figure 1: the genealogy graph view and the closely synchronized table view; a data selection view, which can be used to select which attributes to display; and a family selection view, which allows analysts to switch between or select multiple families.

6.1 Genealogy Graph

An important difference between genealogical trees and general trees is that nodes have not one but two parents. To address this, we introduce the concept of a couple, indicated by a line connecting the partners (see Figure 5(a)). As is common in genealogical graph layouts, the children of a relationship then connect to the line representing the couple instead of directly to the parents.

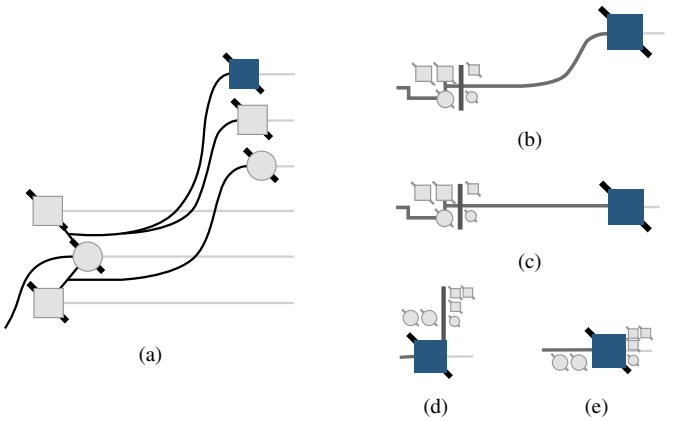


Fig. 5. Different aggregation cases. (a-c) A family where one woman has children with two men. One of the children committed suicide. (a) No aggregation: every person is in their own row. (b) Attribute aggregation: the suicide case is in its own row; the rest of the family is aggregated. Notice the family grid with two male and one female parents, and one daughter and one son. The second son is not in the kid grid because he is a node of interest. (c) Attribute hiding: the family is hidden behind the suicide case, only the attributes of the suicide case will be shown in the table. (d-e) A different family, where the node of interest has children, leading to special cases. (d) Attribute aggregation: the spouses and children are moved to their own row, the line connecting spouses spans two rows. (e) Attribute hiding: the spouses are placed to the left of the suicide case, the children to the right.

We also adopt some of the conventions for drawing genealogical graphs: males are drawn as rectangles, females as circles. Deceased individuals are crossed out. See for example, Figure 7(b), where the top-most node represents a female who is alive and has the POI, while the other nodes with the POI are deceased. Nodes that have the phenotype of interest are filled-in.

As discussed in the previous section, the phenotype of interest can be defined dynamically, either based on (combinations of) categorical values, or by brushing a range of a numerical variable. Figure 7 shows the effect of two different POI functions on the same subtree.

The modifications to the layout algorithm to accommodate couples are minor: couples are always placed in consecutive rows, to avoid long, vertical parent edges. When one of the spouses has offspring with multiple partners, we place all partners in consecutive rows. In case of two partners, we place the person with multiple relationships in the center to avoid edge crossings. Figure 5(a), for example, shows a woman who had children with two different partners. For more than two spouses, however, or spouses who had children with different partners in alternating order, edge crossings are often unavoidable. Similar to Mäkinen et al. [42], we use arrows to indicate that a node is duplicated and to point towards the duplicate. To resolve any ambiguities, we draw an edge connecting the duplicates when hovering over the arrow (see Figure 6).

In contrast to traditional genealogical graphs, we do not lay the nodes out by generation, but use the birth year to position the nodes horizontally [3], as shown in Figure 1. This avoids ambiguities about the birth order and encodes a vital attribute directly in the graph. We also use curved splines instead of the traditional orthogonal edge routing, because continuous edges are easier to follow [58].

6.1.1 *Aggregation Layouts*

With respect to aggregation, the algorithm is only extended by first looking for spouses before descending into a subtree. If both are

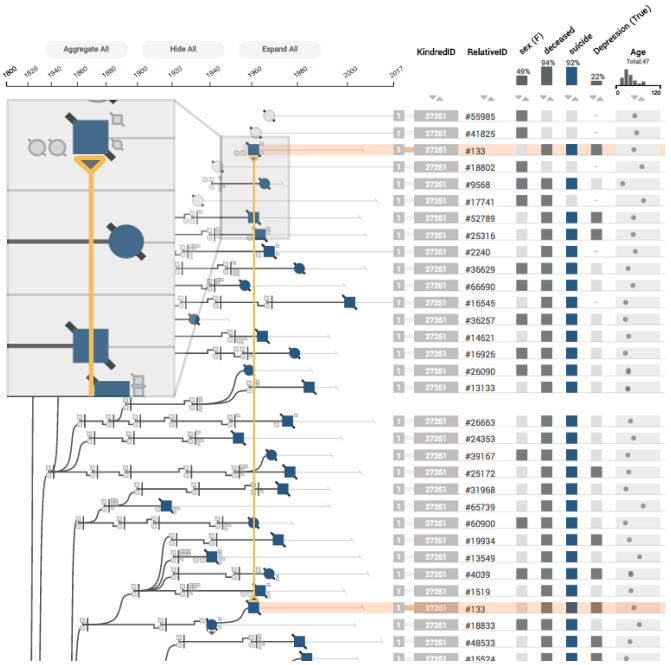


Fig. 6. Visual encoding of nodes that were duplicated in the process of removing cycles from the graph. The arrow glyph, which is shown at all times, indicates both the presence of a duplicate as well as its direction in the graph. Hovering over the arrow draws a line connecting it to its duplicate and highlights the corresponding rows in the table.

nodes of interest, each spouse is assigned their own row.

We previously introduced the concept of kid grids, an ISOTYPE visualization, for aggregated nodes. Indicating hidden nodes using a glyph has been done before for graph layouts, most notably by McGuffin and Balakrishnan [38], who use dots to indicate children in genealogical graphs. Our layout for aggregated genealogies, however, goes beyond a basic indication of existing nodes as they encode both, topological information and attributes. First, we extend the notion of a kid grid that encodes children, to a family grid that encodes all members of a family. Figure 5 shows multiple examples. A family is separated by a vertical line into parents and children. This vertical line represents the line used to connect spouses in unaggregated mode. Parents are placed on the left of the line. In addition to the node shape, we also redundantly encode sex by position, placing the nodes representing males on top and the nodes representing females below. This redundant encoding is useful, as aggregated nodes are rendered significantly smaller, and hence, can be harder to read. In families with multiple partners, we place all partners in the same family grid, so that, for example, a family where a woman who has children with three partners is represented by three squares on top, and one circle at the bottom.

Note that aggregation results in some level of information loss. For families where individuals have offspring with multiple partners, the exact association between children and parents is lost. Also, the attributes for all aggregated nodes in a row are displayed together in the table, and hence the association between a particular individual and a particular attribute is lost; instead the distribution of values in that aggregate is emphasized. When hiding is used the attributes are removed entirely from the table. We found that neither is a problem since it aligns with the analysis tasks outlined earlier: analysts at first are often interested in nodes with the POI. When they want to consider other nodes in detail, they can de-aggregate on demand.

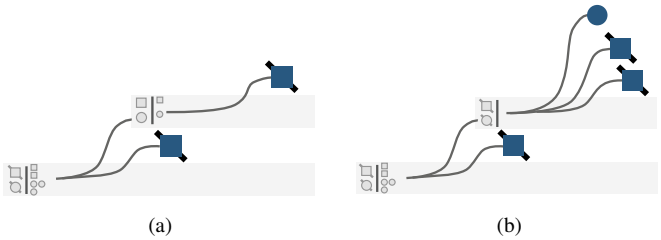


Fig. 7. Different POI functions applied to the same, aggregated subtree. (a) Suicide, as a categorical POI. (b) Age < 40 as a numerical, thresholded POI.

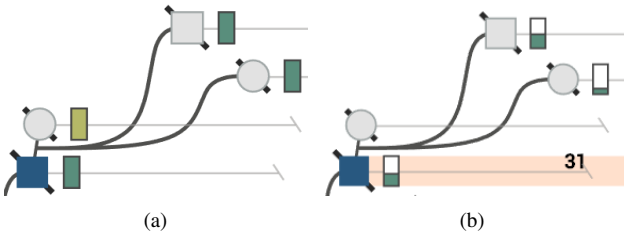


Fig. 8. Attributes encoded directly in the graph. Age lines visualize the life-span of individuals. Age lines for people that are alive continue until the present. Age lines of deceased individuals are terminated at their year of death. We can see that the individual represented by the node of interest died at age 31, and his spouse died only shortly thereafter. Selected attributes can be visualized next to the nodes in glyphs (green rectangles). (a) The categorical variable bipolar disorder is encoded by a dark-green color. (b) The numerical variable number of bipolar diagnoses is encoded as a bar chart.

It is important to note that we break with the convention of placing nodes based on their birth-year for aggregated families. Instead, we place the whole family based on the birth year of the parent with a blood relationships to the ancestors.

6.1.2 Encoding Attributes in the Graph

While we address the problem of encoding multiple attributes for nodes using our linearization approach, direct, on-node encoding of a very small number of attributes provides the best bridge between attribute-based and topology-based tasks. We already discussed how sex (shape), deceased/alive (crossed out), birth year (horizontal position), and POI (fill) is encoded directly in the graph. To enable our collaborators to view an additional variable in the graph, we introduce a glyph, rendered to the right of the nodes, as shown in Figure 8. In case the attribute is categorical we color-code the glyph; for numerical attributes we show a small bar. In both cases, the color coding is also used in the table to highlight the relationship (see the matching colors for biploar disorder in Figures 8(a) and 9). When data is not available for a node, no glyph is shown.

Finally, we also encode the age of individuals directly in the graph by drawing a line from the node, which is placed at the year of birth, to the year of death, or to the current year (see Figure 8). These age lines conveniently encode an important variable in the existing coordinate system. We found that the age lines also help to perceptually connect the nodes to the rows in the table. As we don't draw age lines for aggregates, we found it necessary to indicate the connection to the table using a light-gray background, as can be seen, for example, in Figure 7.

6.2 Table Visualization

The attribute table is designed to visualize both rows representing individuals and aggregates representing multiple individuals in the

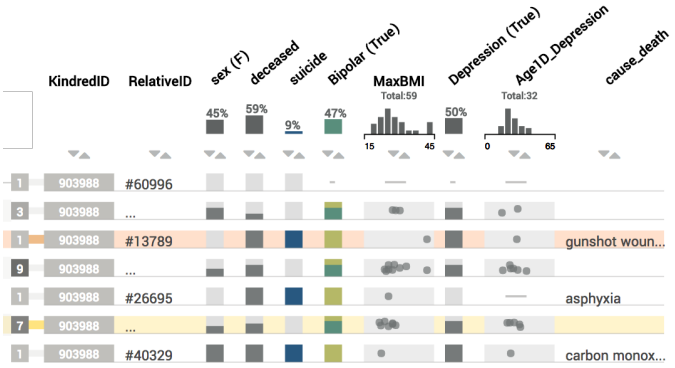


Fig. 9. The table view. The first column encodes how many individuals are aggregated in that row. Binary categories are represented as present/absent (e.g., sex). Aggregates of categorical variables show the proportions of the variable in stacked bars. Numerical values are encoded using a dot plot, which is also used for aggregates.

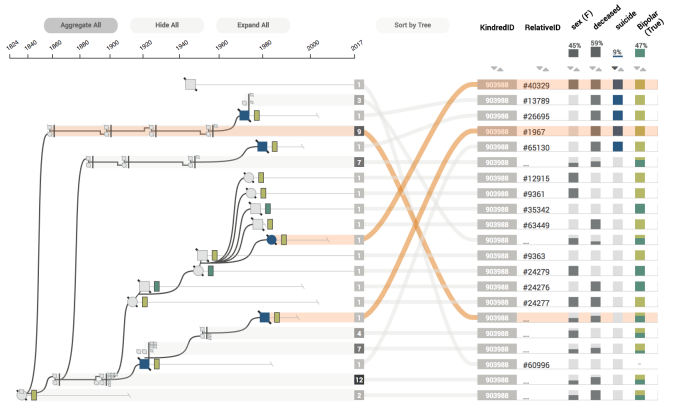


Fig. 10. The table is sorted by suicide, which causes the rows in the table to be in a different order than the rows in the graph. The association between the two is retained by the curves connecting them.

same space. As shown in Figure 9, we use dot plots to encode numerical data. Combined with transparency and jitter, dot plots can also be used to encode aggregate rows.

For categorical values, we distinguish between binary categories, such as deceased or alive, and multi-valued categories, such as race. We encode binary categories in a single column, as can be seen for "sex (f)", where a dark cell corresponds to true and a light cell corresponds to false. For multi-valued categories, we use one column for each value instead of, e.g., using color. An example of a multi-valued attribute is the race category in Figure 1 which contains three possible values: white, black, and other. Hence, we use the strongest visual variable — position — to encode the data. We represent aggregates of binary or categorical values as stacked bars, which are scaled according to the number of individuals in a category.

Text labels and IDs do not have adequate visual representations for groups of elements, so we display an ellipsis (...) for aggregates. Missing values, which are very common in our datasets, are rendered as a dash and to distinguish them from zero or false values.

We also provide a column that shows how many people are in a given row. Here, we print the exact number and use a redundant encoding by value, where darker cells correspond to more individuals in a row.

As we avoid color to encode data, we can employ it to highlight elements of interest, such as to highlight selected rows and to indicate the column that encodes the user-selected phenotype of interest and the primary attribute. In Figure 9, the selected attribute (bipolar) and the POI (suicide) are rendered in color.

The attributes visualized in the table can be chosen using the data panel shown in Figure 1. Numerical and categorical attributes listed in the panel are accompanied by a histogram, so that analysts can judge what to expect before adding an attribute to the table.

In combination with the graph, these features enable analysts to address the tasks related to analyzing individuals (**T2**), and comparing cases (**T3**).

Finally, we also allow analysts to sort the table based on any column. This enables analysts to easily identify clusters of similar items (**T4**). However, sorting by attribute removes the close association with the graph. To partially remedy this, we draw slope charts, similar to what is used in LineUp [21], to relate the rows of the table to the rows of the graph. These connection lines work well for a small number of rows, but often result in significant crossings when dealing with many rows. In that case, interactive highlighting helps to trace the lines. Figure 10 shows an example of a partially aggregated graph sorted by suicide.

6.3 Viewing Multiple Families

One important aspect of our collaborators' workflow is to compare multiple families (**T5**). A requirement for comparison of families is the ability to select families **T1**, which is enabled by the family selection view, shown at the top left in Figure 1. The family selection view shows statistics about the family, such as its size, and the number of people with the currently selected POI. It also allows analysts to sort by these attributes to, for example, quickly identify how many individuals were diagnosed as bipolar before committing suicide, across the whole dataset, and thus enables them to quickly scan for enriched phenotypes. Multiple families are seamlessly integrated into the graph and table views (see Figure 11). To visually separate the families, a bar showing the extent of the family and containing the family label and a unique color is show to the left of each family.

7 Implementation and Pre-Processing

Lineage is open source and is implemented in TypeScript as a Caleydo Phovea client/server application [20] and uses D3 for rendering. The server component is based on Flask and is provided as a Docker container for easy deployment. A prototype of Lineage is available at http://lineage.caleydoapp.org, the source code is available at https://github.com/caleydo/lineage.

The prototype made available publicly uses two different datasets: a synthetic dataset designed to showcase features and to highlight edge cases; and ten selected and anonymized families from the suicide study based on data from the Utah Population Database. The anonymization method and the selection of families was approved by the Utah Resource for Genetic and Epidemiologic Research (RGE). The anonymization process involves randomizing the sex of individuals, randomizing the birth and death years, and randomly deleting individuals, in addition to omitting attributes that could be identifiable. Hence, we do not recommend to make clinical inferences based on the data provided.

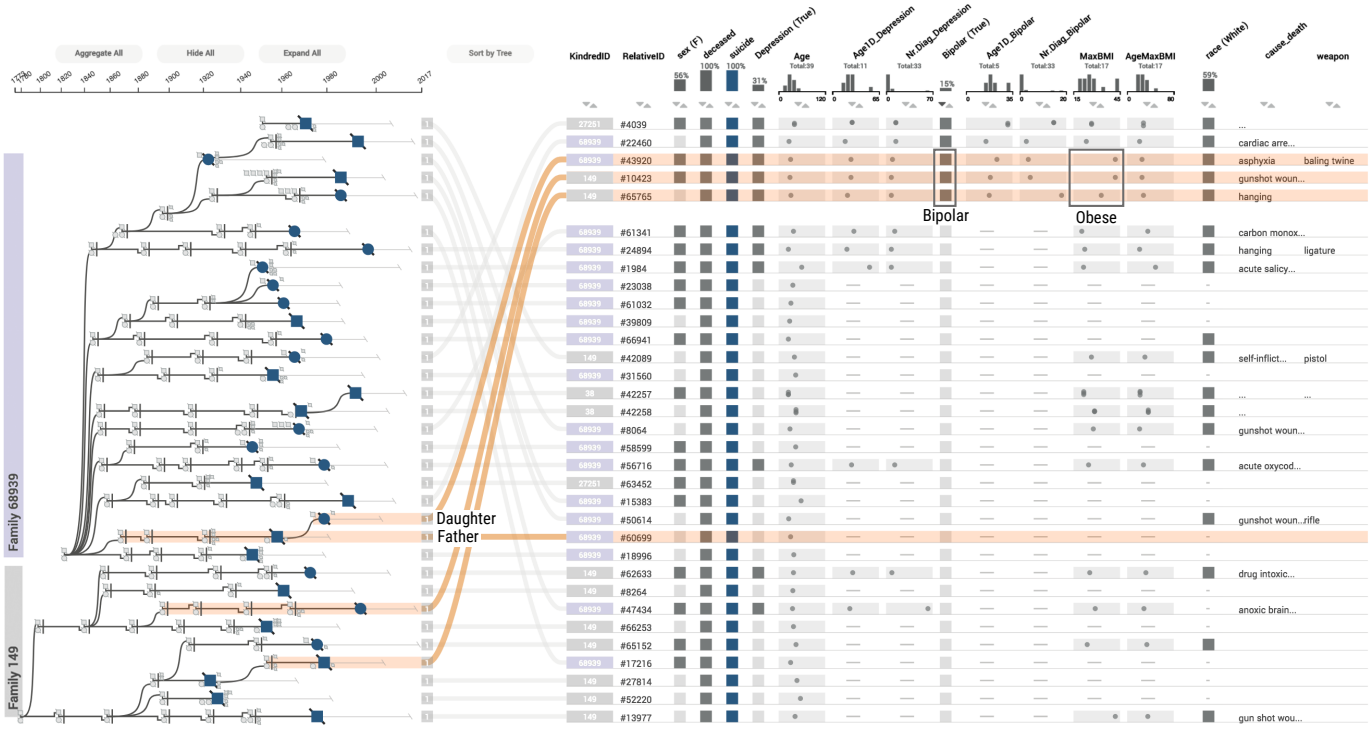


Fig. 11. Three suicide cases, from two separate families, that are both bipolar and obese. The family structure reveals that the female case had a father who also committed suicide at a young age, but for whom no detailed data is available. This suggests an important genetic component.

8 Usage Scenario

In a typical usage scenario, analysts want to use Lineage to identify interesting phenotypes that are familially correlated with suicide. Such discovery will (1) enable selection of familial cases for genotyping and/or sequencing efforts and subsequent genetic risk discovery analyses, or (2) post-hoc description of cases already subjected to genetic analyses who show significant familial sharing of genomic regions, allowing for potential prioritization of genes within these regions based on genetic associations with co-occurring phenotypes. In this usage scenario, an analyst starts with a family and browses for cases that are bipolar (**T1**). She discovers two cases in a family with 10 suicides in total (**T4**). Upon closer investigation (**T3**), she finds that both cases are male, committed suicide around the age of 25, were also diagnosed with depression and were both obese. Looking closer at the number of diagnoses of bipolar disorder, she finds that one of the two was diagnosed multiple times, indicating that the diseases was a significant burden. Exploring the family structure, she sees that they are related to each other, but not too closely, which is promising for a familial genetic analysis, as they might share a relevant gene variant, but will not have the large amount of genomic sharing of close relatives. By checking the *LabID*, she ensures that she has genetic material in the study for these cases.

Next, she adds other families with bipolar suicide cases (**T5**). One other family has three bipolar cases, but one of these cases seems especially interesting, because she is also obese, has also been diagnosed with depression and committed suicide at a young age. Looking at the family structure of this case (**T2**), she realizes that the father of the case has also committed suicide many years before, at about the same age as the case of interest (see Figure 11). She speculates that, while no detailed information is available for the father, he likely had the same comorbidities and hypothesizes that a genetic factor might be contributing to these cases. Hence, she chooses to select these three cases to analyze their shared genomic regions, to see whether they share common variants that may alter gene function or regulation and lead to risk, with variant selection informed by the knowledge of the case phenotypes.

9 Analyst Feedback

We ran an informal feedback session with four analysts (two faculty members, one research scientist, and one PhD student). With the exception of one of the faculty members, who is also a co-author, the participants did not contribute to the design and development of Lineage, except for the requirement analysis as described in Section 3. After an introduction to Lineage, participants were asked to use the tool with their own data and articulate their thought process and observations according to the think aloud protocol. This was followed by a brief interview. The sessions took between 90 minutes and two hours.

The feedback we received was overwhelmingly positive, including statements such as "*This is going to completely change how we do things*". One analyst noted that Lineage will allow them to properly use visualization for exploration of genealogies for the first time, because their current tools are not suitable for discovery, as they can only effectively visualize one or two attributes at the same time, and the tools are essentially static and difficult to use.

The analysts consistently noted that the integration of attributes and family structure is critical for them to make decisions about where to follow up with subsequent analysis, making comments such as "*I think it's really helpful to see the attributes next to the graph. It really helps to pinpoint the important cases*".

We also observed that the analysts largely followed a pattern: after they selected a family, they quickly aggregated it to get an overview of the data. Then they continued to drill down, using selective expansion of branches. After a while, they turned to the table and began to sort by attributes, to look for individuals with interesting phenotypes. Multiple analysts commented that a hierarchical sorting approach, which is currently not supported,

would be useful, so that they can easily find people with a complex phenotype. When using the sorting, they frequently used the row-highlight feature to trace individuals in the table back to the genealogy. They also commented that highlighting the paths to shared ancestors would be very helpful in these cases.

We asked the analysts about their opinions on attribute-preserving aggregation and how it compared to attribute-hiding aggregation. They commented that attribute-preserving aggregation is not particularly useful for their suicide dataset due to the sparse attributes of the non-affected individuals, but that they can imagine it to be very useful when applied to their autism dataset that contains more data on family members. One analyst gave the example that he would be interested to see autism spectrum scores aggregated for a whole family.

The analysts also stated that they believe that Lineage graphs are appropriate for presentations in publications and presentations, as the visual encodings are easy to explain. They asked for some features in support of presentations, such as the ability to hide irrelevant branches and/or nodes of the graph, or to re-define the founder to clean up the genealogy. Finally, we also asked for other features that they wished the tool had. The answers to that were mostly regarding data, i.e., to load more data into the tool and to provide export capabilities for a subsequent statistical analysis.

10 Discussion

While details of our design study and our implementation, such as how we display parents and family grids, are specific to genealogies, we argue that our linearization and attribute-driven aggregation approach can be applied broadly when analyzing multivariate trees or tree-like graphs, such as phylogenies or file directories. We also argue that our strategy of combining explicit node-link layouts, with the implicit layout of the family grids is transferable to other application scenarios.

Our described linearization approach makes the association between nodes and attributes obvious and enables a tight integration of attribute-based and topology-based graph analysis tasks. Both aggregation methods described serve to reduce the space usage of the linearized tree while preserving the topology, and while preserving the desired level of information about the attributes. The aggregation is based on two principles: assigning nodes to be aggregated to the same row, and combining the explicit node-link layout with the implicit encoding for aggregated nodes and their leaves (family grids).

The Lineage genealogy visualization tool specifically can be broadly used with other genealogical datasets, e.g., to study autism, diabetes, or cancer. There are many groups at the University of Utah that make use of the Utah Population Database, and we have already established contact to other potential collaborators who are in need of a clinical genealogy visualization tool. Some of these datasets also have detailed attributes for non-affected cases, which will make our attribute-preserving aggregation approach even more valuable.

While our data is unique with respect to its scope, detailed genealogical datasets are becoming more common, as they have shown immense potential for population genetics [36]. We believe that our approach could also be adapted to datasets containing many small families (siblings, parents, grandparents, of affected individuals) as they are commonly collected to study the genetic disease of one family member.

10.1 Scalability

In contrast to other tools, such as the DOITree [23] our aggregation approach preserves all of the structure of the tree. This is suitable for trees with hundreds of nodes, but not for trees with tens of thousands of nodes or more. To scale to larger trees, these algorithms could be combined with hiding parts or the tree. Also, while the described algorithms work for any tree and any phenotype of interest, they are most efficient if the number of nodes of interest is small compared to the number of nodes in total. A common phenotype of interest for our collaborators is suicide, and the typical genealogies they study contain between 5-15% suicide cases. For these conditions, we found the resulting layouts to be very compact and useful.

We found Lineage to scale well to families with about 1500 individuals, which covers most families in our collaborators dataset (547 out of 550 families have less than 1000 individuals). We also experimented with the largest families in our dataset, which contain about 2500 individuals. For these families, we observe several seconds of wait time until the de-cycling and the layout is computed. We anticipate to be able to address these performance limitations through pre-computing and caching initial layouts.

In terms of the scalability of the visual encodings, we argue that Lineage produces a more readable layout in less space than Progeny, the tool that is currently used by our collaborators for displaying genealogies. Note that Progeny has only very limited capabilities of showing attributes by encoding attributes directly on the nodes, and displaying text underneath nodes, and attributes cannot be dynamically selected or manipulated. For a comparison between Progeny and Lineage, please refer to the supplementary material. When using suicide as a POI (the most common use case) and when using attribute-hiding aggregation, a family with about 400 individuals fits onto a single screen without scrolling (see Supplementary Figure S2). Larger families, attribute-preserving aggregation, or no aggregation more commonly require scrolling.

The number of attributes that can be displayed for each individual is limited by the horizontal screen size. On a large, 2560x1600 pixel display, about 20-40 dimensions can be shown, depending on the type (text and numerical columns need more space than binary categorical, for example). We found that this typically exceeds the number of attributes our collaborators would like to study simultaneously.

11 Conclusion and Future Work

In this paper, we introduced a novel approach for visualizing multivariate trees and tree-like graphs using a linearization approach. We demonstrate the usefulness of our approach by realizing it in the Lineage system, which is designed for the visualization of genealogies in a clinical context. Using Lineage, our collaborators are now able to efficiently explore the structure of large families and even multiple families at the same time, in addition to analyzing dozens of attributes for the individuals in these families. They can use Lineage to identify phenotypes of interest that appear in multiple families, and use this knowledge to inform and narrow down their search for genetic variants.

While Lineage in its current form is already highly useful to our collaborators, there are many directions in which it could be extended. Specifically, we currently deal only with a selected subset of the 3000 dimensions that are available for each of our cases. We plan to develop integrated visual and analytical methods to select dimensions of interest for any given subset of patients. For

example, the system could identify that for a given family, PTSD is a common comorbidity and suggest to the analyst to add PTSD to the table. This kind of approach is going to be especially important when we start to integrate the detailed genetic data that is available for many of these cases.

Finally, we are actively exploring other application areas for Lineage. We are currently talking to evolutionary biologists, who are excited about the potential of our multivariate tree visualization method for trait analysis in large phylogenetic trees.

Acknowledgments

We thank Asmaa Aljuhani and Annie Cherkaev for their contributions. We also thank our collaborators and the Visualization Design Lab at the University of Utah for the feedback, and the Caleydo team for their technical support. This work was supported in part by the US National Institutes of Health (U01 CA198935, R00 HG007583, R01MH099134) and the DoD - Office of Economic Adjustment (OEA), ST1605-16-01. We thank the Pedigree and Population Resource of the Huntsman Cancer Institute, University of Utah (funded in part by the Huntsman Cancer Foundation) for its role in the ongoing collection, maintenance and support of the Utah Population Database (UPDB). We also acknowledge partial support for the UPDB through grant P30 CA2014 from the National Cancer Institute, and from the University of Utah's Program in Personalized Health and Center for Clinical and Translational Science.

References

[1] R. Adrian. Tree Drawing Algorithms. In *Handbook of Graph Drawing and Visualization*, pages 155–192. CRC Press, 2013. Google-Books-ID: lQBrAAAAQBAJ.

[2] A. V. Bakian, R. S. Huber, H. Coon, D. Gray, P. Wilson, W. M. McMahon, and P. F. Renshaw. Acute Air Pollution Exposure and Risk of Suicide Completion. *American Journal of Epidemiology*, 181(5):295–303, 2015.

[3] R. Ball. Visualizing genealogy through a family-centric perspective. *Information Visualization*, 16(1):74–89, 2017.

[4] J. C. Barrett, B. Fry, J. Maller, and M. J. Daly. Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21(2):263–265, 2005.

[5] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1253–1260, 2008.

[6] R. L. Bennett, K. S. French, R. G. Resta, and D. L. Doyle. Standardized Human Pedigree Nomenclature: Update and Assessment of the Recommendations of the National Society of Genetic Counselors. *Journal of Genetic Counseling*, 17(5):424–433, 2008.

[7] R. L. Bennett, K. A. Steinhaus, S. B. Uhrich, C. K. O'Sullivan, R. G. Resta, D. Lochner-Doyle, D. S. Markel, V. Vincent, and J. Hamanishi. Recommendations for standardized human pedigree nomenclature. Pedigree Standardization Task Force of the National Society of Genetic Counselors. *American Journal of Human Genetics*, 56(3):745–752, 1995.

[8] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. D. Fekete. GraphDice: A System for Exploring Multivariate Social Networks. *Computer Graphics Forum (EuroVis '10)*, 29(3):863–872, 2010.

[9] A. Bezerianos, P. Dragicevic, J. Fekete, J. Bae, and B. Watson. GeneaQuilts: A System for Exploring Large Genealogies. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):1073–1081, 2010.

[10] B. Cannon, M. Hiremath, C. Jorcyk, and A. Joshi. CoVE: A Colony Visualization System for Animal Pedigrees. In *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction*, VINCI '14, pages 9:9–9:18, New York, NY, USA, 2014. ACM.

[11] W. S. Cleveland and R. McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[12] G. M. Draper and R. F. Riesenfeld. Interactive fan charts: A space-saving technique for genealogical graph exploration. In *Proceedings of the 8th Annual Workshop on Technology for Family History and Genealogical Research (FHTW 2008)*. Citeseer, 2008.

[13] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95(25):14863–14868, 1998.

[14] C. Fuchsberger, S. Miksch, L. Forer, and C. Pattaro. Analyzing Populations with Visual and Analytical Methods to Identify Family Clustered Diseases. *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, page 2243, 2007.

[15] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23. ACM, 1986.

[16] G. W. Furnas. A Fisheye Follow-up: Further Reflections on Focus + Context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 999–1008, New York, NY, USA, 2006. ACM.

[17] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A.-C. Gavin. Visualization of omics data for systems biology. *Nature Methods*, 7(3):56–68, 2010.

[18] M. Glueck, A. Gvozdik, F. Chevalier, A. Khan, M. Brudno, and D. Wigdor. PhenoStacks: Cross-Sectional Cohort Phenotype Comparison Visualizations. *IEEE Transactions on Visualization and Computer Graphics (VAST '16)*, 23(1):191–200, 2017.

[19] S. Goodwin, J. Dykes, S. Jones, I. Dillingham, G. Dove, A. Duffy, A. Kachkaev, A. Slingsby, and J. Wood. Creative User-Centered Visualization Design for Energy Analysts and Modelers. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2516–2525, Dec. 2013.

[20] S. Gratzl, N. Gehlenborg, A. Lex, H. Strobelt, C. Partl, and M. Streit. Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*, Chicago, IL, USA, 2015. IEEE.

[21] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.

[22] S. Haroz, R. Kosara, and S. L. Franconeri. ISOTYPE Visualization: Working Memory, Performance, and Engagement with Pictographs. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1191–1200, New York, NY, USA, 2015. ACM.

[23] J. Heer and S. K. Card. DOITrees Revisited: Scalable, Space-constrained Visualization of Hierarchical Data. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 421–424, New York, NY, USA, 2004. ACM.

[24] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization (Vis '91)*, pages 284–291, 1991.

[25] I. Jusufi, Y. Dingjie, and A. Kerren. The Network Lens: Interactive Exploration of Multivariate Networks Using Visual Filtering. In *2010 14th International Conference Information Visualisation*, pages 35–42, 2010.

[26] W. Katon. Asthma, Suicide Risk, and Psychiatric Comorbidity. *American Journal of Psychiatry*, 167(9):1020–1022, Sept. 2010.

[27] A. Kerren, H. C. Purchase, and M. Ward, editors. *Multivariate Network Visualization*. Number 8380 in Lecture notes in computer science. Springer, 2014.

[28] E. Kerzner, A. Lex, C. L. Sigulinsky, R. E. Marc, B. W. Jones, T. Urness, and M. Meyer. Graffinity: Visualizing Connectivity in Large Graphs. *Computer Graphics Forum (EuroVis '17)*, to appear, 2017.

[29] N. W. Kim, S. K. Card, and J. Heer. Tracing Genealogical Data with TimeNets. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 241–248, New York, NY, USA, 2010. ACM.

[30] B. Lee, L. Nachmanson, G. Robertson, J. M. Carlson, and D. Heckerman. PhyloDet: a scalable visualization tool for mapping multiple traits to large evolutionary trees. *Bioinformatics*, 25(19):2611–2612, Oct. 2009.

[31] B. Lee, L. Nachmanson, G. G. Robertson, J. Carlson, and D. Heckerman. Det.(distance encoded tree): a scalable visualization tool for mapping multiple traits to large evolutionary trees. *MSR Tech Report MSR-TR-2008-97*, 2008.

[32] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task Taxonomy for Graph Visualization. In *Proceedings of the AVI Workshop on BEyond time and errors: novel evaluation methods for information visualization (BELIV '06)*, pages 1–5, 2006.

[33] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and Evaluation of a Visual Analysis Framework for Gene Expression Data in its Biological Context. In *Proceedings of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE, 2010.

[34] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum (EuroVis '12)*, 31(3):1175–1184, 2012.

[35] Y. Liu, S. Dai, C. Wang, Z. Zhou, and H. Qu. GenealogyVis: A System for Visual Analysis of Multidimensional Genealogical Data. *IEEE Transactions on Human-Machine Systems*, PP(99):1–13, 2017.

[36] T. A. Manolio, F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon, A. Chakravarti, J. H. Cho, A. E. Guttmacher, A. Kong, L. Kruglyak, E. Mardis, C. N. Rotimi, M. Slatkin, D. Valle, A. S. Whittemore, M. Boehnke, A. G. Clark, E. E. Eichler, G. Gibson, J. L. Haines, T. F. C. Mackay, S. A. McCarroll, and P. M. Visscher. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, Oct. 2009.

[37] A. Mazeika, J. Petersons, and M. H. Böhlen. PPPA: Push and Pull Pedigree Analyzer for Large and Complex Pedigree Databases. In *Advances in Databases and Information Systems*, pages 339–352. Springer, Berlin, Heidelberg, Sept. 2006.

[38] M. J. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 16–23, Oct. 2005.

[39] P. McGuffin, A. Marušič, and A. Farmer. What can psychiatric genetics offer suicidology? *Crisis: The Journal of Crisis Intervention and Suicide Prevention*, 22(2):61–65, 2001.

[40] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister. Pathline: A Tool For Comparative Functional Genomics. *Computer Graphics Forum (EuroVis '10)*, 29(3):1043–1052, 2010.

[41] E. S. Mills. Genealogy in the information age: History's new frontier. *National Genealogical Society Quarterly*, 91(91):260–277, 2003.

[42] V.-P. Mäkinen, M. Parkkonen, M. Wessman, P.-H. Groop, T. Kanninen, and K. Kaski. High-throughput pedigree drawing. *European Journal of Human Genetics*, 13(8):987–989, May 2005.

[43] National Center for Health Statistics (US). *Health, United States, 2015: With Special Feature on Racial and Ethnic Health Disparities*. Health, United States. National Center for Health Statistics (US), Hyattsville, MD, 2016.

[44] C. Partl, S. Gratzl, M. Streit, A. M. Wassermann, H. Pfister, D. Schmalstieg, and A. Lex. Pathfinder: Visual Analysis of Paths in Graphs. *Computer Graphics Forum (EuroVis '16)*, 35(3):71–80, June 2016.

[45] C. Partl, A. Lex, M. Streit, D. Kalkofen, K. Kashofer, and D. Schmalstieg. enRoute: Dynamic Path Extraction from Biological Pathway Maps for Exploring Heterogeneous Experimental Datasets. *BMC Bioinformatics*, 14(Suppl 19):S3, 2013.

[46] T. Paterson, M. Graham, J. Kennedy, and A. Law. VIPER: a visualisation tool for exploring inheritance inconsistencies in genotyped pedigrees. *BMC Bioinformatics*, 13(8):S5, 2012.

[47] N. L. Pedersen and A. Fiske. Genetic influences on suicide and nonfatal suicidal behavior: Twin study findings. *European Psychiatry*, 25(5):264–267, June 2010.

[48] Progeny Genetics LLC. Progeny, 2016.

[49] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431 –2440, 2012.

[50] R. Shannon, T. Holland, and A. Quigley. Multivariate Graph Drawing using Parallel Coordinate Visualisations. Technical report, University of St Andrews, 2008.

[51] P. D. Shaw, M. Graham, J. Kennedy, I. Milne, and D. F. Marshall. Helium: visualization of large scale plant pedigrees. *BMC Bioinformatics*, 15:259, 2014.

[52] M. Sokolowski, J. Wasserman, and D. Wasserman. Polygenic associations of neurodevelopmental genes in suicide attempt. *Molecular Psychiatry*, Dec. 2015.

[53] J. Stasko and E. Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization (InfoVis '00)*, pages 57–65. IEEE Computer Society Press, 2000.

[54] H. Thiele and P. Nürnberg. HaploPainter: a tool for drawing pedigrees with complex haplotypes. *Bioinformatics*, 21(8):1730–1732, Apr. 2005.

[55] C. Tuttle, L. G. Nonato, and C. Silva. PedVis: A Structured, Space-Efficient Technique for Pedigree Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1063–1072, Nov. 2010.

[56] S. van den Elzen and J. van Wijk. Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2310–2319, 2014.

[57] R. E. Voorrips, M. C. A. M. Bink, V. D. Weg, and W. Eric. Pedimap: Software for the Visualization of Genetic and Phenotypic Data in Pedigrees. *Journal of Heredity*, 103(6):903–907, Nov. 2012.

[58] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive Measurements of Graph Aesthetics. *Information Visualization*, 1(2):103–110, June 2002.

[59] J. Xu, K. Kochanek, S. Murphy, and B. Tejada-Vera. National Vital Statistics Reports. Deaths: Final Data for 2007. *Center for Disease Control and Prevention Division of Vital Statistics*, 58, 2010.

Carolina Nobre is a PhD student at the Scientific Computing and Imaging Institute and the School of Computing at the University of Utah. Carolina works under the supervision of Alexander Lex at the Visualization Design Lab (VDL) and is interested in data visualization, particularly related to large multivariate networks.

Nils Gehlenborg is an Assistant Professor in the Department of Biomedical Informatics at Harvard Medical School. Nils obtained his PhD from the University of Cambridge and the European Bioinformatics Institute (EMBL-EBI). He is interested in visualization of biomedical data, in particular with applications in genomics, epigenomics, and cancer biology.

Hilary Coon is a Research Professor of Psychiatry at the Department of Psychiatry at the University of Utah. She received her PhD in Psychology from the University of Colorado and Institute for Behavioral Genetics, Boulder, Colorado in 1991. Hilary's primary research interests include finding genetic risk factors that lead to susceptibility to suicide, and also autism genetic risk. Her interests also include the development and application of statistical methods to genetic and phenotypic data, and genetics of other complex disorders, including cardiovascular disease, obesity, and lung disorders.

Alexander Lex is an Assistant Professor of Computer Science at the Scientific Computing and Imaging Institute and the School of Computing at the University of Utah. Before joining Utah he was a lecturer and a post-doctoral visualization researcher at the Harvard School of Engineering and Applied Sciences. He received his PhD from the Graz University of Technology in 2012. His primary research interests are data visualization, especially applied to molecular biology, and human computer interaction.