Interactive Maps for Static and Dynamic Relational Data

Stephen G. Kobourov and Michael F. Hammer
University of Arizona

1 Introduction

The main research goal of the project is to design and implement practical algorithms for static and dynamic visualization of relational data. Graphs are often used to capture relationships between objects and graph drawing techniques allow us to visualize such relationships. While much is known about static graph visualization, much remains to be done in the dynamic case, as well as in the case when graphs are not represented in the traditional node-and-link fashion. The focus of this proposal is on *map representations*, where clusters of vertices form countries and neighboring countries correspond to nearby clusters, and on *dynamic map visualization*, where the goal is to present relationships that evolve over time. Specifically, the design and implementation of algorithms for modeling and visualizing static and dynamic relational data with such non-traditional representations can make an impact on:

1. information visualization: modeling and visualizing static and dynamic data sets using the map metaphor, focusing on creating representations which make the underlying data understandable and visually appealing;

2. knowledge discovery: evaluating the impact of map representations, compared to traditional graphs, charts, and plots, especially focusing on the ability to discover non-trivial patterns and trends;

3. population genetics: combining data from the Y chromosome, mitochondrial DNA, and autosomal data and being able to interactively see the result should help distinguish the genomic footprint of natural selection from the signatures of demographic processes.

Map representations provide a way to visualize relational data with the help of the map metaphor. A classic example of a map representation are contact graphs, where regions represent vertices and edges are represented by the corresponding regions sharing borders. However, such representations are by definition limited to planar graphs. We can generalize the notion of a map representation to non-planar graphs as follows. Clusters of well-connected vertices form countries, and countries share borders when neighboring clusters are interconnected; see Fig. 1.

Dynamic map visualization deals with the problem of effectively presenting relationships as they change over time. Traditionally, dynamic relational data is visualized by animations of point-and-line graphs, in which vertices and edges fade in and out as needed. One of the main problems in dynamic visualization is that of obtaining individually readable layouts for each moment in time, while at the same time preserving the viewer's mental map (by not moving too many vertices from one layout to the next). A related problem is that of visualizing multiple relationships on the same dataset. Just as with dynamic data, the main problem is guaranteeing readability while preserving the viewer's mental map. Contact graphs and map representations offers great potential for intuitive and visually appealing presentation of dynamic data and for visualization of multiple relationships on the same dataset.

Population genetics studies hypotheses about population history. While the patterns of genetic inheritance dictate that there is a single ancestor (and a single phylogenetic tree) for all non-recombining portions of our genome, each portion of our genome will not necessarily have the same history. In other words, a gene tree can tell us about the history of a single locus, but not about the structure of the whole population. One of the major problems in evolutionary biology is to better understand the genomic and evolutionary factors shaping patterns of human variation and to test models of human origins. Given the human genome sequence we are now in a position to examine patterns of variation across the entire genome in multiple human populations, using multiple relationships on the same set of objects. Interactively combining these multiple relationships, in an intuitive and visually appealing fashion, will facilitate knowledge discovery and hypothesis testing.

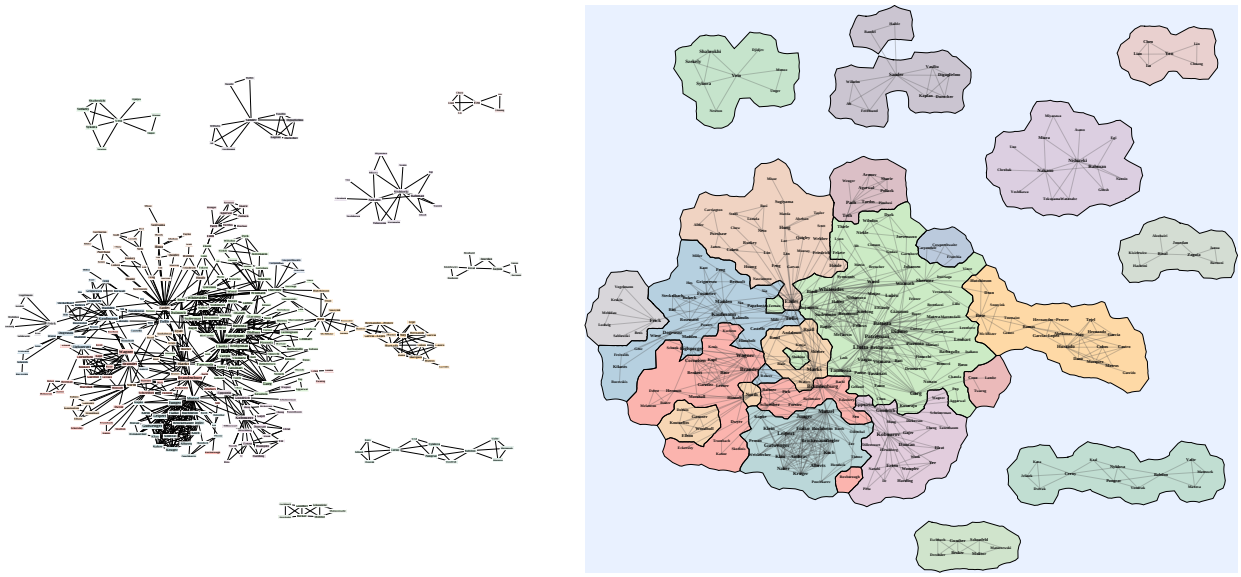


Figure 1: A collaboration graph shown as a node-link diagram and as a map. Vertices are researchers in the graph drawing community and edges connect pairs of authors who have collaborated on at least one research paper. (This is a zoomable high resolution image.)

1.1 Motivation

Attempts to visualize data often lead to overly complicated-looking images which convey the complexity of the underlying data, but which do not necessarily make that data easier to understand. A recent example that made the news due to its overwhelming complexity is the US counter-insurgency strategy in Afghanistan, visualized as a directed graph; see Fig. 2.

A great deal of information visualization tools that attempt to deal with large and complex underlying data end producing impressive looking images, and provide powerful ways to interact with the data. However, often the results are nearly as complex as the underlying data itself and the learning curve required for productive interaction is steep. As a result, most information visualization tools that are used in practice are traditional ones such as pie-charts, box-plots, and node-and-link graphs. Familiarity with a visualization metaphor can be a powerful motivation for its use, so instead of creating new visualization metaphors, we would like to focus on making better use of existing ones.

Maps are a familiar way of presenting geographic data. Additional properties can also be shown with the help of contours, color-relief, and 3D terrain overlays. Due to abundance of maps – subway maps, train maps, political maps – most people find them intuitive and non-intimidating. Moreover, people are familiar with the traditional way of interacting with digital maps via zooming and panning. This is one of the reasons why maps offer a promising way for visualizing data. While many general users may be put off by overwhelming-looking infographics, maps offer a familiarity which allows us to present complex information. The second reason is that maps tend to encourage viewers to spend time examining them. Preliminary informal experiments indicate that people spend twice as long looking at a map, than at a graph of the same data. While in most computer science applications we like to get things done faster, in this case it is an advantage to have a visualization that is aesthetically appealing and unobtrusively ends up encouraging the viewer to spend more time. Our third reason is that people perform unsolicited knowledge discovery tasks with maps. For example, a map of 1000 books from Amazon led several mothers with children to observe that the "gateway" to the Twilight books seems to be Victorian literature such as Jane Austen and Emily Bronte novels. Several men pointed out that Ayn Rand's "Atlas Shrugged" is the main connection from mainstream literature to the fringe with books such as "Liberal Fascism: The Secret History of the The American Left"; see Fig 3.

There is more than just anecdotal evidence to support the idea that maps of data can be a powerful and effective way to visualize relational data. An excellent recent experiment with user-generated graph layout collected data from over 70 users of IBM's ManyEyes online data visualization tool, to explore the types of graph layouts that people prefer [79]. The results show that users invariably construct layouts that distinctively group clusters in a spatial region that does not overlap with the spatial region occupied by another cluster. This is accomplished at the expense of "stretching" some edges, and "shrinking" others. Moreover, 80% of the users used the edges in a cluster to visually

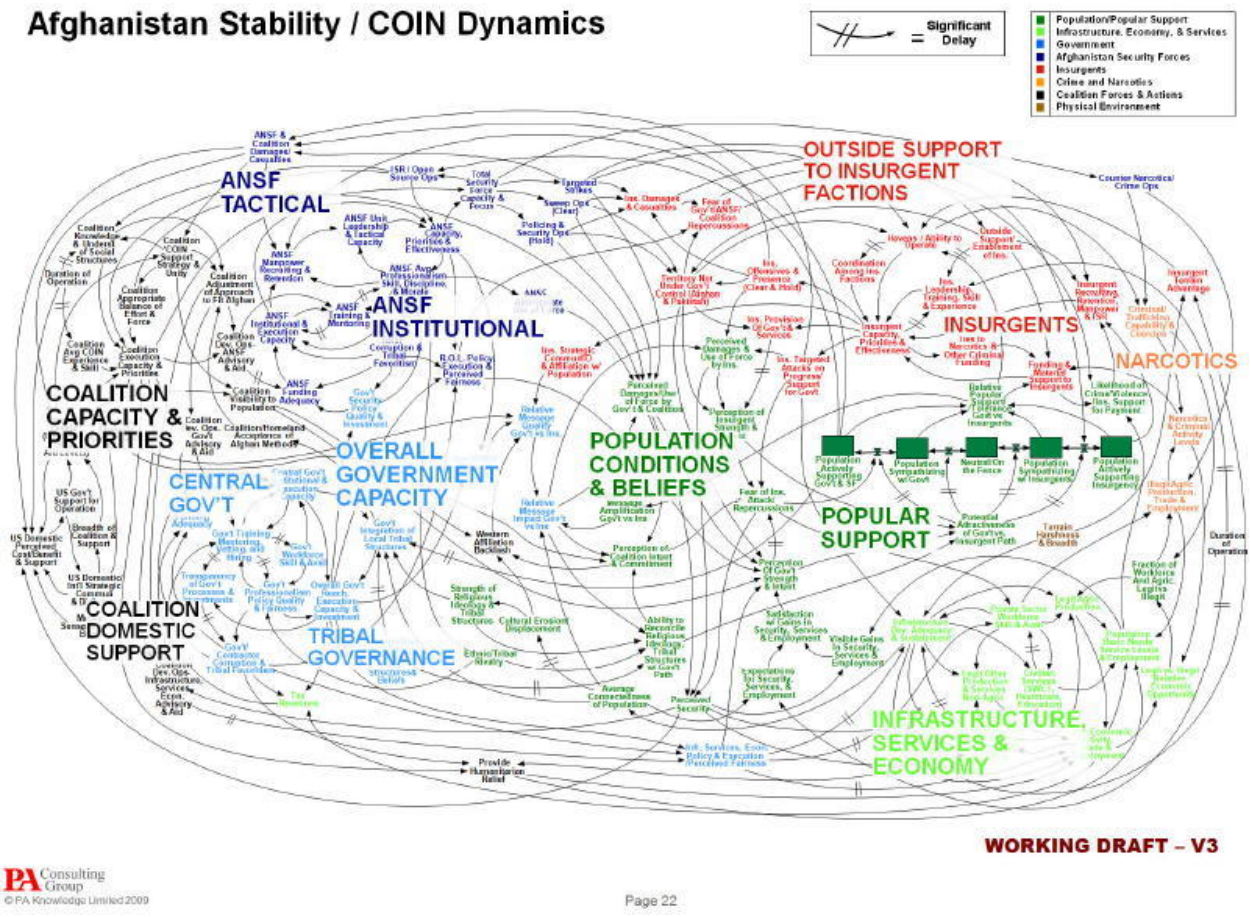


Figure 2: A hair-ball of a graph: the US counter-insurgency strategy in Afghanistan.

delineate the cluster itself, creating a convex hull around the nodes in the cluster. As the authors of the study point out, although this might seem obvious in retrospect, no automated layout algorithms explicitly attempt to do that. Our map layouts, with clusters represented by regions with clear cluster-hugging borders, achieve both of these user-desired features: explicit clustering and explicit cluster boundaries.

2 Map Representations

Relational datasets are often visualized as collection of points in 2D space using principal component analysis [60], multidimensional scaling [67], and force directed algorithms [52], which tend to put similar items next to each other. Visual examination often suffices to identify the presence of clusters. Sometimes, however, the clusters are not as easy to see and additional visual clues are needed to highlight them. One possibility is to use cluster analysis algorithms, such as k-means or hierarchical clustering algorithms [59, 71] to explicitly define clusters. While in small examples it is possible to convey the cluster information with the use of colors and proximity, this becomes difficult to do with large data. Common problems include dense clusters, overlapping labels, and clusters that lack clearly defined boundaries.

In this part of the project we propose the use of maps as a way to achieve this explicit visual definition of clusters. There are several reasons that such a representation can be more useful. First, by explicitly defining the boundary of the clusters and coloring the regions, we make the clustering information clear. Second, as most dimensionality reduction techniques lead to a 2-dimensional positioning of the data points, a map is a natural generalization. Finally, while graphs, charts, and tables often require considerable effort to comprehend, a map representation is more intuitive, as most people are very familiar with maps and even enjoy carefully examining maps.

Figure 1 shows a collaboration graph with a traditional node-and-link representation and with the proposed geographic-

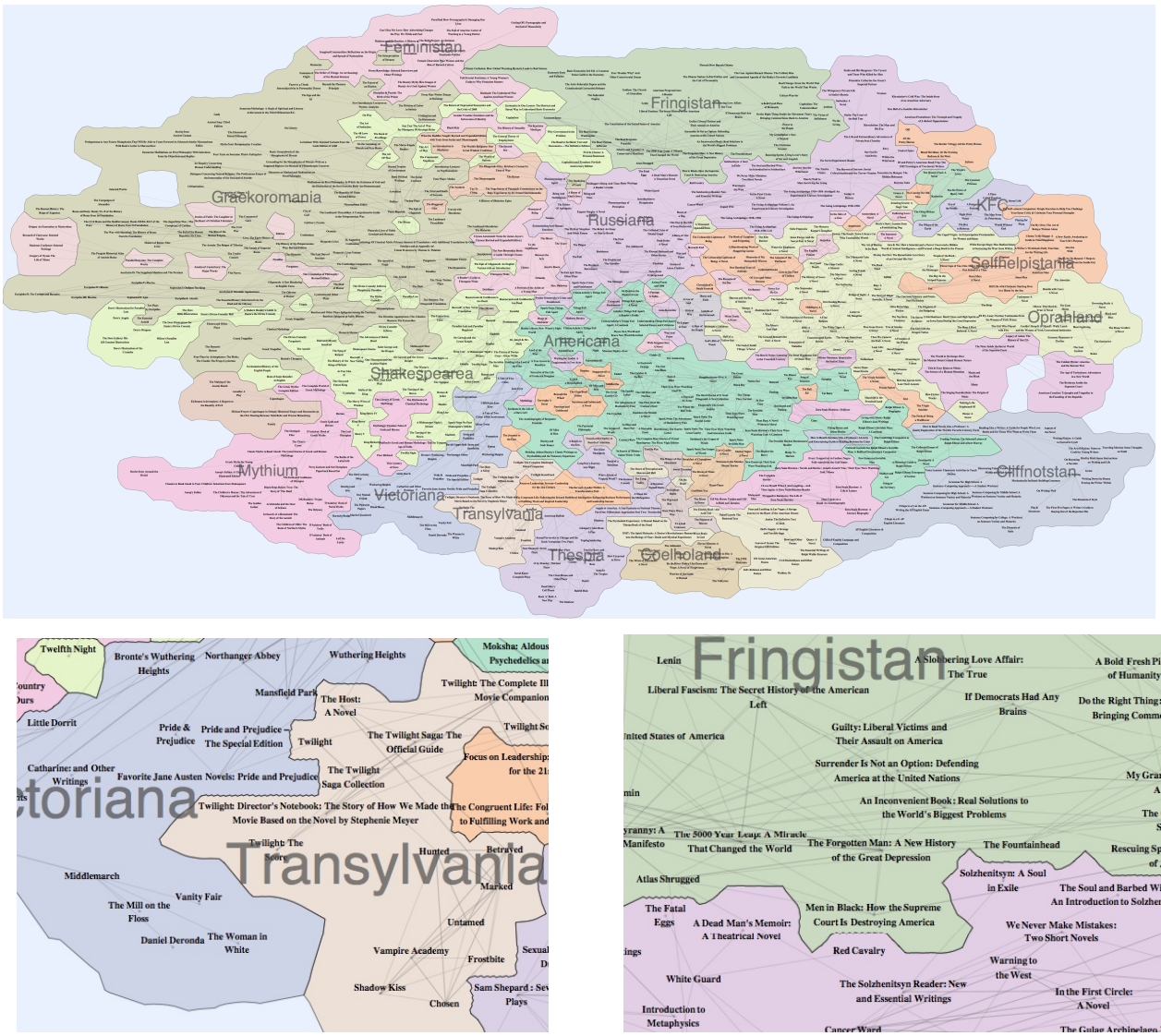


Figure 3: **Top**: BookLand map of the 1000 books most related to George Orwell's 1984, based on the "Customers who bought this item also bought" relationship from `Amazon.com`. **Left**: Jane Austen and Emily Bronte as gateways to the Twilight vampire literature. **Right**: Ayn Rand's Atlas Shrugged as the gateway to fringe literature.

map metaphor. Vertices are researchers in the graph drawing community and edges connect pairs of authors who have collaborated on at least one research paper. Note that the vertices and edges are placed in the exact same locations in both embeddings and even the colors of the various clusters is the same. Yet, the map makes the grouping of tightly connected groups of vertices explicit. Moreover, when a cluster is broken down into several disjoint components, it draws the viewer's eye to it and leads to data discovery. For example, the orange cluster in the SouthWest corner of the map is broken into three components. One of the first questions people looking at this maps ask is why is this cluster fragmented. Most answer that question when they observe that the most peripheral fragment contains researchers who do not collaborate with anyone else, while the most central cluster contains the most active researchers from this group. People familiar with the graph drawing community can also supply the details that this cluster contains mostly North American researchers (whereas almost no other clusters do). The middle fragment contains one researcher, Stephen North, who is the head of the Information Visualization group at AT&T Research Labs, and the most peripheral fragment contains researchers from that group who do not have collaborations outside that group. Note that not all map artifacts are as meaningful at this (e.g., the two small green regions containing researchers Shubina, and Fernau).

2.1 Prior Work on Map Representations

There is little previous work on generating map representations of graphs. Most related work deals with representing accurately and appealingly a given geographic region, or on re-drawing an existing map subject to additional constraints. Examples of the first kind of problem are found in traditional cartography, e.g., the Mercator's 1569 projection of the sphere into 2D Euclidean space. Examples of the second kind of problem are found in cartograms, where the goal is to redraw a map so that the country areas are proportional to some metric, an idea which dates back to 1934 [74] and is still popular today (e.g., the New York Times red-blue presidential election maps of the US, with states drawn proportional to population). Somewhat similar to cartograms, treemaps [75] and squarified treemaps [10] represent hierarchical information by means of space-filling tilings, allocating area proportional to some important metric.

Placing imagined places on a map as if they were real countries also has a long history, e.g., the 1930's Map of Middle Earth by Tolkien [77]. A more recent example is xkcd's Map of Online Communities [1]. While most such maps are generated in an ad hoc manner and are not strictly based on underlying data, they are often visually appealing. We proposed the use of a geographic map metaphor for visualizing relational data in the context of visualizing recommendations, where the underlying data is TV shows and the similarity between them [55]. This approach combines graph layout and graph clustering, together with appropriate coloring of the clusters and creating boundaries based on clusters and connectivity in the original graph.

2.2 Research Problems on Map Representations

In this part of the project, we will consider practical approaches to visualizing relational data with the help of the geographic map metaphor. No assumptions will be made about the nature of the underlying data, other than that it consists of objects and relationships between these objects. For example, the objects can be books and the relationship is "people who bought this book at Amazon also bought", or the objects could be researchers and the relationship is research paper collaboration, or the objects could be human DNA samples and the relationship is similarity, based on a set of DNA markers. In this section we consider static datasets and in the next section dynamic datasets and datasets with multiple relationships.

Our main goal is to obtain a map of the underlying relational data that is visually appealing and which shows more than just the underlying vertices and edges. Specifically, by explicitly grouping vertices into different colored regions, we can easily see the individual clusters and the relations between the clusters. Moreover, this explicit grouping makes it easy to identify central and peripheral clusters, as well central and peripheral vertices within each cluster. Finally, cut vertices and edges, typically end up on the border between two clusters, making clear which objects and/or relations allow for the connection between two disparate parts of the data.

P1. Canonical Map: In exploratory visualization, the goal is not just a snapshot of the data, but a visualization which can be used repeatedly to answer different questions about the underlying data. While many people find traditional node-link graphs too complex and intimidating, most are very comfortable with map-related concepts: items within a country are similar to each other, areas separated by a mountain range are difficult to connect, islands might have atypical qualities, etc. With this in mind, we would like to compute a *canonical map* representation of the data. The map metaphor becomes more powerful as a viewer becomes familiar with one fixed map layout, which we call the canonical map layout. Different properties of the data can be visualized as "heat map" overlays on top of the canonical map. In the collaboration example, the canonical map is based on data collected over 10 years. By coloring the regions around recently active researchers with hot colors and regions around inactive researchers with cool colors, we can easily highlight researchers active in the last 5 years. Consider asking the question: who works on "systems" and who works on "algorithms" in this community? We can answer this question with a couple of 3D terrain overlays that correlate the occurrence of the two keywords in papers by these authors as "height". Such a view would make it easy to see that North American researchers, and group associated with research labs dominate the peaks in the "systems" terrain, whereas European groups claim most of the high mountains in the "algorithms" terrain.

Here we describe the proposed method for generating a canonical map that captures, as well as it is possible in one fixed layout, the relationships between the objects in the underlying dataset. The input to the algorithm is a relational dataset from which a graph $G = (V, E)$ is extracted. The set of vertices V corresponds to the objects in the data (e.g., biological species) and the set of edges E corresponds to the relationship between pairs of objects (e.g., genetic similarity between pairs of species). In its full generality, the graph is vertex-weighted and edge-weighted, with vertex weights corresponding to some notion of vertex importance and edge weights corresponding to some notion of the closeness between a pair of vertices.

In the first step we will obtain an embedding of the graph in the plane using our multi-dimensional embedding algorithm for large graphs [53]. In the second step, a cluster analysis is performed in order to group vertices into clusters. For this step we can use off-the-shelf algorithms such as modularity-based clustering [72] or geometric clustering algorithms such as k-means [71]. In the third step we will create a geographic map corresponding to the dataset, based on a modified Voronoi diagram of the vertices, which in turn is determined by the embedding and clustering. Here "countries" are created from clusters, and "continents" and "islands" are created from groups of neighboring countries. Borders between countries and at the periphery of continents and islands are created in fractal-like fashion. Finally, colors are assigned so that no two adjacent countries have the same or similar shades. Further geographic components are added to strengthen the map metaphor. For instance, edges can be made semi-transparent or even modified to resemble road networks.

The quality of the resulting canonical map can be evaluated in a quantitative fashion by measuring data distortion, for example, by the rate of false positives (pairs of objects that have low similarity but are represented with small Euclidean distance in the map) and false negatives (pairs of objects that have large similarity but are represented with large Euclidean distance in the map) in the map. The goal of minimizing false positives and false negatives is similar to the goals of PCA [60] and MDS [67]. However, with maps we can visually alleviate the impact of data distortion as follows: with false positives we can separate physically close regions that are not similar by adding lakes, rivers, or mountain ranges; with false negatives we can indicate that the objects are similar, even if they are not very close to each other by making them belong to the same (possibly disjoint) country, colored with the color or labeled with the same name.

P2. Knowledge Discovery with Maps: Informal experiments indicate that the data from Fig. 1 presented as a map attracts more viewers and holds the viewer's attention longer than the same data presented as a graph. As part of this informal experiment the PI posted a large printout of a graph outside his office for a week, and then replacing it with a map. The number of people (faculty, staff, and students) who stopped by to look at the map was more than double that of the people who stopped to look at the graph. The average amount of time spent looking at the map was also more than double that for the graph.

It is possible that the "novelty" of the map image is responsible for the greater interest and longer viewing times, but we believe that it is the "less boring" aspect of the map visualization compared to the node-and-link graph visualization that is at work here. We would like to test this formally, with the help of a carefully designed user study that would test whether maps indeed are more "attractive" way for representing the underlying data, and whether they lead to longer voluntary exploration times.

We also noticed that people tend to perform spontaneous knowledge discovery, even without being asked to do so. This leads us to believe that presenting the underlying data as a map, might have tangible and measurable advantages over presenting the data as a graph. Typical experiments testing the effectiveness of one graph drawing algorithm over another include performing timed tasks, such as finding the shortest path between two vertices and finding the average degree of a set of vertices. The type of knowledge discovery that seems conducive to map visualization is different. We will work on carefully designing experiments that would test how well maps do at exploratory visualization (what is in the data and where) and as well as on targeted tasks, such as identifying vertices that would disconnect the graph (e.g., the "gateway" vertices that users spontaneously discovered).

P3. Aesthetically Appealing Maps: Here we consider several aspects of the map generation that would help generate visually appealing maps, such as the shapes of countries, islands, and continents. Recall that the third step of the proposed approach for obtaining a map layout is the one that deals explicitly with the map-like properties of the representation.

A naive way to obtain countries from the clustered and embedded data would be to create a Voronoi diagram of the vertices, together with fours points on the four corners of the bounding box; see Fig. 4(a). This would result in aesthetically unappealing maps with jagged outer boundaries and sharp corners. A more map-like appearance can be obtained by placing additional dummy points that are sufficiently far away from the set of real vertices, would lead to more rounded boundaries. The addition of random points on the outskirts would lead to some randomness of the outer boundaries (continents and islands), thus making them look less artificial; see Fig. 4(b).

We would also like to address the practical problem of country size. In the research collaboration example, very active researchers could be associated with larger areas, and tightly connected clusters of active researchers should be associated with larger countries. One way to accomplish this would be to associate large vertices and large countries with large label sizes and ensure that labels do not overlap. To make areas proportional to the label size, we can add more dummy points along the bounding boxes of the labels; see Fig. 4(c). To ensure non-jagged inner boundaries, we can perturb the dummy points randomly, instead of having the boundaries defined by the rectangle bounding boxes.

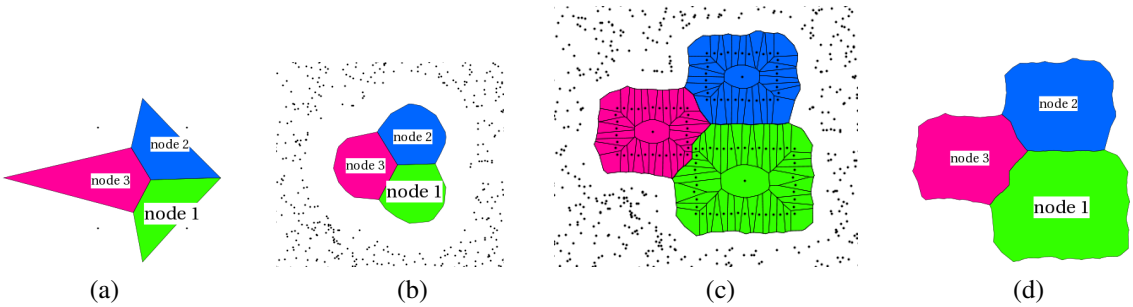


Figure 4: (a) Map from a Voronoi diagram of vertices and corners of the bounding box. (b) Better construction of outer boundaries through placement of random points. (c) Voronoi diagram with additional points around the bounding boxes of the labels. (d) The final map.

Voronoi cells that belong to the same vertex will be colored in the same color, and cells that correspond to the dummy points will not be shown. Cells of the same color will be merged to give the final map; see Fig. 4(d).

P4. Map Coloring Problem: The 4-color theorem ensures that a conventional geographical map can be colored with four colors in such a way that no neighboring countries have the same color. However, this only holds true under the assumption that each country is a single contiguous region in the plane. Countries in our maps can, and indeed do, have multiple disjoint regions. When the clustering and embedding algorithm are well-paired such fragmentation can be meaningful as it can indicate weak clusters or subclustering. Thus, once a color has been assigned to a country, that color cannot be reused for another country. As a result, the problem of assigning the most suitable color becomes very important. Specifically, we would like to minimize the color-similarity between neighboring countries.

With this in mind, we define the *country graph*, $G_c = \{V_c, E_c\}$, to be the undirected graph where countries are vertices, and two countries are connected by an edge if they share a non-trivial boundary. We then consider the problem of assigning colors to nodes of G_c so that the color distance between nodes that share an edge is maximized. More formally, let C be the color space, i.e., a set of colors; let $c : V_c \to C$ be a function that assigns a color to every vertex; and let $w_{ij} \geq 0$ be weights associated with edges $\{i, j\} \in E_c$. Let $d : C \times C \to R$ be a color distance function. Define the vector of color distances along edges to be $v(c) = \{w_{i,j}\ d(c(i), c(j)) \mid \{i, j\} \in E_c\}$. Then we are looking for a color function that maximizes this vector with respect to some cost function, such as:

$$\max_{c \in C} \sum_{\{i,j\} \in E_c} w_{i,j}\ d(c(i), c(j))^2 \quad \text{(2-norm)}$$

The weights along the edges can be used to model the undesirable effect of two nearby but not connected countries having very similar colors by making the country graph a complete graph, and assigning edge weights to be the inverse of the distance between two countries.

Dillencourt *et al.* [23] investigated the case where all colors in the color spectrum are available. They proposed a force-directed model aimed at selecting $|V_c|$ colors as far apart as possible in the color space. However in our map coloring problem, for aesthetic reasons, we are limited to "map-like" colors, and our color space is discrete. Therefore we model our coloring problem as one of vertex labeling, where our color space is $C = \{1, 2, \ldots, |V_c|\}$, and the color function we are looking for is a permutation that maximizes the labeling differences along the edges.

Although this is natural vertex labeling problem, there seem to be no prior results for it. The complementary problem of finding a permutation that *minimizes* the labeling differences along the edges is well-studied. For example, in the context of minimum bandwidth or wavefront reduction ordering for sparse matrices, it is known that the problem is NP-hard, and a number of heuristics [57] were proposed.

We will study the complexity of this new coloring problem, looking for a practical polynomial time approximation algorithm. One potentially good heuristic could be to order vertices using the Fiedler vector from the spectral decomposition of the country graph [13]. To do this we solve the continuous problem:

$$\max \sum_{\{i,j\} \in E_c} w_{i,j}(c_i - c_j)^2, \text{ subject to } \sum_{k \in V_c} c_k^2 = 1,$$

where $c \in R^{|V_c|}$. In this solution c is the eigenvector corresponding to the largest eigenvalue of the weighted

Laplacian of the country graph, while the Fiedler vector (the eigenvector corresponding to the second smallest eigenvalue) minimizes the objective function above. Once the above minimization is complete, we can use the ordering defined by the eigenvector as a (heuristic) solution for our coloring problem.

P5. Defragmentation: In large datasets with thousands or tens of thousands of data points we might have hundreds of "countries". Coloring each country with a unique color would be required if they are made up of many disjoint regions. This would make identifying regions with countries difficult. In such a setting it might be highly desirable to have contiguous countries, and a small fixed set of colors which can be reused as in traditional geographic maps. In order to do this we can carefully pair the clustering and embedding algorithms to ensure contiguous clusters, or we post-process the data to "defragment" the map.

With the first option we can choose our clustering and embedding algorithms so that they complement each other, in the sense that vertices in the same cluster end up geometrically close to each other in the embedding. For example, geometric clustering such as k-means can be computed after the data points have been embedded in the plane, so that each cluster is contiguous. Alternatively, force-directed embedding algorithms rely on similar principles as modularity-based clustering, as pointed out by Noack *et al.* [73].

With the second approach we can apply an arbitrary combination of clustering and embedding algorithms and post-process the data to ensure contiguous clusters. Specifically, one way to accomplish this would be to "strengthen" the edges between vertices in the same cluster. Alternatively, we can add a dummy cluster-vertex for each cluster and connect it to all the vertices in that cluster. Then by strengthening these edges appropriately, we can "pull" disjoint regions together.

3 Dynamic Map Visualization

While static graphs arise in many applications, dynamic processes give rise to graphs that evolve through time. Such dynamic processes can be found in software engineering, telecommunications traffic, computational biology, and social networks, among others. Visualization of large evolving graphs was the topic of a Dagstuhl Seminar, co-organized by the PI [63]. Two other recent Dagstuhl seminars were organized on the related topics of dynamic software visualization [3] and visualization of evolving networks [76].

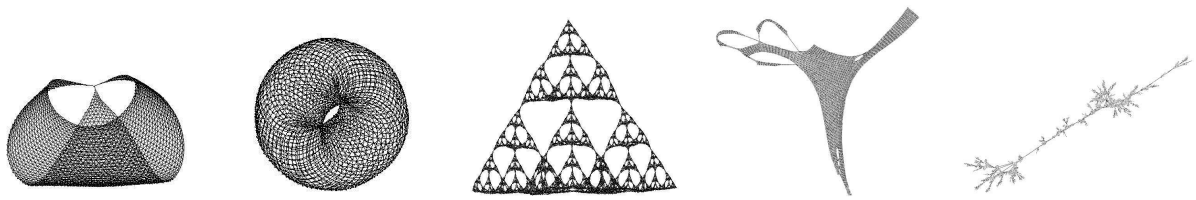
The input to this problem is a series of graphs defined on the same underlying set of vertices. If the graphs in the series are trees or other planar graphs, as is the case in some biological applications (phylogenetic trees and split networks), then some of our planar simultaneous embedding techniques can be used to derive good layouts. For general graphs, however, there is little that we can guarantee. As a consequence, nearly all existing approaches to visualization of evolving and dynamic graphs are based on the force-directed graph embedding method discussed below.

For layout of evolving and dynamic graphs, there are two important criteria to consider: (1) *readability* of the individual layouts, which depends on aesthetic criteria such as display of symmetries, uniform edge lengths, and minimal number of crossings; and (2) *mental map preservation* in the series of layouts, which can be achieved by ensuring that vertices and edges that appear in consecutive graphs in the series, remain in the same location. These two criteria are often contradictory. If we obtain individual layouts for each graph, without regard to other graphs in the series, we may optimize readability at the expense of mental map preservation. Conversely, if we fix the common vertices and edges in all graphs once and for all, we are optimizing the mental map preservation yet the individual layouts may be far from readable. Thus, we can measure the effectiveness of various approaches for visualization of evolving and dynamic graphs by measuring the readability of the individual layouts, and the overall mental map preservation.

Extending traditional graph embedding algorithms from static to dynamic graphs is a difficult problem. The main challenges are preserving the viewer's mental map under the dynamics in the data, readability of each individual layout, and effective visualization of the changes happening on the map. Extending the map visualization to large-scale dynamic data poses additional challenges. Whereas in dynamic graph embedding it is perfectly reasonable to have vertices move from one moment in time to the next, moving "countries" and "cities" within the countries in a map representation should be carefully designed and only rarely used.

3.1 Prior Results on Dynamic Maps

We began studying the problem of drawing and displaying series of related general graphs by exploring theoretical and practical aspects of the problem. In the theoretical aspects we focused on simultaneous embedding of multiple



(a) (b) (c) (d) (e)

Figure 5: Synthetic (a-c) and real-world graphs (d-e), drawn with our GRIP algorithm [54]: (a) knotted-triangular mesh with 2,016 nodes; (b) torus with 4,000 nodes; (c) Sierpinski pyramid with 8,194 nodes; (d) financial-data with over 10,000 vertices; (e) representation of an electronic circuit with over 20,000 vertices.

planar graphs, and in the practical aspects we focused on algorithms for visualizing dynamic graphs, without additional assumptions about the nature of the underlying data.

Our first theoretical result was to show the existence of planar simultaneous geometric embeddings for pairs of simple graphs (paths, cycles, caterpillars) as well as showing the infeasibility of such embeddings for other pairs (general planar graphs, outer-planar graphs) [6, 33]. Using a modified force-directed approach, we developed a framework for visualizing a series of related graphs, while controlling the balance between mental map preservation (keeping common vertices and edges in the same places across the graphs in the series) and readability of each of the graphs [17, 30, 31, 35].

Using planar simultaneous geometric techniques allowed us to solve an open problem about the geometric thickness of low degree graphs. In particular, we showed that degree-four graphs have geometric thickness of two [24]. We also considered the simultaneous embedding of a planar graph and its dual on a small integer grid such that the edges are represented with straight-line segments and the only crossings are between primal-dual pairs of edges. While the existence of such embeddings was known (Brightwell and Scheinerman [9] and Tutte [78]), the bounds on the necessary area were unknown. We gave an $O(n)$ time algorithm that simultaneously embeds a 3-connected planar graph and its dual on a $(2n-2) \times (2n-2)$ integer grid, where n is the total number of vertices in the graph and its dual [32].

Practical graph embedding algorithms are based on the force-directed method as defined by Tutte [78], or the more general the spring layout method of Eades [26] and Fruchterman and Reingold [51]. In these methods, there are repulsive forces between all nodes, but also attractive forces between nodes which are adjacent. Alternatively, forces between the nodes can be computed based on their graph theoretic distances, determined by the lengths of shortest paths between them. The algorithm of Kamada and Kawai [61] uses spring forces proportional to the graph theoretic distances. In general, force-directed methods define an objective function which maps each graph layout into a number in $\mathcal{R}^+$ representing the energy of the layout. This function is defined in such a way that low energies correspond to layouts in which adjacent nodes are near some pre-specified distance from each other, and in which non-adjacent nodes are well-spaced. A layout for a graph is then calculated by finding a (often local) minimum of this objective function.

The utility of the basic force-directed approach is limited to small graphs and results are poor for graphs with more than a few hundred vertices. We developed an algorithmic framework that extends the functionality of force-directed methods to graphs with tens of thousands of vertices [53]. Our algorithm employs a multi-scale technique based on a maximal independent set filtration of vertices of the graph. While most existing force-directed algorithms begin with an initial random placement of all the vertices, our algorithm attempts to place vertices "intelligently" close to their final positions. Furthermore, we embed the graph in high dimensional space and obtain 2D or 3D layout by computing appropriate projections. Our implementation of this algorithm, GRIP [54], produces good layouts for graphs with hundreds of thousands of vertices in a few seconds; see Fig. 5.

We began trying to extending this work to dynamic graphs by considering aggregated views, where all the graphs are displayed at once, merged views, where all the graphs are stacked above each other, and animation views, where only one graph is shown at a time, and morphing is used when changing between graphs (fading in/out vertices and edges that appear/disappear). Focusing on the animation/morphing approach, we developed a framework for controlling the balance between the readability of individual graphs and the overall mental map preservation in our system for Graph Animations with Evolving Layouts, GraphAEL [31, 45]. We have used this framework to visualize software evolution [17], social networks [30], and the behavior of dynamically modifiable code [25].

3.2 Research Problems on Dynamic Map Visualization

P6. Combined Mapping: Mental map preservation is important when visualizing dynamic data or when visualizing multiple relationships defined on the same underlying set of objects. In general, vertices and edges may appear and disappear over time. If a vertex appears, then disappears, and appears again, it would be desirable to use the same location in the layout.

Our approach will be to rely on the "canonical map" described above, but now redefined to accommodate the time component or the multiple relationships. In order to build a generic canonical map for the entire dataset we can take the union of all vertices and edges that appear in the data. We then compute the canonical map, which stores the position information for every data point in the dataset at any moment in time. This map is implicit, and is typically going to be associated with a much larger graph than the map that is actually shown. For example, if we are visualizing the evolution of a research collaboration over 10 years, we take as input the union of all the 10 graphs defined for each year. We compute a canonical map for this data, which stores the position of every researcher. When viewing a particular year we only show the data relevant for that year. In this way, as long as the same canonical map is used, the same vertices appear in the same position, thereby helping preserve the viewer's mental map.

The challenging problem here is that of combining the datasets in the best possible way. Specifically, different data sets might contain more "reliable" information, or within a given dataset, some individual relationships may be more reliable than others. We will design an interactive system to allow the user to explore ways to combine the input datasets, in such a way that some datasets can influence the final results more than others, and within datasets some relationships can influence the final result more than others. This process can be automated for data which comes with pre-assigned uncertainty values, and can be interactive otherwise. For data that comes with training and testing sets this process can also be automated.

P7. Map Readability: Building a canonical map for the entire dataset (defined over time or by several different relationships on the same underlying set of objects) can be a difficult task. Both the clustering and embedding algorithms used typically work well for small non-uniform datasets. For example, clustering algorithms have a difficult time with fully connected graphs where all the edges have the same weight. The only "natural" cluster is the entire graph.

If we build the canonical map for the entire dataset by taking the union of all vertices and edges that appear in the data, as described above, we might end up with a uniform graph. With large dynamic datasets, as well as with data associated with multiple relationships, the union of all the data might obscure whatever meaningful structural information that might have been present. In this case, the clusters might not be very meaningful, and the countries might be very fragmented.

There are several possible ways to alleviate such problems. In the case of dynamic data we might give more weight to recent relationships and then use a subset of the recent relationships for the embedding and clustering steps of the canonical map creation. For example, in the research collaboration graph we might use just the last 2-3 years instead of all 10 in the generation of the canonical map. In the case of multiple relationships on the same data we might choose one or several dominant relationships which can be used in the embedding and clustering steps. For example, in phyologenetics, the relationships between species defined by gene similarity might be used to generate the canonical map. In both cases, post-processing for defragmentation can also be applied to ensure more contiguous (or only contiguous) countries.

P8. Visualizing the Change: Using a map, one possible way to visualize changes over time is to modify the font sizes of the labels according to some importance metric and to show appearing and disappearing edges (or edges that get stronger/weaker over time). This is similar to traditional geographic maps, where the names of major cities are drawn with large fonts, and smaller towns with smaller fonts. By applying this modification to each map frame and concatenating them, we can visualize the trends with an animation.

It is likely that in animations created by the above procedure, the differences between consecutive time slices may be difficult to spot, as the font size change may be too subtle. Too much of a good thing (mental map preservation) can be bad. As we would like to keep the mental map of viewers unchanged from one frame to the next, and just changing the font sizes does not convey the changes in the data, we can employ another visual cue that is well suited to maps, namely heat maps; see Figure 6.

P9. Map Tool: Static graph visualization has reached the stage where several general-purpose tools, such as GraphViz [28] and yED [82] that provide excellent functionality. However, there are no off-the-shelf tools for visualizing dynamic data, or data defined by multiple relationships on the same set of objects. The most tangible practical goal of this project would be to generate exactly such a tool. With the help of graduate and undergraduate students, we will put together a software tool that takes as input a relational dataset and produces a customizable map as output. A

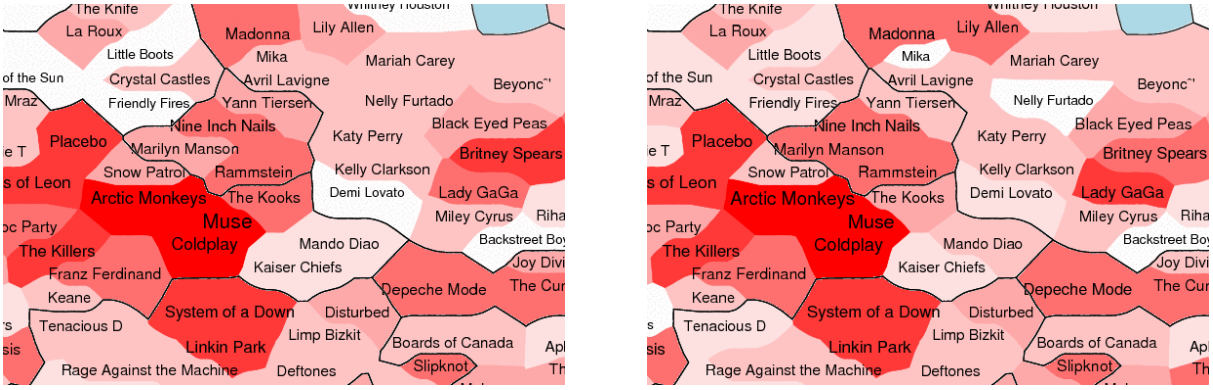


Figure 6: Two consecutive heatmaps, one month apart in the fall of 2009, showing increased interest in Lady Gaga at the expense of nearby Britney Spears. The data comes from the Internet radio station `last.fm`, which tracks millions of listeners.

default setting should produce a generic map with default clustering and embedding settings, and optional settings can be used to control the number of countries, the clustering algorithm, the layout algorithm, the level of fragmentation, heatmap overlays, etc. The map viewer should allow for standard operations such as pan and zoom, to allow for the exploration of large datasets. Details about how undergraduates students will be used in the development of this tool are discussed in Section 5 where we address the educational component of this project.

4 Population Genetics Applications

One of the major problems in evolutionary biology is to better understand the genomic and evolutionary factors shaping patterns of human variation and to test models of human origins. The human genome holds clues to the mystery of human origins. With the recent completion of the first draft of the human genome sequence we are now in a position to examine patterns of variation across the entire genome in multiple human populations. Such an undertaking will facilitate an understanding of events that took place tens of thousands of years ago, and the better design of studies to map genes involved in human disease today.

4.1 Human Origins and Genetic Variation Background

Genetics is a powerful tool for uncovering the evolutionary history of our species. DNA is comprised of four building blocks: guanine (G), adenine (A), thymine (T), and cytosine (C). The sequence of the vast majority of these 3 billion building blocks (base pairs or nucleotides) is identical among humans. However, sprinkled throughout the genome are small differences in the DNA sequence known as polymorphisms. Most polymorphisms are simple substitutions of one base for another, called single nucleotide polymorphisms or SNPs. These kinds of polymorphisms are found at only about 1 out of 1000 nucleotide sites among humans. When many of these polymorphisms are compared among individuals from different human populations, they are informative for inferring the timing and order of branching events among ancestral populations, as well as the history of changes in population sizes. The signature of these historical processes, as well as natural selection, can be read from patterns of polymorphism in our genome.

DNA is found in nearly every cell of our body. Most cells have two compartments that contain DNA. Nuclear DNA is packaged in chromosomes found in the nucleus of the cell. Mitochondrial DNA (mtDNA) is a circular molecule found in the mitochondria in the cytoplasm of our cells. Humans have 23 pairs of chromosomes, with one chromosome of each pair inherited from our mother and the other from our father. The first 22 pairs of chromosomes are called the autosomes, while the 23rd pair is called the sex chromosomes, namely the X and Y chromosomes. Females are born with two X chromosomes while males are born with an X and a Y chromosome. The Y chromosome is inherited only from the fathers line, and because there is no shuffling of genetic material (known as recombination) between the Y and X chromosomes, most of the Y chromosome (the non-recombining portion of the Y chromosome or NRY) traces to only a single male in each previous generation. Similarly, mitochondrial DNA (mtDNA) is inherited only through females and does not undergo recombination, so our mtDNA traces back to a single mother in each previous generation. These non-recombining parts of the genome are called haploid.

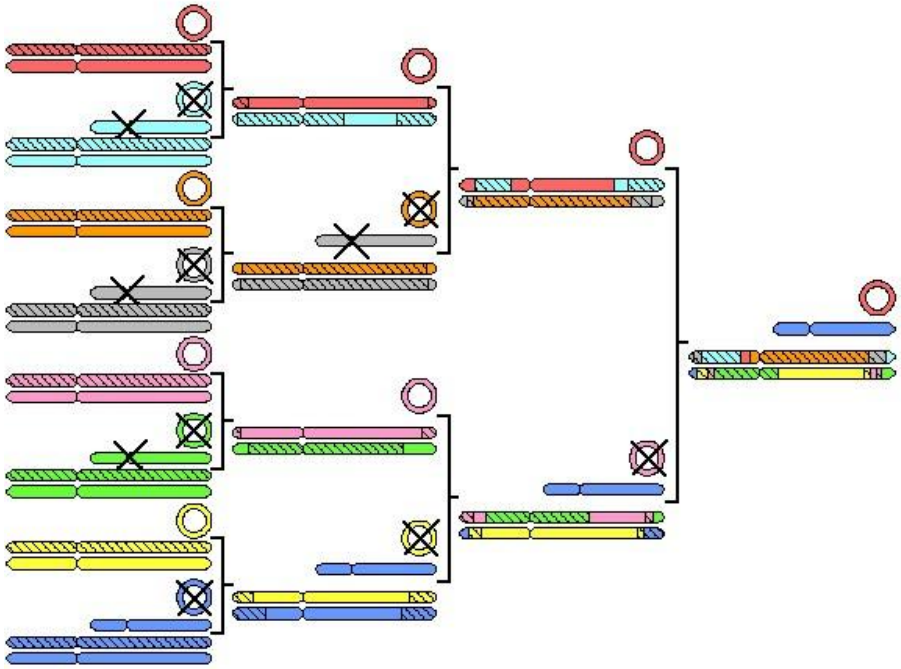


Figure 7: Inheritance patterns over four generations.

Inheritance patterns for mtDNA, the Y chromosome, and autosomes: Figure 7 shows the different inheritance patterns of the NRY, mtDNA, and autosomes for four generations (tiers) of a hypothetical population. On the left, four sets of parents mate, producing four offspring in the succeeding generation. These four individuals in turn mate to produce the two individuals in the third generation. Finally, these two individuals mate to produce the single male in the fourth generation. A large pair of chromosomes represents the 22 pairs of autosomes, a single small chromosome represents the NRY, and the circular symbol stands for mtDNA. Due to recombination, the autosomes in the fourth generation (rightmost) are a complex mixture of maternal and paternal sequences (indicated by different colors and patterns) traceable to multiple ancestors. The NRY and mtDNA in the fourth generation have only a single ancestor in the first generation. Y chromosomes and mtDNA molecules lost during genetic transmission between generations are indicated by crosses through their symbols.

Evolutionary processes differentially affect different parts of the genome: When making inferences about human history from DNA sequence data, it is important to take account of processes that differentially affect the haploid parts of the genome (NRY and mtDNA), the X chromosome, and autosomes. In addition to the different inheritance patterns among the four parts of the genome, there are differences in copy number, recombination rate, and mutation rate. While genes carried on the mtDNA or NRY are present in only a single copy in an individual, X-linked genes are present in two copies in females and only one copy in males (but can be transmitted by either sex). Autosomal genes are maintained on two chromosomes in both sexes. As a result, when equal numbers of males and females are breeding, the relative effective population size of the autosomes, X chromosome, NRY, and mtDNA is 4:3:1:1, respectively (see leftmost column in Fig. 7, which depicts 16 autosomes, 4 Y chromosomes, and 4 mtDNA molecule). The reduced effective population size of the haploid loci is expected to result in shallower times to the most recent common ancestor (TMRCA), higher levels of differentiation among human populations, and possibly smaller effects of natural selection. The larger effective population sizes of the X chromosome and autosomes means that loci in these genomic compartments are expected to have deeper ancestry, allowing inferences of evolutionary processes that took place well before the TMRCA of the haploid regions.

4.2 Research Problems in Population Genetics

These research problems deal with the utility of map visualization for answering specific population genetics questions, using an amazingly rich set of gene data.

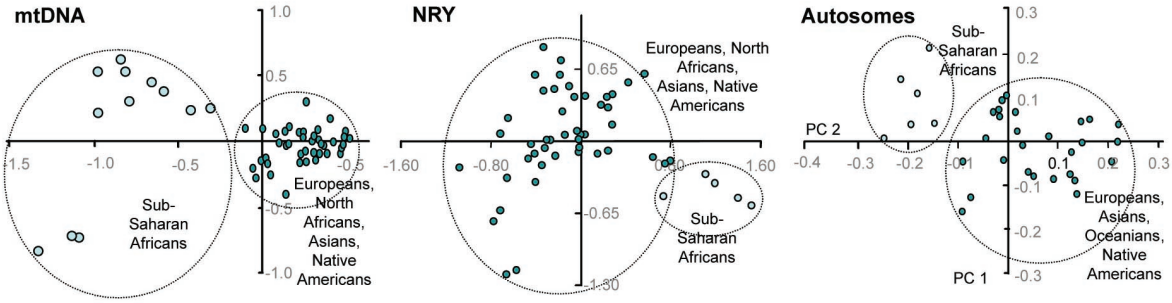


Figure 8: Scatterplots showing the relationships among human populations based on mtDNA, NRY, and autosomal data.

P10. Integration of Patterns across Loci: Until recently, questions about human prehistory have primarily been addressed with polymorphism data from mtDNA and the NRY. However, these parts of the genome make up only a small fraction of our total DNA. While the mtDNA and NRY each trace back to only a single ancestor each generation, the hundreds of thousands of independent segments comprising the remainder of our genome trace to many different ancestors in the past. While the patterns of genetic inheritance dictate that there is a single ancestor (and a single phylogenetic tree) for all non-recombining portions of our genome, each portion of our genome will not necessarily have the same history. In other words, a gene tree can tell us about the history of a single locus, but not about the structure of the whole population or about the forces that determine the spread of alleles at that locus. Due to chance effects and selection, a single gene tree may not accurately reflect the phylogenetic relationships of the populations studied. Since neither selection nor chance effects is expected to influence all loci equally, gene trees from independently segregating loci must be compared to test hypotheses about population history. It is only through an examination of many independent regions that we are in a stronger position to reconstruct the evolutionary history of our species. However, it is difficult to make inferences from patterns of variation in different parts of the genome. Figure 8 shows three scatterplots representing the relationships among human populations based on mtDNA, NRY, and autosomal data. These plots clearly show different patterns. Currently, there is not a single methodology that allows inference across these systems. Using the map representation would allow us to put together these different relationships defined on the same set of objects, while allowing the interactive exploration of the different ways to combine them.

The different relationships come from a variety of markers which will be utilized in this proposal. The kind of marker will vary according to the part of the genome being studied. For example, two different sets of markers on the NRY will be employed. The first kind of marker, known as a short tandem repeat or STR, has a high mutation rate and exhibits multiple alleles in human populations. A total of 38 Y-STRs will be typed and the combination of allelic states at all 38 STR markers is called a haplotype. The second kind of marker, already referred to above, is called a SNP. SNPs have slow mutation rates and typically exhibit only two allelic states in human populations. The combination of allelic states at all 85 SNPs on a single Y chromosome is called a haplogroup. SNPs will also be typed on the autosomes and mtDNA, albeit by different methodologies. For mtDNA, SNP genotyping will be performed by direct DNA sequencing of a given region after PCR amplification, or by use of a variety of PCR-based SNP-typing methods such as allelic-specific PCR, Taqman assays, or Sequenome MassARRAY (which will also be used for NRY genotyping). The latest microarray-based methods to genotype large numbers of SNPs (e.g., 50,000-1 million SNPs) in a single reaction will be employed for autosomal SNP typing.

P11. Assigning Ancestry: In addition to the allelic states at all the different kinds of markers for each genetic system (NRY, mtDNA, and autosomes), the data for each sample in the database will include labels for the population name, geographic region, ethnicity, and language affiliation. The incorporation of a geo-spatial visualization framework facilitates ease of use and rapid interpretation of analytical results. A two-pronged strategy will be employed for assigning each sample to a NRY haplogroup. First, the Y-STR data are used to predict the Y-SNP haplogroup of each sample. Given a large data set of marker scores correctly labeled with haplogroups, a function is learned to map, or classify, new marker scores to a haplogroup. After the Y-SNP haplogroup of each sample is predicted, each sample is typed with the appropriate Y-SNP to confirm its haplogroup. Accuracy of haplogroup prediction is typically 95-99%, confirming the high quality of the Y-STR data.

The interactive map representation described in this proposal will be used to predict haplogroups, as well as the

geographic or ethnic background of samples. Comparisons will be made with machine learning methods, which are also under development to assign genotypes to geographic/ethnic groups and haplogroup prediction. One advantage of our proposed methods will be their ability to co-model NRY, mtDNA, and autosomal data, which has been a challenge with other approaches. Because of the unique mode of inheritance of the two sex-specific regions and the multiple classes of polymorphic markers they contain, the NRY and mtDNA offer the opportunity to infer individual ancestry with more geographical and temporal resolution than similar sized sets of markers from other parts of the genome. The uniparental inheritance and the lack of recombination mean that mutations accumulating on the NRY and mtDNA provide a nested series of markers for tracing ancestry to common ancestors at various points in the past. The range of mutation rates associated with different classes of polymorphism on both the NRY and mtDNA means that both very recent and distant ancestry can be reconstructed. While the identification of geographic population structure and genetic ancestry on the basis of a minimal set of markers is desirable from a cost perspective, the absence of sharp discontinuities in the neutral genetic diversity among human populations implies that, in practice, a large number of neutral markers will be required to identify the genetic ancestry of one individual. This means that thousands of autosomal SNPs will be needed, and hence, inference will come from the integration of signal from different data sets. Many recent studies have shown that a large set of autosomal SNPs is able to recover the genetic ancestry of individuals from different continents, and even from within continents.

5 Integration of Research and Education

The proposed research component is integrated with an *educational component* aiming to involve both graduate and undergraduate students through project-oriented courses. Building on prior teaching experience and published work in the area of computer science education [7, 8, 18–20], we focus on graduate student advising, and the involvement of graduates and undergraduates in research. While graduate student involvement is central to this proposal, we provide more details about the more challenging plan to involve undergraduates.

The impact of an undergraduate's research experience, both for the student and the faculty supervisor, increases directly with the length of research involvement. This is due both to the cultural and technical learning curves involved, which take an investment of time and energy from both parties; and to the fact that a student who engages in research over a longer period of time can contribute more to, and benefit more from, the research project. Therefore, the benefits of undergraduate research, for both the students and the supervising faculty members, are greatest when students become engaged in research early in their student careers. However, a pragmatic problem arises here: a typical college freshman or sophomore has little technical knowledge, little idea of what research involves, whether or not it is interesting, and how to go about finding a suitable research project to work on. Unfortunately, by the time undergraduates figure out the answers to these questions, they generally do not have enough time left in their college careers to allow for a significant length of involvement in research.

To address these problems and create an environment where undergraduates can be systematically exposed to the research process and given opportunities to become involved with research projects, we propose a gradual model for undergraduate research. The idea is to begin by exposing students to ongoing research projects and activities as early as their sophomore year; then have them gain experience with more advanced topics and research in project-oriented classes in their junior year; and, finally, integrate them into active research groups, and have them independently carrying out research activities by their senior year.

The undergraduate projects would be designed to meet the following criteria: (1) a project should not require overly sophisticated technical background to get started; (2) the eventual research goals should be clearly defined and should be attainable via a well-defined series of small steps; and (3) there should be groups of related projects, while keeping each individual project sufficiently independent. While these criteria are difficult to meet, careful planning and advance preparation can lead to projects that are both appealing to the students and worthwhile to pursue. In particular, implementation of new algorithms and applications of generic techniques to specific problems seem to make great undergraduate projects.

Groups of students working on related research problems in a cooperative and collaborative manner foster a sense of community. This gives students a sense of engagement, belonging, and ownership in this body of knowledge and makes the work more real. The PI began working on real research problems with undergraduates as a part of an NSF CAREER grant and more than a dozen of our publications have at least one undergraduate co-author [12, 14–18, 25, 30, 31, 34, 45, 64]. Since then several other faculty members have begun involving undergraduates in their research. With the help of exciting, visually appealing projects likely to result from the proposed research, we hope to continue this trend and make it a permanent feature of our undergraduate program.

6 Results from Prior NSF Support

PI Stephen Kobourov: Kobourov is PI on NSF grant ACR-022920, entitled *Visualization of Giga-Graphs and Graph Processes*, for the period September 2002 through August 2005, $240,358. This project has resulted in a number of publications on visualization of large graphs and graphs that evolve through time [6, 17, 25, 30, 32, 33, 35, 45] and several software systems for graph visualization, including GraphAEL [31] (for visualization of computing literature) and GMorph [36] (for intersection-free morphing of planar graphs). Many students have been involved in this research, both of the graduate and undergraduate levels. Three PhD students completed PhD theses on work partly funded by this grant:

1. Cesim Erten co-authored ten papers [6, 27, 30–34, 36, 37, 58] and completed a PhD on this topic in 2004 [29].
2. Joe Fowler co-authored 8 papers [5, 39–41, 47–50] and completed a PhD in 2009 [46].
3. Alejandro Estrella-Balderrama co-authored 8 papers [4, 11, 39–44] and completed a PhD in 2009 [38].

Four MS students were co-authors on papers related to the project [2, 12, 17, 30, 31, 36, 37, 45, 65, 66]. More than a dozen of our publications have at least one undergraduate co-author [12, 14–18, 25, 30, 31, 34, 45, 64]. Undergraduates who have worked on this project have won several awards:

- Gary Yee: Outstanding Senior, UA Department of Computer Science, 2004. Outstanding Senior, UA College of Science, 2004. CRA Outstanding Undergraduate Award, 2004.
- Kyriakos Pavlou: Outstanding Senior, UA Dept. of Computer Science, 2005. Outstanding Senior, UA College of Science, Fall 2005;
- Kevin Wampler: Galileo Circle Award, UA College of Science, 2005
- Anand Iyer: Outstanding Senior, UA Dept. of Computer Science, 2006;
- David Forester: Outstanding Senior, UA Dept. of Computer Science, 2007;

co-PI Michael Hammer: Hammer is co-PI on NSF grant 0725470: Anthropological Modeling of Social Structure, Genetics, and Language Speciation in Indonesia; for the period August 2007 through July 2010, $1,247,928. The main goal of this grant has been to build and test anthropological models to explain observed patterns of genetic and linguistic variation at the levels at which they originate. While most studies of genetic and linguistic evolution and differentiation have focused on large-scale regional or continental patterns, our approach is to gather information at the community level to address community based, island based and region based questions. In collaboration with Indonesian researchers and public health teams, we collected genetic, medical, environmental and ethnographic data from 69 villages on 13 Indonesian islands, as well as word lists and phonological samples from nearly a thousand languages. All of these data, as well as data on the prevalence of six diseases, is now assembled in or linked to a geographic information system. A combination of modeling and inferential approaches has been developed to investigate the processes under study. Many publications resulted on the origins and demographic history of Indonesian populations, and the social factors that shape patterns of genetic variation at the community scale [21, 22, 56, 62, 68–70, 80, 81, 83].

References

[1] Map of online communities. http://xkcd.com/256.

[2] J. Abello, S. G. Kobourov, and R. Yusufov. Visualizing large graphs with compound-fisheye views and treemaps. In *12th Symposium on Graph Drawing (GD)*. To appear in 2005.

[3] T. Ball, S. Diehl, D. Notkin, and A. Zeller. Dagstuhl seminar #05261: *Multi-Version Program Analysis*, June 2005.

[4] C. Binucci, E. D. Diacomo, W. Didimo, A. Estrella-Balderrama, F. Frati, S. G. Kobourov, and G. Liotta. Directed graphs with an upward straight-line embedding into every point set. In *21th Canadian Conference on Computational Geometry (CCCG)*. To appear in 2009.

[5] U. Brandes, C. Erten, J. J. Fowler, F. Frati, M. Geyer, C. Gutwenger, S.-H. Hong, M. Kaufmann, S. G. Kobourov, G. Liotta, P. Mutzel, and A. Symvonis. Colored simultaneous geometric embeddings. In *13th Conference on Computing and Combinatorics (COCOON)*, pages 254–263, 2007.

[6] P. Brass, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous graph embedding. *Computational Geometry: Theory and Applications*, 36(2):117–130, 2007.

[7] S. Bridgeman, M. T. Goodrich, S. G. Kobourov, and R. Tamassia. PILOT: An interactive tool for learning and grading. In *Proceedings of the 31st Technical Symposium on Computer Science Education (SIGCSE 2000)*, pages 139–143, 2000.

[8] S. Bridgeman, M. T. Goodrich, S. G. Kobourov, and R. Tamassia. SAIL: A system for generating, archiving, and retrieving specialized assignments using LaTex. In *Proceedings of the 31st Technical Symposium on Computer Science Education (SIGCSE 2000)*, pages 300–304, 2000.

[9] G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics*, 6(2):214–229, May 1993.

[10] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Press, 1999.

[11] J. Cappos, A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Simultaneous graph embedding with bends and circular arcs. *Comput. Geom.*, 42(2):173–182, 2009.

[12] J. Cappos, S. G. Kobourov, M. Miles, M. Stepp, K. Pavlou, and A. Wixted. Collaboration with diamondtouch. In *10th International Conference on Human Computer Interaction (INTERACT)*, pages 986–989, 2005.

[13] F. Chung. Spectral graph theory, 1997.

[14] C. Collberg, S. G. Kobourov, E. Carter, and C. Thomborson. Error-correcting graphs for software watermarking. In *29th Workshop on Graph Theoretic Concepts in Computer Science*, pages 156–167, 2003.

[15] C. Collberg, S. G. Kobourov, S. Kobes, B. Smith, S. Trush, and G. Yee. Tetratetris: An application of multi-user touch-based human-computer interaction. In *9th International Conference on Human-Computer Interaction (INTERACT)*, pages 81–88, 2003.

[16] C. Collberg, S. G. Kobourov, J. Louie, and T. Slattery. SPLAT: A system for self-plagiarism detection. In *Proceedings of the IADIS Conference WWW/Internet*, pages 508–514, 2003.

[17] C. Collberg, S. G. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *ACM Symposium on Software Visualization (SoftVis)*, pages 77–86, 2003.

[18] C. Collberg, S. G. Kobourov, and S. Westbrook. Algovista: A tool to enhance algorithm design and understanding. In *7th Symposium on Innovation and Technology in Computer Science Education (ITICSE)*, pages 228–237, 2002.

[19] C. S. Collberg, S. Debray, S. G. Kobourov, and S. Westbrook. 'increasing undergraduate involvement in computer science research. In *8th World Conference on Computers in Education (WCCE)*, pages 342–352, 2005.

[20] C. S. Collberg, S. G. Kobourov, and S. Westbrook. Algovista: an algorithmic search tool in an educational setting. In *35th Symposium on Computer Science Education (SIGCSE)*, pages 462–466, 2004.

[21] M. P. Cox, A. J. Redd, T. M. Karafet, C. A. Ponder, J. S. Lansing, H. Sudoyo, and M. F. Hammer. A polynesian motif on the y-chromosome. *Human Biology*, 79:525–535, 2007.

[22] M. P. Cox, A. Woerner, J. D. Wall, and M. F. Hammer. Intergenic dna sequences from the human x chromosome reveal high rates of global gene flow. *BMC Genetics*, 9:76–87, 2008.

[23] M. B. Dillencourt, D. Eppstein, and M. T. Goodrich. Choosing colors for geometric graphs via color space embeddings. In *14th Symposium on Graph Drawing (GD)*, pages 294–305, 2006.

[24] C. A. Duncan, D. Eppstein, and S. G. Kobourov. The geometric thickness of low degree graphs. In *20th Annual ACM-SIAM Symposium on Computational Geometry (SCG)*, pages 340–346, 2004.

[25] B. Dux, A. Iyer, S. Debray, D. Forrester, and S. G. Kobourov. Visualizing the behaviour of dynamically modifiable code. In *13th IEEE Workshop on Porgram Comprehension*, pages 337–340, 2005.

[26] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[27] A. Efrat, C. Erten, and S. G. Kobourov. Fixed-location circular-arc drawing of planar graphs. In *11th Symposium on Graph Drawing*, pages 147–158, 2003.

[28] J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz - open source graph drawing tools. In *Graph Drawing*, pages 483–484, 2001.

[29] C. Erten. *Simultaneous Embedding and Visualization of Graphs*. PhD thesis, University of Arizona, 2004.

[30] C. Erten, P. J. Harding, S. Kobourov, K. Wampler, and G. Yee. Exploring the computing literature using temporal graph visualization. In *Visualization and Data Analysis*, pages 45–56, 2004.

[31] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee. GraphAEL: Graph animations with evolving layouts. In *11th Symposium on Graph Drawing*, pages 98–110, 2003.

[32] C. Erten and S. G. Kobourov. Simultaneous embedding of a planar graph and its dual on the grid. *Theory of Computing Systems*, 38(3):313–327, 2005.

[33] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *Journal of Graph Algorithms and Applications*, 9(3):347–364, 2005.

[34] C. Erten, S. G. Kobourov, A. Navabia, and V. Le. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *11th Symposium on Graph Drawing (GD)*, pages 437–449, 2003.

[35] C. Erten, S. G. Kobourov, A. Navabia, and V. Le. Simultaneous graph drawing: Layout algorithms and visualization schemes. *Journal of Graph Algorithms and Applications*, 9(1):165–182, 2005.

[36] C. Erten, S. G. Kobourov, and C. Pitta. Intersection-free morphing of planar graphs. In *11th Symposium on Graph Drawing*, pages 320–331, 2003.

[37] C. Erten, S. G. Kobourov, and C. Pitta. Morphing planar graphs. In *20th ACM Symposium on Computational Geometry*, 2004. To appear in 2004.

[38] A. Estrella-Balderrama. *Simultaneous Embedding and Level Planarity*. PhD thesis, University of Arizona, 2009.

[39] A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Colored simultaneous geometric embeddings and universal pointsets. In *21th Canadian Conference on Computational Geometry (CCCG)*. Accepted, to appear in 2009.

[40] A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. On the characterization of level planar trees by minimal patterns. In *17th Symposium on Graph Drawing (GD)*. To appear in 2009.

[41] A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Characterization of unlabeled level planar trees. In *14th Symposium on Graph Drawing (GD)*, pages 367–379, 2006.

[42] A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Graph simultaneous embedding tool, GraphSET. In *16th Symposium on Graph Drawing (GD)*, pages 169–180, 2008.

[43] A. Estrella-Balderrama, F. Frati, and S. G. Kobourov. Upward straight-line embeddings of directed graphs into point sets. In *34th Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 122–133, 2008.

[44] A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In *15th Symposium on Graph Drawing (GD)*, pages 280–290, 2007.

[45] D. Forrester, S. G. Kobourov, A. Navabi, K. Wampler, and G. Yee. graphael: A system for generalized force-directed layouts. In *12th Symposium on Graph Drawing (GD)*. To appear in 2004.

[46] J. Fowler. *Unlabeled Level Planarity*. PhD thesis, University of Arizona, 2009.

[47] J. J. Fowler, C. Gutwenger, M. Jünger, P. Mutzel, and M. Schulz. An spqr-tree approach to decide special cases of simultaneous embedding with fixed edges. In *16th Symposium on Graph Drawing (GD)*, pages 157–168, 2008.

[48] J. J. Fowler, M. Jünger, S. G. Kobourov, and M. Schulz. Characterizations of restricted pairs of planar graphs allowing simultaneous embedding with fixed edges. In *34th Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 146–158, 2008.

[49] J. J. Fowler and S. G. Kobourov. Characterization of unlabeled level planar graphs. In *15th Symposium on Graph Drawing (GD)*, pages 37–49, 2007.

[50] J. J. Fowler and S. G. Kobourov. Minimum level nonplanar patterns for trees. In *15th Symposium on Graph Drawing (GD)*, pages 69–75, 2007.

[51] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Softw. – Pract. Exp.*, 21(11):1129–1164, 1991.

[52] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force directed placement. *Software - Practice and Experience*, 21:1129–1164, 1991.

[53] P. Gajer, M. T. Goodrich, and S. G. Kobourov. A fast multi-dimensional algorithm for drawing large graphs. *Computational Geometry: Theory and Applications*, 29(1):3–18, 2004.

[54] P. Gajer and S. G. Kobourov. GRIP: Graph dRawing with Intelligent Placement. *Journal of Graph Algorithms and Applications*, 6(3):203–224, 2002.

[55] E. Gansner, Y. Hu, S. Kobourov, and C. Volinsky. Putting recommendations on the map - visualizing clusters and relations. In *Proc. 3rd ACM Conference on Recommender Systems*, pages 345–354, 2009.

[56] D. W. Garrigan and M. F. Hammer. Reconstructing human origins in the genomic era. *Nature Reviews Genetics*, 7:669–680, 2006.

[57] Y. F. Hu and J. A. Scott. A multilevel algorithm for wavefront reduction. *SIAM Journal on Scientific Computing*, 23:1352–1375, 2001.

[58] A. Iyer, A. Efrat, C. Erten, D. Forrester, and S. G. Kobourov. A force-directed approach to sensor localization. In *13th Symposium on Graph Drawing (GD)*. To appear in 2005.

[59] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32:241–254, 1967.

[60] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[61] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inform. Process. Lett.*, 31:7–15, 1989.

[62] T. M. Karafet, J. S. Lansing, A. J. Redd, J. C. Watkins, I. W. Ardika, S. P. K. Surata, L. Mayer, M. Bamshad, L. Jorde, and M. F. Hammer. A balinese y chromosome perspective on the peopling of indonesia: Genetic contributions from pre-neolithic hunter-gatherers, austronesian farmers, and indian traders. *Human Biology*, 77:93–114, 2005.

[63] S. Kobourov, P. Mutzel, and M. Junger. Dagstuhl seminar #05191: *Graph Drawing*, May 2005.

[64] S. G. Kobourov, A. Efrat, D. Forrester, and A. Iyer. Force-directed approaches to sensor network localization. In *8th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 108–118, 2006.

[65] S. G. Kobourov and C. Pitta. An interactive multi-user system for simultaneous graph drawing. In *12th Symposium on Graph Drawing (GD)*. To appear in 2004.

[66] S. G. Kobourov and K. Wampler. Non-Euclidean spring embedders. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):757–767, 2005.

[67] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Press, 1978.

[68] J. S. Lansing, M. P. Cox, S. S. Downey, B. Hallmark, T. M. Karafet, P. Norquest, B. M. Gabler, J. W. Schoenfelder, H. Sudoyo, J. C. Watkins, and M. F. Hammer. Coevolution of languages and genes on the island of sumba, eastern indonesia. *Proceeding of the National Academy of Sciences*, 104:16022–16026, 2007.

[69] J. S. Lansing, T. M. Karafet, J. W. Schoenfelder, and M. F. Hammer. A dna signature for the expansion of irrigation in bali. *Past Human Migrations in East Asia and Taiwan: Matching Archaeology, Linguistics and Genetics*, pages 374–394, 2008.

[70] J. S. Lansing, J. C. Watkins, B. Hallmark, M. P. Cox, T. M. Karafet, H. Sudoyo, and M. F. Hammer. Male dominance rarely skews the frequency distribution of y chromosome haplotypes in human populations. *Proceeding of the National Academy of Sciences*, 105:11645–11650, 2008.

[71] S. Lloyd. Last square quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[72] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103:8577–8582, 2006.

[73] A. Noack. Modularity clustering is force-directed layout. *Physical Review E*, 79, 2009.

[74] E. Raisz. The rectangular statistical cartogram. *Geographical Review*, 24(2):292–296, 1934.

[75] B. Shneiderman. Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[76] F. M. a. d. H. St. Leonardi and D. Wagner. Dagstuhl seminar #05361: *Algorithmic Aspects of Large and Complex Networks*, September 2005.

[77] J. R. R. Tolkien. *The Shaping of Middle-Earth*. Houghton Mifflin Harcourt, 1986.

[78] W. T. Tutte. How to draw a graph. *Proc. London Math. Society*, 13(52):743–768, 1963.

[79] F. van Ham and B. E. Rogowitz. Perceptual Organization in User-Generated Graph Layouts. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, 14(6), NOV 2008.

[80] J. D. Wall, M. P. Cox, F. L. Mendez, A. Woerner, T. Severson, and M. F. Hammer. A novel dna sequence database for analyzing human demographic history. *Genome Research*, 18:1354–1361, 2008.

[81] J. D. Wall and M. F. Hammer. Archaic admixture in the human genome. *Current Opinion in Genetics and Development*, 16:606–610, 2006.

[82] R. Wiese, M. Eiglsperger, and M. Kaufmann. yfiles: Visualization and automatic layout of graphs. In *Graph Drawing*, pages 453–454, 2001.

[83] A. Woerner, M. P. Cox, and M. F. Hammer. Recombination-filtered genomic datasets by information maximization. *Bioinformatics*, 23:1851–1853, 2007.