Egocentric Analysis of Dynamic Networks with EgoLines

Jian Zhao[1] **Michael Glueck**[1] **Fanny Chevalier**[2] **Yanhong Wu**[3] **Azam Khan**[1]
[1]Autodesk Research [2]Inria [3]Hong Kong University of Science and Technology
{jian.zhao, michael.glueck, azam.khan}@autodesk.com, fanny.chevalier@inria.fr, yanhong.wu@ust.hk

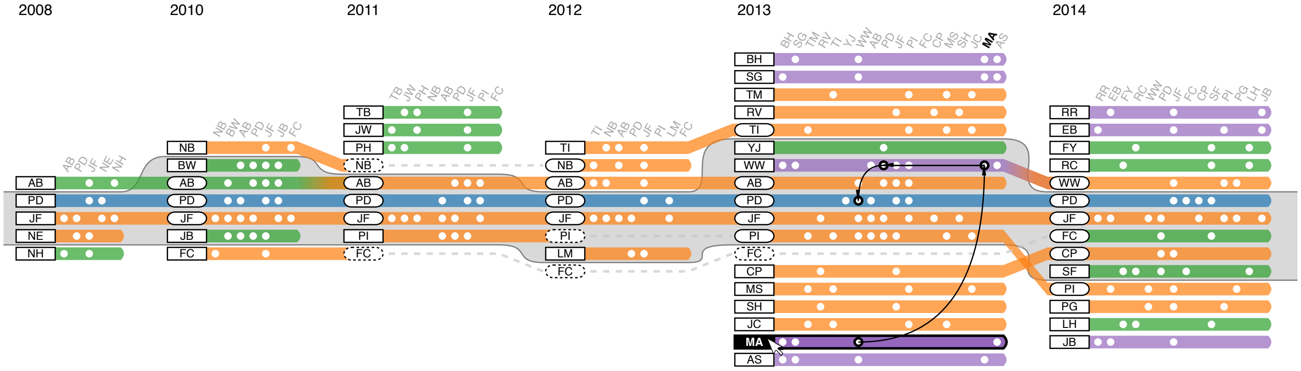


Figure 1. EgoLines supports the investigation of temporal patterns in dynamic ego-networks. Here, a 2-level ego-network of academic collaborations is shown for the author P. Dragicevic (PD). Collaborations between the author, his co-authors, and their co-authors are displayed, indicating how PD interacts with the other authors during his academic career. Using a *subway map* metaphor, authors are shown as *actor lines* across time steps (years; data absent for 2009). Colors indicate clusters of co-authors who collaborated more frequently with one another. Actor lines are tightly packed to create blocks of line segments at each time step akin to adjacency matrices. Authors directly connected to PD are indicated using a light-gray convex hull, similar to *fare zones* in a subway map. For 2013, the shortest path between the hovered-over author (MA) and PD is traced using curved arrows, revealing that WW is the connection between them.

ABSTRACT

The egocentric analysis of dynamic networks focuses on discovering the temporal patterns of a subnetwork around a specific central actor (i.e., an ego-network). These types of analyses are useful in many application domains, such as social science and business intelligence, providing insights about how the central actor interacts with the outside world. We present EgoLines, an interactive visualization to support the egocentric analysis of dynamic networks. Using a "subway map" metaphor, a user can trace an individual actor over the evolution of the ego-network. The design of EgoLines is grounded in a set of key analytical questions pertinent to egocentric analysis, derived from our interviews with three domain experts and general network analysis tasks. We demonstrate the effectiveness of EgoLines in egocentric analysis tasks through a controlled experiment with 18 participants and a use-case developed with a domain expert.

Author Keywords

Dynamic network; egocentric network; graph visualization.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces

CHI'16, May 07–12, 2016, San Jose, CA, USA.

ACM 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858488

INTRODUCTION

A network is a ubiquitous data structure found in a range of application domains that can be used to describe concepts such as social networks, mobile device connections, and neural pathways. Many of these networks are dynamic, i.e., the topology of a network and/or the attributes of its nodes and links vary over time, revealing relationship dynamics in real-world systems. Information visualization techniques have been shown effective in many scenarios, helping people understand how these networks change over time [7]. One key method of dynamic network analysis uses an *egocentric* approach. In contrast to whole-network analysis, egocentric analysis focuses on the local subnetwork around a particular node, the *ego*, and its surrounding neighbors, the *alters* [29]. The ego is the central actor of interest in a particular domain (e.g., an individual, a device, or a synapse). This subnetwork is called an *ego-network* and its boundary is defined in terms of *levels*. For example, a 1-level ego-network includes only alters directly connected to the ego, while a 2-level ego-network includes all alters within a path distance of two, and all connections between them. In practice, only 1-level and 2-level ego-networks are typically considered [29].

The temporal dynamics of ego-networks can provide insight into how an ego affects, or is affected by alters over time. For example, medical experts have shown that an individual's health is strongly associated with many social factors (e.g., number of friends) [27]; analysts in management and business intelligence have made informed decisions about marketing strategies by identifying and observing the most influential people in social networks [15]; and computer networking researchers have enhanced situational awareness by tracing and studying the context of specific devices in a mobile ubiquitous system [8]. Each of these insights can be demonstrated using dynamic ego-network visualizations.

However, most dynamic network visualizations are designed for analyzing the network as a whole, i.e., at the macro-level (e.g., [6, 10, 21]), making them ill-suited to egocentric analysis tasks that are at the micro-level. While one could partition network data into multiple ego-networks and visualize each individually, it is difficult to switch between different ego-networks and perform in-depth comparisons. Moreover, macro-level visualizations mainly focus on tracking changes of the entire network rather than characteristics of ego-networks. Thus, some egocentric analytical questions are cumbersome to answer, for example: how do specific ego-alter relationships (e.g., alter levels) change over time; how do alter communities evolve (e.g., splitting and merging); what is the stability of 1-level or 2-level ego-networks?

In this paper, we present EgoLines, an interactive visualization system that supports egocentric analysis of dynamic networks. As shown in Figure 1, we extract and encode the 2-level ego-network of a particular actor from a dynamic network dataset using a *subway map* metaphor. Much like a subway map represents individual routes as lines (with rectangles and circles indicating major and minor stops), each actor in the ego-network is represented as an *actor line* traveling across time steps. Actor lines are carefully packed together to minimize crossings, whilst revealing the network topology at each time step in a compact block, akin to an adjacency matrix. Compared to the basic approach that shows a dynamic network with a series of matrices (e.g., [6, 4]), EgoLines allows users to better track individual actors along time (e.g., joining/leaving the network, and switching between communities). EgoLines also provides an overview that shows the entire network aggregated across all time steps, as well as a tabular view with rows displaying the major characteristics of each extracted ego-network (e.g., network sizes and densities). Rich interactions, such as linking, filtering, and reordering, are incorporated to assist data exploration in the main visualization (Figure 1), as well as facilitate coordination across different visualization views.

We distill important egocentric analysis tasks of dynamic networks from our interviews with three domain experts and task taxonomies of static and dynamic graph analyses in the literature [25, 38]. The design of EgoLines is grounded in these tasks. We conducted a controlled experiment to compare EgoLines with two alternatives: a node-link variation of EgoLines similar to [19] and a traditional small-multiple representation (a series of node-link diagrams). Additionally, a use-case scenario was developed with a domain expert to demonstrate the usage of EgoLines in performing an egocentric analysis of email communications within an organization.

RELATED WORK

A large body of research has focused on visualizing networks. Node-link diagrams and adjacency matrices are the two most common methods for representing static networks. Node-link diagrams are better suited to topological tasks, since edges are explicitly encoded, but suffer from increased visual occlusion when the network becomes larger and denser [23]. While adjacency matrices have been shown effective and more scalable in many graph analysis tasks [18], the abstract encoding of topology makes it difficult to trace paths between nodes.

Node-link diagrams and adjacency matrices have been extended to visualize dynamic networks (see [7] for a comprehensive survey). In general, there are three main approaches: animation-based, timeline-based, and a hybrid of the two.

Animation-based techniques were first used to show transitions across individual snapshots of dynamic networks in node-link diagrams. Staged animations, i.e., dividing the animation into several steps such as (i) fading-out removed elements, (ii) transforming topology, then (iii) fading-in new elements, are widely employed to reduce the effort required in tracking changes between time steps [5, 17]. Highlighting key events (e.g., node or link removal) can also be incorporated into the staged animations to ease the understanding of changes [5]. Showing dynamic networks amongst a sequence of animations, however, could impose a high cognitive load on users, forcing them to remember the network at different time steps [3]. This becomes more pronounced when viewing changes over longer periods of time.

In timeline-based approaches, one method of extending node-link diagrams is to show each time step as a vertical line of nodes with arcs indicating links between them [19]. Similarly, parallel edge splatting draws links between two consecutive lines of nodes and reduces visual clutter using a pixel-based method [10]. Alternatively, nodes can be positioned using a circular layout to display time steps on concentric rings [35]. Although these techniques scale well for large numbers of time steps, they do not visualize the network topology well due to the restrictive positioning of nodes. Adjacency matrices have also been used with timeline-based techniques. For example, MultiPiles represents a dynamic network as a series of matrices, where each matrix could be a single time step or an aggregation of several time steps [4]. Cubix employs a 3D cube metaphor to encode time in a third dimension, and users can interactively spread out slices of networks [6]. Organizing the matrices in a zigzag manner has also been proposed [41]. Unlike these techniques, EgoLines is more space efficient because only the subset of relevant actors is shown at each time step. Alternatively, Brandes et al. encoded temporal information within each matrix cell using Gestalt-lines to reveal bi-directed edge weight changes [9].

There are a few visualizations using the hybrid approach. For example, DiffAni allows users to divide the whole network sequence into several aggregation views to show time steps with differences, animations, and small multiples [31]. Hadlak et al. proposed an in-situ exploration of dynamic networks that combined many different visualization techniques such as node-link diagrams and adjacency matrices [21].

However, all the above techniques are designed for visualizing temporal variations over an entire network. The nature of egocentric analysis is to browse specific subnetworks, where the primary focus of the analyst is on the dynamics between the individual ego and their alters rather than the overall topology. Only a few projects have investigated visualizing dynamic ego-networks, adopting a radial layout where alters are positioned around the ego and the temporal relationship between the ego and an alter is encoded along the radius [14, 30]. These visualizations are inappropriate to show 2-level ego-networks, and alter-alter connections are either missing or difficult to reveal. This makes certain egocentric analyses difficult to perform, such as determining where the ego-alter relationships come from (e.g., one alter introduced by another alter) or how 2-level alters become 1-level (e.g., sharing more common friends with the ego) [20]. In contrast to these techniques, EgoLines better supports such tasks by showing all connections in dynamic 2-level ego-networks.

Unlike the radial layout, Shi et al. proposed a timeline-based method to show an aggregated ego-network across time steps in 1.5D, by drawing node-link diagrams of alters along the ego's timeline [33]. However, this makes it difficult to track specific ego-alter relationships, such as when an alter leaves or rejoins the network, due to edge crossings. Along the same line, egoSlider uses glyphs to facilitate the overall comparison of dynamic ego-networks, but it fails to reveal the network topologies [39]. ManyNets, which displays decomposed subnetworks in a table, may be adapted to egocentric analysis [16], but it does not support temporal information.

Our work is also related to techniques using lines to show temporal patterns. StoryFlow [26] and StoryLine [34] visualize entities in a story as timeline paths to illustrate their dynamic interactions. A similar design was used in TimeNets for genealogical data [24]. NeuroLines shows branching patterns in synaptic pathways of human brains using a subway map metaphor [1]. However, these techniques are not suitable to show dense and dynamic connections among different line entities, which is required for network analysis.

ANALYTICAL QUESTIONS

To better understand the egocentric analysis of dynamic networks, we carried out interviews with three domain experts. Two of them were computer scientists focusing on graph mining techniques, and one was a management school professor specializing on social network analysis. With the help of the experts, we familiarized ourselves with the background and high-level questions of ego-network analysis. Their focuses were investigating the temporal evolutions of ego-alter and alter-alter relationships and of topological change patterns. Based on these insights, we found there existed some overlap with the general network analysis tasks discussed in the literature [25, 38]. Thus, to concretize the tasks, we derived a set of ego-network analytical questions in corresponding to the task lists in [25, 38]. Next, we conducted another round of interviews with the experts to validate the tasks. The consolidated and revised analytical questions are as follows.

A. On dynamic ego-network evolutions

A1. Birth, death, and recurrence: How long is the lifespan of an alter in the ego-network? When do alters join, leave, and rejoin the ego-network?

A2. Convergence and divergence: Do several alter clusters (i.e., communities of alters) converge into a single cluster? Does an alter cluster diverge into multiple clusters?

A3. Stability and replacement: Are alters in the ego-network stable? Are they frequently replaced by new alters? How often do alters come and go?

A4. Attributes trending: Does the overall size of the ego-network grow or contract over time? How about the numbers of 1-level and 2-level alters? Does a specific attribute value (e.g., betweenness centrality [29]) of the ego or an alter increase or decrease? Are there peaks and valleys in these trends?

B. On specific ego-network time steps

B1. Adjacency and accessibility: Who is directly connected to the ego or an alter? What are the shortest paths connecting the ego and an alter, or two different alters?

B2. Connectivity: Who are the common connections of the ego and an alter, or two different alters? What clusters of alters exist in the ego-network? Who are the bridges between clusters?

B3. Accessing attributes: What is a specific attribute value (e.g., betweenness centrality) of the ego or an alter? What about attributes on a connection?

C. On the whole network

C1. Overview, focus, and context: What does the entire network look like? Where is the current ego-network with respect to the entire the network? What other ego-networks are near the current ego-networks? How about the above questions at a specific time step?

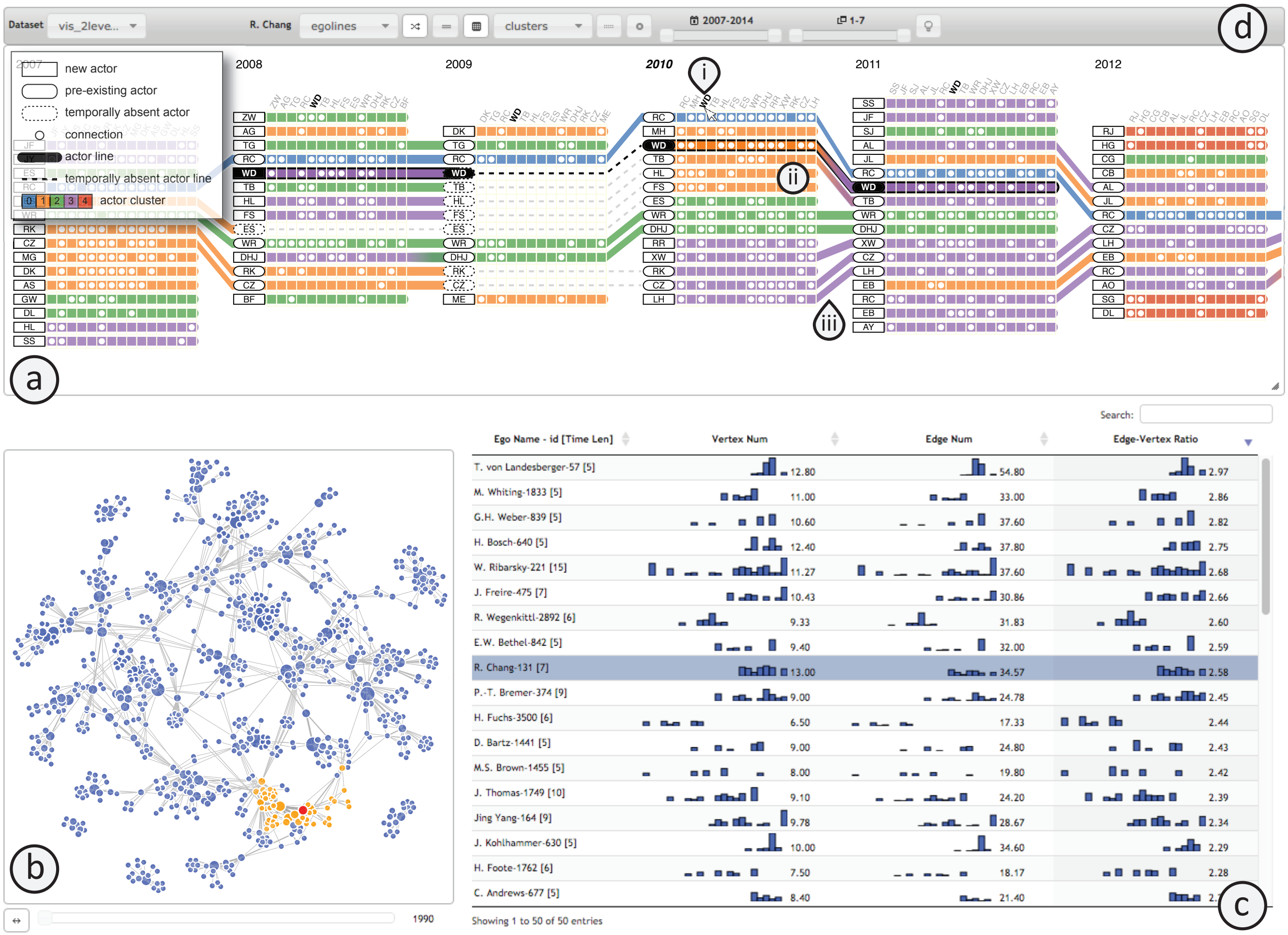
C2. Distributions: What are the overall temporal distributions of certain metrics (e.g., the number of actors) of an ego-network? What do the distributions look like across all ego-networks extracted from the whole network? Are there similarities between two ego-networks in terms of these metrics?

EGOLINES

Here, we describe the design of EgoLines that was guided by the aforementioned analytical questions. The EgoLines interface consists of three interactively coordinated views (Figure 2): (a) a main view showing the selected ego-network, (b) an overview of the entire network, and (c) a table view summarizing all ego-networks and their characteristics.

Core Visualization

We employ a subway map metaphor to reveal the evolutionary patterns of a dynamic ego-network (Figure 1). Each actor (the ego and alters) in the network is represented as an actor line from the time step when the actor first joins the network to the time step when he/she last appears, where dashed lines indicate temporary absences. Representing temporal information as lines has proven intuitive and effective [26, 34]. It allows a user to identify the lifespan of an actor in the ego-network and pinpoint the events of entering and leaving the network (A1). Users can also observe the stability of an ego-network by viewing the lengths of the lines (A3). For example, a large number of shorter actor lines indicates that there are frequent turnovers of alters in the ego-network.



Search:

Ego Name - id [Time Len]	Vertex Num	Edge Num	Edge-Vertex Ratio
T. von Landesberger-57 [5]	12.80	54.80	2.97
M. Whiting-1833 [5]	11.00	33.00	2.86
G.H. Weber-839 [5]	10.60	37.60	2.82
H. Bosch-640 [5]	12.40	37.80	2.75
W. Ribarsky-221 [15]	11.27	37.60	2.68
J. Freire-475 [7]	10.43	30.86	2.66
R. Wegenkittl-2892 [6]	9.33	31.83	2.60
E.W. Bethel-842 [5]	9.40	32.00	2.59
R. Chang-131 [7]	13.00	34.57	2.58
P.-T. Bremer-374 [9]	9.00	24.78	2.45
H. Fuchs-3500 [6]	6.50	17.33	2.44
D. Bartz-1441 [5]	9.00	24.80	2.43
M.S. Brown-1455 [5]	8.00	19.80	2.42
J. Thomas-1749 [10]	9.10	24.20	2.39
Jing Yang-164 [9]	9.78	28.67	2.34
J. Kohlhammer-630 [5]	10.00	34.60	2.29
H. Foote-1762 [6]	7.50	18.17	2.28
C. Andrews-677 [5]	8.40	21.40	2.2

Showing 1 to 50 of 50 entries

Figure 2. The EgoLines interface consists of four UI components: a) a main view showing the current selected ego-network, b) an overview displaying the entire network, c) a table view listing all the extracted ego-networks, and d) a toolbar for accessing different system functions. In the main view, line colors indicate the clusters of actors in the ego-network. In 2010, lines are sorted by clusters, which illustrates that the orange and purple clusters merged into one in the next year. Also, it indicates the highlighted actor (WD) was a bridge that connected all the clusters.

Actor lines are color-coded with different actor attributes on demand, including both categorical and numerical (discrete or continuous) attribute values. This allows a user to observe changes in attributes over time. For example, when encoded with alter cluster categories, which represent alter communities in the ego-network, Figure 2-a(ii,iii) shows that the orange and purple clusters in 2010 merged into one big (purple) cluster in 2011 (A2). Color gradients are also used to ease the viewing of cluster transitions. Moreover, when colored with betweenness centrality—which reflects an actor's influence on the transfer of items through the network [29]—interesting temporal patterns can emerge. For example, in Figure 4 *JF* (the second actor from the top) had a valley of the metric in 2013, suggesting that *JF*'s impact was lower than usual in that year (A4).

At each time step of an actor line, the actor is indicated as white rectangles (*actor nodes*), similar to the *major stops* of a subway. For the actor nodes, a squared-corner rectangle indicates the start of the actor line (i.e., first joining into the network), and a rounded-corner rectangle indicates a subsequent occurrence. Further, dashed borders represent an absence of the actor at the time step (A1). For example, in Figure 1 that shows the academic co-authorship network centered at *PD*, *FC* collaborated with the ego *PD* in 2010, stopped for 3 years, and collaborated with *PD* again in 2014. The actor's connections to others within each time step are rendered as small white circles (*connection dots*), like *minor stops* of a subway. Since the actor lines are tightly packed together, they form a block of line segments at each time step where the connection dots display the ego-network topology in a representation akin to an adjacency matrix. Taking advantage of the matrix representations [2, 6], users can perform topological analysis tasks on denser and larger networks (B1, B2). The size of these blocks also reveals the overall growth or contraction of ego-networks (A4). To pack actor lines for revealing network topologies in matrices while maintaining an aesthetic layout with fewer crossings and bends, an algorithm similar to [11] is employed using an inside-out heuristic (Figure 5).

The design of EgoLines is sufficiently general and can represent different types of ego-networks, in addition to undirected unweighted networks as described above (Figures 1,2). Figure 3-a shows a weighted ego-network where the connection dots indicate their weights using color density. To enhance

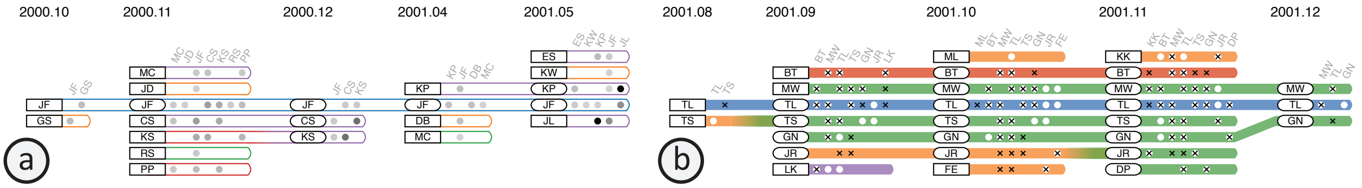


Figure 3. EgoLines can be used for visualizing a) weighted and b) directed dynamic ego-networks. Connection weights are shown as the color density of small circles. Connection directions are indicated as either circles (outgoing) or crosses (incoming), or both (bi-directional).

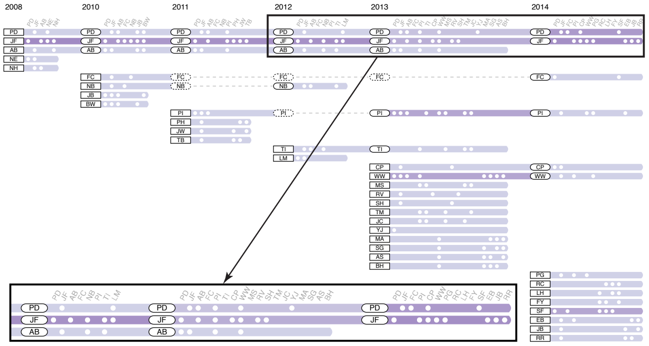


Figure 4. Visualizing the ego-network of P. Dragicevic (PD) using an unpacked view, compared to the compact view in Figure 1. The line colors indicate the betweenness centrality of actors at different time steps using a white-purple color scheme. By tracing the color of the actor line of JF, a valley of the betweenness centrality metric is observed in 2013 (magnified here).

the perception of the connection weights, the outline color of the actor lines is used to encode the original actor attributes, from which users can access property values associated with both actors and connections (B3). Moreover, Figure 3-b shows a directed ego-network where white dots and black crosses indicate out-going and in-coming connections respectively, relative to the actor represented by the actor line.

Interaction

EgoLines incorporates rich interactions to support various egocentric analysis tasks of dynamic networks. Smooth animations are employed to ease a user's understanding of the changes between visual states.

Unpacked view. The user can split the above compact visualization into a list of actor lines that are displayed separately (Figure 4). The actor lines are sorted by their starting time and then length. Since all the lines are straight, this view is convenient for certain analysis tasks, such as browsing characteristics of actor lifespans (A1) and tracing specific attribute values (A4). However, it is less space efficient and breaks-up the adjacency matrix layout.

Alter levels. To observe patterns of only 1-level or 2-level alters, the user can reveal the *alter boundary* of the 1-level ego-network with a light-gray convex hull, which is akin to *fare zones* in a subway system (Figure 1). This can be used to determine the change of the 1- or 2-level network sizes over time (A4). The packing algorithm described above further satisfies the restriction of placing 1-level alters closer to the ego. However, this may introduce more crossings.

Line reordering. The user can reorder the actor lines according to the visualized actor attribute (e.g., cluster categories) at a given time step. Line segments of the same actors in other time steps are also reordered to avoid too many crossings. This could be useful to identify bridges of a network (B2). For example, Figure 2-a(i) shows that the highlighted alter *WD* is a bridge (other than the ego *RC*) connecting the orange, green and purple alter clusters in 2010, since *WD* had connections spreading across all the clusters.

```
Input: Actor lines $A_i : (s_i, n_i)$ with starting time step $s_i$ and length $n_i$
Output: Packing positions of actor lines at each time step $P_{A_i} : [p_1, \ldots, p_{n_i}]$
sort $A_i$ by $n_i$ in descending order ;            /* use stable sort */
sort $A_i$ by $s_i$ in ascending order ;             /* use stable sort */
$P_{A_0} \leftarrow [0, \ldots, 0]$ ; /* the earliest and the longest one */
$S_{up} \leftarrow 0, S_{down} \leftarrow 0, L_{up} \leftarrow [0, \ldots, 0], L_{down} \leftarrow [0, \ldots, 0]$;
foreach $A_i$ in the rest of the sorted list do
    if $S_{up} < S_{down}$ then                          /* place above */
        $S_{up} \leftarrow S_{up} + n_i$;
        for $j = s_i$ to $s_i + n_i - 1$ do
            $L_{up}[j] \leftarrow L_{up}[j] + 1, P_{A_i}[j - s_i] \leftarrow -L_{up}[j]$ ;
    else                                                 /* place below */
        $S_{down} \leftarrow S_{down} + n_i$;
        for $j = s_i$ to $s_i + n_i - 1$ do
            $L_{down}[j] \leftarrow L_{down}[j] + 1, P_{A_i}[j - s_i] \leftarrow L_{down}[j]$ ;
```

Figure 5. Actor lines reordering algorithm.

Filtering. Several data filtering mechanisms allow the user to exclude actor lines above and below user defined thresholds, such as the alter's lifespan length (A3), start and end time steps (A1), or alter level (A4).

Highlighting. When the user hovers over an actor node, occurrences of that actor in other time steps are highlighted, as well as the associated actor line. The corresponding column in the matrix block is also visually emphasized. Similarly, when a connection dot is hovered over, connections of that actor are highlighted across the whole visualization. These features can help users identify the presence and absence of the same actor over time. When needed, the user can choose to overlay grids to increase the matrix readability (Figure 2).

Paths. When an alter is hovered over, a series of curved arrows trace the shortest paths from that alter to the ego (B2), using a similar visualization proposed by Sheny et al. [32]. This addresses the recognized inefficiency of adjacency matrices for tracing paths [18]. For example, Figure 1 shows the shortest path from the highlighted alter *MA* to the ego *PD* via *WW*. The ego is the default sink for calculating these shortest paths, but any actor can be selected as a sink to view the shortest paths. The overlay of shortest paths can also be locked with a right click, allowing further investigation of actors along those paths.

Overview and Network Table

To support high-level analysis tasks, EgoLines offers an overview of the entire network and a table view of all 2-level ego-networks extracted from the data.

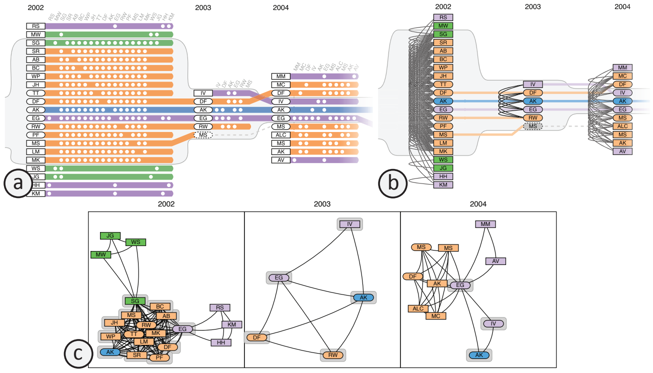


Figure 6. Dynamic ego-networks visualization techniques compared in the study (all showing the same data): a) EgoLines (EL), b) node-link diagrams visualization (NL), and c) small multiples visualization (SM).

The overview shows an aggregation of the whole network across all time steps in a node-link diagram (Figure 2-b). The size of a node indicates the number of time steps where the actor exists in the dataset. The thickness of a link indicates the total occurrences of that specific connection. The selected ego-network is highlighted with the ego in red and alters in orange. This provides data context (C1), allowing users to easily navigate to ego-networks of surrounding actors (by clicking the desired node). To browse the entire network along time steps, a user can drag a slider to fade out non-related nodes and links with animations (C1).

EgoLines lists all the extracted ego-networks in a table similar to [16] (Figure 2-c). Each row represents an ego-network, and the columns show temporal distributions of different graph metrics in small histograms, such as vertex or edge numbers, and edge-vertex ratios. The x-axes of the histograms are aligned by time, allowing users to spot trends, discover similar ego-networks based on the metrics, and find missing time steps in the data (C2). A user can then select a table row to display the ego-network in the main view. Searches and sorting by the metric of each column are also supported.

CONTROLLED STUDY

We conducted a user study to assess the strengths and weaknesses of EgoLines for the analysis of dynamic ego-networks, comparing EgoLines against two other techniques. We focused on the core visualization showing a specific dynamic ego-network (i.e., addressing analytical question sets A and B), since the node-link overview and the table view have already been studied [5, 16]. The experiment used 2-level ego-networks extracted from an undirected and unweighted temporal academic co-authorship network derived from [22].

Techniques

In addition to EgoLines (EL), we considered two other timeline-based techniques: a node-link diagram based visualization similar to [19] (NL), and the most common small multiples visualization of dynamic networks (SM). They used the same visual language as EgoLines. In NL (Figure 6-b), the same layout of actors was maintained and similar lines connected the same actors at different time steps. The nodes were color-coded, since the actor lines might be occluded by the links. In SM (Figure 6-c), a force-directed layout was used to display a time step with a node-link diagram.

T1	What years did *person* first join and last leave the network?	A1
T2	Did *person* ever leave and rejoin the network? If so, what year did he/she first rejoin the network?	A1
T3	Who had the longest relationship with the ego?	A1
T4	Did the overall network size increase or decrease in *year1–year2*?	A4
T5	Did the 1-level network size increase or decrease in *year1–year2*?	A4
T6	Did *cluster1* and *cluster2* in *year* integrate into one next year	A2
T7	Did *cluster* in *year* split into multiples next year?	A2
T8	How many people had relationships with the ego for *n*+ years?	A3
T9	How many people were directly connected to the ego in *year*?	B1
T10	Was *person* directly connected to the ego? If not, how many shortest paths connected him/her to the ego?	B1
T11	Who had the largest number of connections in *year*?	B1
T12	Who were the common connections between the ego and *person*?	B2
T13	Who bridged *cluster1* and *cluster2* in *year*?	B2

Table 1. Experimental tasks (columns from left to right: task number, task description, and analytical question type).

For simplicity, the filtering operations in all three techniques were disabled, but interactive highlighting was supported. For example, when hovering over an actor, all the connections of that actor were revealed, as well as occurrences of that actor in other time steps. With SM, a user could expose all 1-level alters with a gray outlined region (corresponding to the boundary of 1-level ego-network in EL and NL), and could reposition actors to eliminate occlusions. Further, the color gradients on actor lines in EL were removed to have a fair comparison, as the colors in SM were rendered discretely.

Participants and Apparatus

We recruited 18 volunteers, 13 males and 5 females, aged 23–34 (μ=27.6, σ=3.5). All of them had some familiarity with the data domain, i.e., academic publications and co-authorships, but not this particular dataset. Participants were from various backgrounds, including science, engineering, humanities, and economics. All had normal or corrected-to-normal vision without color vision deficiency. The experiments were conducted on a desktop computer (2.53GHz Intel Xeon CPU, 24GB memory) with a 24″ monitor. The effective area of the visualization on screen was 1600×1200 pixels.

Tasks and Design

We developed 13 tasks about the academic co-authorship network data based on the aforementioned analytical questions for dynamic ego-networks (Table 1). We focused on topology-centric tasks (i.e., A1-3 and B1-2) and high-level attribute-centric tasks (i.e., A4: network size trends), since low-level attribute retrievals (B3) are included in some other tasks (e.g., obtaining the cluster of an alter in A2). Each task included a multiple-choice question (with at least four choices) and a visualization showing data specific to the question. In addition to these choices, there was a “do not know” option to discourage random guessing. Participants were presented with the three techniques in a counter-balanced order. Within each technique, they completed two repetitions of the 13 tasks, in which the tasks were shown in a fixed order. Thus, the whole study contained 3 techniques × 13 tasks × 2 repetitions = 68 trials. To avoid duplications, six alternative questions with similar levels of difficulty (in terms of data size and visual complexity) were developed for each task. The six task sets (3 techniques × 2 repetitions) were randomly generated from the alternatives for each participant.

Q1	Technique *X* was easy to learn.	general
Q2	Technique *X* was easy to use.	general
Q3	Technique *X* helped discover people relationship events (joining and leaving).	A1
Q4	Technique *X* helped discover people relationship lengths.	A2
Q5	Technique *X* helped discover changes in clusters (merging and splitting).	A3
Q6	Technique *X* helped discover network size trends (1- and 2-level).	A4
Q7	Technique *X* helped discover connections of a specific person (direct and indirect).	B1
Q8	Technique *X* helped discover paths between two people.	B2
Q9	Please rank the three techniques with preferences.	preference

Table 2. Questionnaire (X=EL/NL/SM). Responses are collected (except Q9) using a 7-point Likert scale (strongly disagree to strongly agree).

Procedure

The study began with a brief introduction to the data domain and the three visualizations. Participants were then asked to try all techniques with an example ego-network to learn the system features. After that, for each technique, they completed a training block and then a testing block.

During training, participants were instructed to think aloud, and the investigator helped answer questions and overcome difficulties. The training block was the same for all participants, comprised of the same six tasks (T1, T2, what was the network size in *year*, how many clusters in *year*, T9, T12) on the same dataset. The training tasks and dataset were different from those in the actual experiment. Participants were prompted for the correct answer after every trial, and if they answered incorrectly, they needed to go back to the same task to figure out why.

In the testing block, participants went through 26 tasks for the technique. There was no time limit to complete any of the trials, but to avoid frustration on difficult tasks, a dialog popped up every minute asking participants if they needed more time or wanted to skip the task. Task completion times and participant answers were recorded. We also conducted observations and screen-captured the entire session. After the study, participants completed a post-study questionnaire (Table 2). The study lasted around 1.5 hour.

Results

Here we report results obtained from the controlled study. We compared the three techniques by their task completion times, task error rates, and participants' preferences. For every technique, we computed the completion time (correctly answered) and the error rate of each task for each participant by averaging corresponding trials. We then calculated the means and 95% confidence intervals (CI) by taking all participants into consideration (Figure 7).

Completion Time

Across all tasks, EL achieved the fastest average task completion time (17.5s, CI: [14.5, 19.6]), compared to NL (21.2s, CI: [17.7, 23.4]), and SM (23.8s, CI: [20.0, 26.7]). There were a larger effect between EL and SM, and a smaller effect between EL and NL.

For temporal tasks about dynamic ego-network evolutions (T1-8, analytical question set A), we observed time performance effects: EL and NL were, in many cases, faster than SM (except for T4, T5, T6, for which there is no observable effect), with strong evidence that EL and NL has a substantial advantage over SM for tasks T3 and T8 (requiring assessment of relationship lengths). This was reasonable as the actor lines in EL and NL could help users discover low-level temporal patterns (A1-3). Overall, EL and NL had similar completion times in T1-8. This could be because the main differences between the two techniques rests in representing the topology of time step networks (i.e., matrices v.s. node-links), which was not used by participants in temporal tasks.

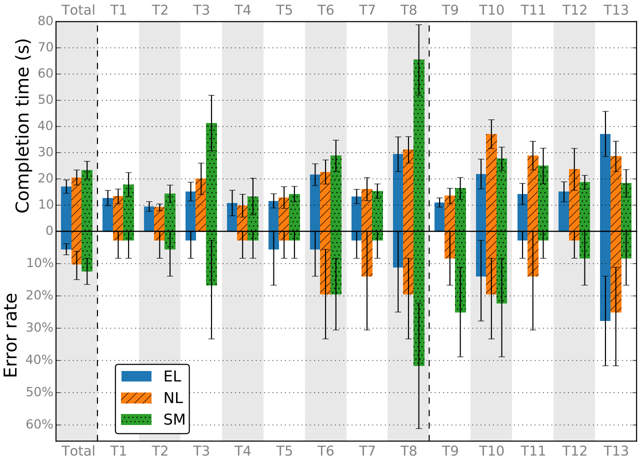


Figure 7. Completion times and error rates of all three techniques for each task. Error bars indicate 95% confidence intervals (n = 18) estimated using bootstrap [37]. Completion times include correct trials only, and zero error rate indicates no mistake at all.

For topological tasks about specific ego-network time steps (T9-13, analytical question set B), there is evidence to support that EL performed the best in T9-12, which substantiates the benefits of adjacency matrices in browsing network topology (B1-2) [18]. Especially in T11 (finding the most connected actor), EL showed an important improvement in completion time. Surprisingly, EL performed much worse than SM in T13 which consisted of finding bridges between clusters. We thought that the bridge pattern was obvious after cluster sorting in EL, e.g., through observing the connections in the *WD* column in Figure 2-a(i). However, this requires in-depth understanding of adjacency matrices, which explain the lower performance. The large confidence interval of EL also reflects the variance of participants' comprehension. In general, NL performed the worst in these tasks, which could be due to visual clutter of the links. Although visual clutter also existed in SM, the level of complexity seemed more impactful when aligning actors linearly in NL.

In short, EL and NL achieved similar performance in temporal tasks, but NL's performance clearly dropped in topological tasks. SM was not suited for temporal tasks but had advantages in topological tasks. However, there is evidence that SM is slower than EL for topological tasks.

Error Rate

Overall, EL had the lowest error rate (5.6%, CI: [3.8%, 7.4%]) compared to NL (10.3%, CI: [6.2%, 14.7%]) and SM (12.4%, CI: [9.6%, 17.9%]), and EL had the smallest 95% CI—revealing lower variations in error rates (Figure 7).

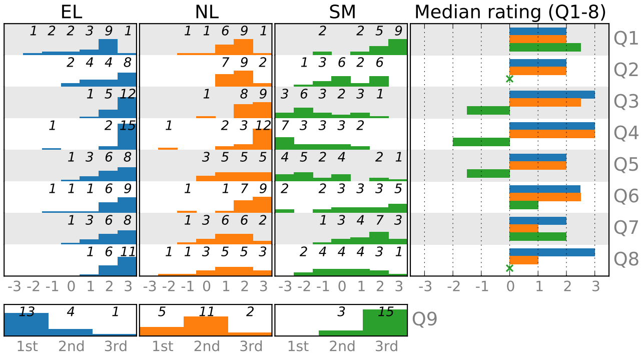


Figure 8. A summary of participant responses to the questionnaire (Table 2). The first three columns show detailed distributions of ratings (7-point Likert scale) and rankings (from the most to the least preferred), and the last column shows median ratings.

Within temporal tasks (T1-8), there are three tasks in which EL had zero errors across all participants (T1, T2 and T4). There is fair evidence that EL outperforms the two other techniques for T6 and T8, which were related to cluster changes (A2) and network stability (A3). The large variation in these tasks, as indicated by the large CIs, call for further examination. In other tasks (T3, T5 and T7), the results are largely inconclusive, with EL having comparable performance to NL or SM, whichever had fewer errors. For topological tasks (T9-13), there is strong evidence that EL outperforms NL and SM on tasks T9 and T12, with a null error rate. EL also seemed to yield fewer errors in T10, but the effect is uncertain. These latter three tasks all related to both accessibility and connectivity (B1-2). Overall, our results support the advantages of adjacency matrices [18], except for finding the shortest paths, which yielded high performance variations. Finally, there is good evidence that SM is the best suited for T13 (finding the bridge). It could be because the force-directed layout made the bridge between two densely-connected clusters visually salient. In summary, the overall benefit of EL on accuracy is observable, but our results suggest that a future redesign of the time step blocks (adjacency matrices) are needed to make T13 less abstract.

Questionnaire Results

From participants' ranking of the three techniques (Q9, Table 2), EL was the most preferred, with 13 out of 18 users ranking it first. Figure 8 shows detailed questionnaire results. EL and NL received similar ratings in Q1-6 which were related to overall impressions (Q1-2) and temporal tasks (Q3-6). Except for the ease of learning (Q1)—for which EL and NL were rated a bit lower than SM overall, participants consistently preferred these two techniques over SM. For topological tasks, NL was less popular. For example, in Q7 (finding connections), EL and SM were more favored (by 1 in median rating); EL was the most preferred in identifying paths (Q8) which might be due to the curve overlay.

USE-CASE SCENARIO

To study the effectiveness of the entire EgoLines system in practice, we conducted two one-hour interview sessions with a domain expert who is a professor at the management school of a university. His research focuses on analyzing the social dynamics of people in large organizations and online communities. He uses many egocentric analysis methods to study patterns of interactions between individuals and their close social networks. In the first interview, we introduced the EgoLines interface with the academic collaboration dataset and discussed potential usages of the tool. In the second session, we asked the expert to explore a dataset matching his research interests and conducted an in-depth discussion with him. The dataset was an email communication network of employees at Enron, a bankrupted company due to its financial scandals, which contained emails among 142 employees from Nov. 1998 to Jun. 2002 [13]. We constructed a dynamic network based on email exchange with the time step set to one month, resulting in weighted networks with connection weights representing the numbers of emails exchanged between two people. With the help of our expert user, we derived the following use-case of EgoLines.

From this Enron email exchange network data, the analyst aims to explore communication networks centered at influential people of the company (i.e., ego-networks) that reflect evolutionary patterns of people's relationships across time. The analyst wants to identify overall differences and similarities of these ego-networks, as well as to investigate career developments of these key people, social interactions among them, and their relations to important events during the financial scandal.

After loading the data into EgoLines, the analyst first browses the overview by dragging the time slider (Figure 9-a) to observe the growth of the company: the size of the whole network becomes larger and larger (C1). Then, the analyst sorts the ego-networks in the table view by density (Figure 9-b). Two Enron CEOs (at different times of the company), D. Delainey and J. Lavorato, are ranked at the top. The analyst wonders what other CEOs' ego-networks look like. Using the search box, the analyst further finds two other CEOs, J. Skilling and K. Lay, according to the company's profile. From the histograms that summarize distributions of key metrics of the dynamic ego-networks, he observes that although all four CEOs have similar lengths of tenure (20–28 months), Delainey and Lavorato seem to maintain relatively larger and more constant ego-networks than the other two CEOs (C2). Next, the analyst quickly examines the ego-networks of the four CEOs in the main view. Using filtering operations on alter levels and relationship lengths, he further confirms the above observations and finds that Delainey and Lavorato both have more stable 1-level ego-networks across time than the other two CEOs (A3).

The analyst then focuses on Skilling's 1-level ego-network, because he is the key player in the financial scandals according to the public information (Figure 9-c). He first spots two anomalies of Skilling's ego-network in terms of size: significant peaks and more clusters in Apr. 2001 and Aug. 2001 (A4), as shown in Figure 9-c(i,ii). The former coincides with the incident when Skilling verbally attacked Wall Street analyst Richard Grubman who questioned Enron's unusual accounting practice; and the latter corresponds to his resignation as the CEO. By switching the view to show connection weights as color density, the analyst further

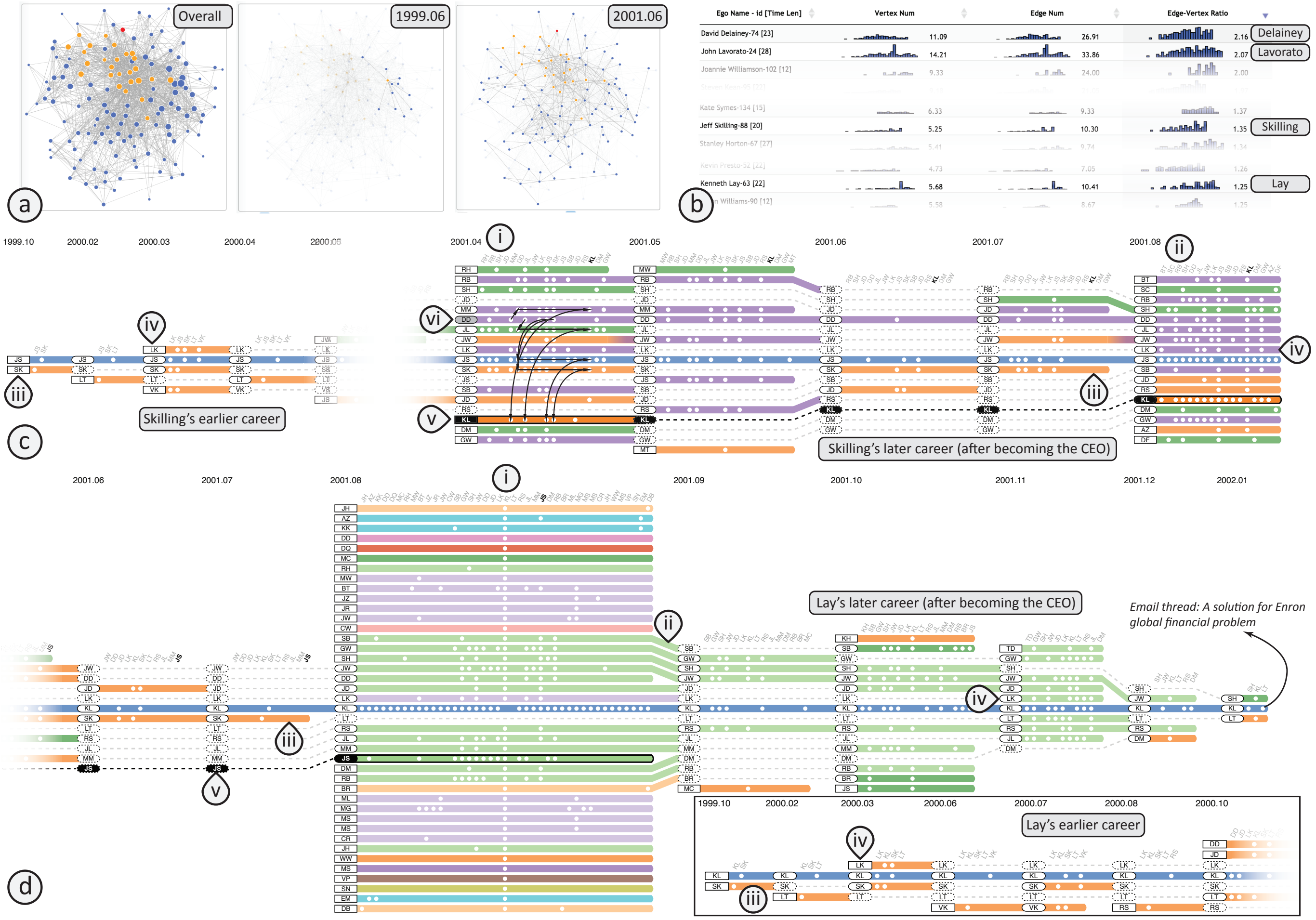


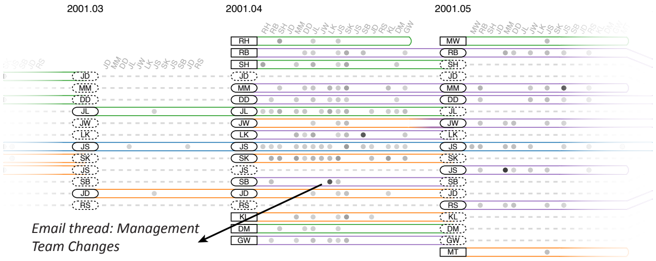
Figure 9. Using EgoLines to explore a dynamic network of email communications among employees at Enron. a) The overview shows the structure of the entire network. b) The table view reveals general patterns in temporal distributions of several network metrics. The main view visualizes the 1-level dynamic ego-network of a specific individual: c) Jeff Skilling (JF) and d) Kenneth Lay (KL), the two Enron CEOs (in different time of the company).

identifies the most email exchanges in Apr. 2001 happened between *LK* (L. Kitchen, President) and *SB* (S. Beck, COO) regarding: *Management Team Changes*, which could reflect Skilling's later resignation (Figure 10). The analyst then switches to the unpacked view that shows each actor line in a row, and he finds that *SK* (S. Kean, Vice President) and *LK* had the longest relationships with Skilling (A1), where the starting and ending time steps are indicated in Figure 9-c(iii,iv). However, the analyst is intrigued that the actor line of *KL* (K. Lay, the next CEO) is very short (Apr.-Aug. 2001), suggesting that Lay and Skilling did not have much interaction (Figure 9-c(v)). With *KL* selected in Apr. 2001, the analyst hovers over a number of actors and reveals the shortest paths from them to *KL*. Although *KL* only had five connections in that month, four of them connected him to many key persons at Enron, such as *DD* (D. Delainey, the former CEO) as shown in Figure 9-c(vi) (B1, B2).

Further, the analyst opens Kenneth Lay's ego-network in the main view, and immediately finds its size soared in Aug. 2001 when Lay became the CEO (Figure 9-d); and so did the number of clusters. Lay was also the central bridge, since his connections spread across multiple clusters (B2), as seen from the *KL* column in Figure 9-d(i). But only a few coworkers (mostly in higher management level) had continued communications with *KL*, who are mainly in the light green cluster (Figure 9-d(ii)). The ego-network shrank dramatically and ended in Jan. 2002 when Lay resigned the CEO. One of the last emails associated with him had the subject: *A solution for Enron global financial problem*. With the unpacked view, the analyst examines people who had the longest relationships with Lay, which turns out to also having *SK* and *LK* (Figure 9-c(iii,iv)), similar to Skilling. This illustrates the social science concept, *ghost ties* [28]: Skilling and Lay had stronger ties than it appeared, although they only had four months (including two months absence) in each other's ego-networks (Figure 9-b(v),c(v)).

DISCUSSION

Although both EL and NL were both likely to be unfamiliar to participants, the results indicate they were not hard to learn and much easier to use than SM (Figure 8). Still, we observed that some participants chose suboptimal approaches when completing certain tasks, which indicates that EL and NL require some training to acquire the right reflex. For example, in T5 (1-level ego-network size trends), some users remembered the feature of showing the boundary of 1-level networks, but some just hovered over the ego in each year to



Email thread: Management Team Changes

Figure 10. Revealing connection weights (email exchanging frequencies) of Skilling's ego-network with the color density of connection dots.

count its direct connections (i.e., the dots or links). Similarly, for assessing the relationship lengths between the ego and alters (T3 and T8), the unpacked view (Figure 4) supported in EL and NL definitely makes the task much easier, but several participants insisted on using the default compact view. Thus, the learning of new visualization techniques includes not only familiarity of novel visual encodings, but also the ability of reasoning hand-to-hand with the visualization, which requires a good understanding of the features and what analysis they enable. Another interesting observation was that some participants prompted "Wow, a subway map?" before we even introduced the visual encodings of EgoLines. Applying the subway map metaphor made the visualization more understandable and memorable to the general audience. Several participants used the metaphor in solving tasks, saying to themselves: "The green line stops here...I can change line here". It was also found more effective to communicate with the visual encodings in plain terms such as "subway lines" and "stops" rather than jargons such as "adjacency matrices".

The results of both studies indicate the effectiveness and usefulness of EgoLines in performing egocentric analysis on dynamic networks. Yet, there are still several limitations. First, EgoLines might not scale well to dynamic networks very long in time that generate very large visualizations. Aggregations of multiple time steps into one network with the spirit of MultiPiles [4] could solve this issue. Second, when the number of actors in the network becomes larger, the matrix representation of each time step grows and may include many empty rows indicating the absence of actors. However, EgoLines could be extended with multi-scale exploration, such as grouping actors into hierarchical clusters [12], and an option to hide the dashed actor line segments temporarily. Third, larger datasets may also result in more crossings of the actor lines, so advanced layout algorithms (e.g., [34]) need to be applied to generate a clearer visualization. Fourth, EgoLines did not show advantages in finding bridges, which could be very effective if users understood the notion of adjacency matrices better, suggesting that more intuitive visual designs are warranted. Last, the overview of EgoLines (Figure 2-b)—that is not the focus of this paper—can be enhanced with filtering and aggregation techniques (e.g., [36, 40]) to reduce the visual clutter.

While we conducted extensive evaluation of EgoLines with both quantitative and qualitative methods, there still exist limitations in the studies. First, we compared three significantly different techniques in the controlled study. But it is an interesting future work to assess the effect of EgoLines' subway map metaphor design compared to traditional approaches using sequential adjacency matrices (e.g., [6, 4, 41]). Second, we used citation networks and email communication networks as the testing datasets, which may not reflect experimental results with other types of networks, such as large brain neural pathways. Third, in this paper we focus on egocentric analysis of networks, so the study results may have limited implication on whole network analysis for which most existing techniques are designed.

It is also worth noting that although EgoLines is designed for dynamic ego-networks analysis, the main visualization (Figure 2-a) can be applied for visualizing general dynamic networks by not differentiating ego and alters. Many of the functions, such as sorting actor lines, filtering with lengths and starting and ending time, remain useful in such more general setting. Moreover, weighted and directed dynamic networks can be visualized in the same manner introduced in Figure 3. As a convention of egocentric analysis, we currently only consider ego-networks containing alters up to two steps away from the ego, but EgoLines is not restricted to visualizing 2-level ego-networks. Further, the visual metaphor of alter boundary can be certainly extended to multiple levels by placing alters with different distances gradually from the center in different colored zones. But this may introduce more crossings as the network size grows, and a method for neatly aggregating actors are required.

CONCLUSION AND FUTURE WORK

We have presented EgoLines, an interactive visualization for assisting egocentric analysis of dynamic networks. EgoLines provides a novel visual design that represents a dynamic ego-network using a "subway map" metaphor. In addition, it offers an overview and a table view to support the browsing of all extracted ego-networks with high-level context and aggregations. The design of EgoLines was guided by a series of dynamic ego-network analytical questions that were derived from interviews of experts and the literature. An experiment comparing different dynamic network visualizations and a case study of real-world data suggested that EgoLines is effective and useful in conducting ego-network analysis.

In the future, we would like to address the scalability of the main EgoLines visualization (Figure 2-a) by developing and integrating the techniques discussed above. We also plan to experiment with different designs to enhance the efficiency of EgoLines in completing tasks related to finding bridges in networks. Moreover, we wish to conduct more case studies of EgoLines with real-world scenarios, and evaluate the design in visualizing more general dynamic networks.

ACKNOWLEDGMENT

We thank all the participants and the domain experts involved in the studies. We also thank Michelle Annett for the constructive feedback on the early draft of this paper and the reviewers for their insightful comments and suggestions.

REFERENCES

1. A. Al-Awami, J. Beyer, H. Strobelt, N. Kasthuri, J. Lichtman, H. Pfister, and M. Hadwiger. 2014.

NeuroLines: A Subway Map Metaphor for Visualizing Nanoscale Neuronal Connectivity. *IEEE Transactions on Visualization and Computer Graphics* 20, 12, 2369–2378. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346312

2. B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. 2013. Weighted Graph Comparison Techniques for Brain Connectivity Analysis. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*. 483–492. DOI: http://dx.doi.org/10.1145/2470654.2470724

3. D. Archambault, H. C. Purchase, and B. Pinaud. 2011. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics* 17, 4, 539–552. DOI: http://dx.doi.org/10.1109/TVCG.2010.78

4. B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski. 2015. Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks. *Computer Graphics Forum* 34, 3, 31–40. DOI:http://dx.doi.org/10.1111/cgf.12615

5. B. Bach, E. Pietriga, and J.-D. Fekete. 2014a. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics* 20, 5, 740–754. DOI:http://dx.doi.org/10.1109/TVCG.2013.254

6. B. Bach, E. Pietriga, and J.-D. Fekete. 2014b. Visualizing Dynamic Networks with Matrix Cubes. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*. 877–886. DOI: http://dx.doi.org/10.1145/2556288.2557010

7. F. Beck, M. Burch, S. Diehl, and D. Weiskopf. 2014. The state art in visualizing dynamic graphs. In *EuroVis State-of-the-Art Reports (STARS)*.

8. P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini. 2012. A Survey of Context Data Distribution for Mobile Ubiquitous Systems. *Comput. Surveys* 44, 4, 45 pages. DOI:http://dx.doi.org/10.1145/2333112.2333119

9. U. Brandes and B. Nick. 2011. Asymmetric relations in longitudinal social networks. *IEEE Transactions on Visualization and Computer Graphics* 17, 12, 2283–2290. DOI: http://dx.doi.org/10.1109/TVCG.2011.169

10. M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf. 2011. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics* 17, 12, 2344–2353. DOI: http://dx.doi.org/10.1109/TVCG.2011.226

11. L. Byron and M. Wattenberg. 2008. Stacked Graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics* 14, 6, 1245–1252. DOI:http://dx.doi.org/10.1109/TVCG.2008.166

12. N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete. 2008. ZAME: Interactive Large-Scale Graph Visualization. In *Proc. IEEE Pacific Symposium on Visualization*. 215–222. DOI:http://dx.doi.org/10.1109/PACIFICVIS.2008.4475479

13. Enron Data. accessed in 2015. http://enrondata.org.

14. M. Farrugia, N. Hurley, and A. Quigley. 2011. Exploring temporal ego networks using small multiples and tree-ring layouts. In *Proc. International Conference on Advances in Computer-Human Interactions*. 79–88. http://www.thinkmind.org/download.php?articleid=achi_2011_4_20_20100

15. D. Fisher. 2005. Using egocentric networks to understand communication. *IEEE Internet Computing* 9, 5, 20–28. DOI: http://dx.doi.org/10.1109/MIC.2005.114

16. M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. 2010. ManyNets: an interface for multiple network analysis and visualization. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*. 213–222. DOI:http://dx.doi.org/10.1145/1753326.1753358

17. C. Friedrich and P. Eades. 2001. The Marey Graph Animation Tool Demo. In *Graph Drawing*, Joe Marks (Ed.). Vol. 1984. Springer, 396–406. DOI: http://dx.doi.org/10.1007/3-540-44541-2_37

18. M. Ghoniem, J.-D. Fekete, and P. Castagliola. 2005. On the Readability of Graphs Using Node-link and Matrix-based Representations: A Controlled Experiment and Statistical Analysis. *Information Visualization* 4, 2, 114–135. DOI: http://dx.doi.org/10.1057/palgrave.ivs.9500092

19. M. Greilich, M. Burch, and S. Diehl. 2009. Visualizing the evolution of compound digraphs with TimeArcTrees. *Computer Graphics Forum* 28, 3, 975–982. DOI:http://dx.doi.org/10.1111/j.1467-8659.2009.01451.x

20. M. Grossetti. 2005. Where do social relations come from?: A study of personal networks in the Toulouse area of France. *Social Networks* 27, 4, 289–300. DOI: http://dx.doi.org/10.1016/j.socnet.2004.11.004

21. S. Hadlak, H. Schulz, and H. Schumann. 2011. In Situ Exploration of Large Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics* 17, 12, 2334–2343. DOI: http://dx.doi.org/10.1109/TVCG.2011.213

22. P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. 2015. Visualization Publication Dataset. http://vispubdata.org.

23. R. Keller, C. M. Eckert, and P. J. Clarkson. 2006. Matrices or Node-link Diagrams: Which Visual Representation is Better for Visualising Connectivity Models? *Information Visualization* 5, 1, 62–76. DOI: http://dx.doi.org/10.1057/palgrave.ivs.9500116

24. N. W. Kim, S. K. Card, and J. Heer. 2010. Tracing Genealogical Data with TimeNets. In *Proc. International Conference on Advanced Visual Interfaces*. 241–248. DOI: http://dx.doi.org/10.1145/1842993.1843035

25. B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. 2006. Task Taxonomy for Graph Visualization. In *Proc. AVI workshop on BEyond time and errors (BELIEVE)*. 1–5. DOI: http://dx.doi.org/10.1145/1168149.1168168

26. S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. 2013. StoryFlow: Tracking the Evolution of Stories. *IEEE Transactions on Visualization and Computer Graphics* 19, 12, 2436–2445. DOI: http://dx.doi.org/10.1109/TVCG.2013.196

27. A. J. Omalley, S. Arbesman, D. M. Steiger, J. H. Fowler, and N. A. Christakis. 2012. Egocentric social network structure, health, and pro-social behaviors in a national panel study of Americans. *PLoS One* 7, 5, e36250. DOI: http://dx.doi.org/10.1371/journal.pone.0036250

28. M. Papagelis, F. Bonchi, and A. Gionis. 2011. Suggesting Ghost Edges for a Smaller World. In *Proc. International Conference on Information and Knowledge Management*. 2305–2308. DOI: http://dx.doi.org/10.1145/2063576.2063952

29. C. Prell. 2011. *Social Network Analysis: History, Theory and Methodology*. SAGE. https://us.sagepub.com/en-us/nam/social-network-analysis/book231856

30. F. Reitz. 2010. A Framework for an Ego-centered and Time-aware Visualization of Relations in Arbitrary Data Repositories. *CoRR - Computing Research Repository* abs/1009.5183. http://arxiv.org/abs/1009.5183

31. S. Rufiange and M. J. McGuffin. 2013. DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics* 19, 12, 2556–2565. DOI: http://dx.doi.org/10.1109/TVCG.2013.149

32. Z. Sheny and K.-L. Maz. 2007. Path Visualization for Adjacency Matrices. In *Proc. Joint Eurographics / IEEE VGTC Conference on Visualization*. 83–90. DOI: http://dx.doi.org/10.2312/VisSym/EuroVis07/083-090

33. L. Shi, C. Wang, and Z. Wen. 2011. Dynamic network visualization in 1.5 D. In *Proc. IEEE Pacific Visualization Symposium*. 179–186. DOI: http://dx.doi.org/10.1109/PACIFICVIS.2011.5742388

34. Y. Tanahashi, C.-H. Hsueh, and K.-L. Ma. 2015. An Efficient Framework for Generating Storyline Visualizations from Streaming Data. *IEEE Transactions on Visualization and Computer Graphics* 21, 6, 730–742. DOI: http://dx.doi.org/10.1109/TVCG.2015.2392771

35. S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. 2014. Dynamic Network Visualization with Extended Massive Sequence Views. *IEEE Transactions on Visualization and Computer Graphics* 20, 8, 1087–1099. DOI: http://dx.doi.org/10.1109/TVCG.2013.263

36. M. Wattenberg. 2006. Visual Exploration of Multivariate Graphs. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. 811–819. DOI: http://dx.doi.org/10.1145/1124772.1124891

37. M. Wood. 2005. Bootstrapped Confidence Intervals as an Approach to Statistical Inference. *Organizational Research Methods* 8, 4, 454–470. DOI: http://dx.doi.org/10.1177/1094428105280059

38. J. wook Ahn, C. Plaisant, and B. Shneiderman. 2014. A Task Taxonomy for Network Evolution Analysis. *IEEE Transactions on Visualization and Computer Graphics* 20, 3, 365–376. DOI: http://dx.doi.org/10.1109/TVCG.2013.238

39. Y. Wu, N. Pitipornvivat, J. Zhao, S. Yang, G. Huang, and H. Qu. 2016. egoSlider: Visual Analysis of Egocentric Network Evolution. *IEEE Transactions on Visualization and Computer Graphics* 22, 1, 260–269. DOI: http://dx.doi.org/10.1109/TVCG.2015.2468151

40. J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan. 2013. Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12, 2080–2089. DOI: http://dx.doi.org/10.1109/TVCG.2013.167

41. J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. 2015. MatrixWave: Visual Comparison of Event Sequence Data. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*. 259–268. DOI: http://dx.doi.org/10.1145/2702123.2702419