BranchingSets: Interactively Visualizing Categories on Node-Link Diagrams

Francesco Paduano
Dept. of Computer Science
University of Illinois at Chicago
fpadua2@uic.edu

Ronak Etemadpour
Computer Science Dept.
Oklahoma State University
etemadp@cs.okstate.edu

Angus G. Forbes
Dept. of Computer Science
University of Illinois at Chicago
aforbes@uic.edu

ABSTRACT

Node-link diagrams are widely used for visualizing relational data in a wide range of fields. However, in many situations it is useful to provide set membership information for elements in networks. We present *BranchingSets*, an interactive visualization technique that uses visual encodings similar to *Kelp Diagrams* in order to augment traditional node-link diagrams with information about the categories that both nodes and links belong to. *BranchingSets* introduces novel user-driven methods to procedurally navigate the graph topology and to interactively inspect complex, hierarchical data associated with individual nodes. Results indicate that users find the technique engaging and easy to use. This is further confirmed by a quantitative study that compares the effectiveness of the visual encodings used in *BranchingSets* to other techniques for displaying set membership within node-link diagrams, finding our technique more accurate and more efficient for facilitating interactive queries on networks containing nodes that belong to multiple sets.

1. INTRODUCTION

Node-link diagrams are a popular way to visually represent data elements and the relationships between them. However, it can also be relevant to visualize the categories that each node or link is a member of. For instance, it could be meaningful to indicate the original source of particular elements when multiple data sets are merged together; nodes from different datasets could be grouped into different categories and visually distinguished based upon their origin. A more complex, real-world example comes from the domain of systems biology, where an important task involves analyzing signaling pathways in order to understand the biochemical interactions between cellular components. These complex pathway structures may need to be pieced together from multiple experiments drawn from various publications or databases, and it can be helpful to the biologist to provide a clear idea of the provenance of the various elements and relationships in the pathway [24]. Dynamic graphs could also benefit from a clear representation of which set or sets links and nodes are members of. A user might be interested in which links and nodes appear, disappear, or remain over a sequence of time [4]. A particular range of time could be represented by a different category, and a link or node could be part of this set if it overlapped that span of time.

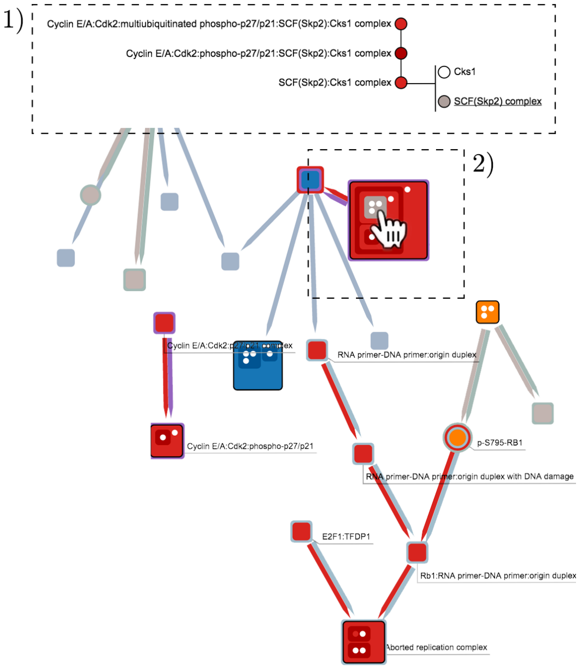

1)
Cyclin E/A:Cdk2:multiubiquitinated phospho-p27/p21:SCF(Skp2):Cks1 complex
Cyclin E/A:Cdk2:phospho-p27/p21:SCF(Skp2):Cks1 complex
SCF(Skp2):Cks1 complex
Cks1
SCF(Skp2) complex
2)
Cyclin E/A:Cdk2:p27/p21 complex
RNA primer-DNA primer:origin duplex
p-S795-RB1
Cyclin E/A:Cdk2:phospho-p27/p21
RNA primer-DNA primer:origin duplex with DNA damage
E2F1:TFDP1
Rb1:RNA primer-DNA primer:origin duplex
Aborted replication complex

Figure 1: A prototype application that uses the *BranchingSets* technique to facilitate the navigation of a complex, hierarchical biological pathway dataset where both nodes and links are members of one or more sets. A user can interactively inspect a protein complex to better see relevant information about the hierarchical structure of the data. Here, we see the "pruned tree" pop-up that provides details about the hierarchical structure of the complex (1) and the complex that the user is inspecting (2).

A range of solutions have been proposed to provide visual clues about the set membership of data elements, such as *Venn Diagrams*, *Euler Diagrams* [23], the *UpSet* visualization [18], *PivotPaths* [11], *RadialSets* [2], *TimeArcs* [9], and many others, as delineated in recent survey papers [3, 26]. Itoh et al. [16] introduce *multiple-category graphs* which are

VINCI '16, September 24-26, 2016, Dallas, TX, USA
© 2016 ACM. ISBN 978-1-4503-4149-3/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2968220.2968229

defined using a space-filling algorithm coupled with a force-directed layout algorithm. *MapSets* [12] uses a geographic metaphor to visualize embedded and clustered graphs. Techniques such as *Bubble Sets* [6], *LineSets* [1], *KelpFusion* [21], and *Kelp Diagrams* [10] are designed to be overlaid on top of existing visualizations and can readily be adapted to provide additional information about node-link diagrams. These latter techniques also address the challenge of representing set membership even when multiple intersections exist.

Bubble Sets displays set relations using isocountours, which can produce problematic representations when an element belongs to many multiple sets, in some cases causing elements to appear as if they are included in sets that they are not members of. *LineSets* represents sets as smooth curves and uses distinct colors to indicate set membership. All the nodes that belong to the same set are connected by a single curved lines, and nodes which belong to multiple sets are located at the intersection of multiple lines. Identifying all nodes that belong to a given set can be achieved by finding the respective curve and then visually following all the nodes placed on the line, similar to looking up what subway stops are part of a particular subway line. Nodes that are members of multiple sets can be identified by looking for the nodes positioned at the intersections of two differently-colored curves. This solution generally offers better readability when sets overlap than *Bubble Sets*, but can look tangled when visualizing multiple sets. Xu et al. [28] make use of this technique to facilitate analysis of set membership of nodes embedded in graphs.

Similarly, *Kelp Diagrams* depict set relations over points with predefined positions. The layout provides a sophisticated algorithm to size and color data elements in order to maximize the aesthetic quality and clarity of the visual encodings indicating set membership. Although they have been applied to network data, such as metabolic networks, they have been evaluated primarily on geospatial data, where location has intrinsic meaning. Moreover, the linking, or "link space" between nodes is explicitly *not* representative of links in a node-link diagram, but rather an additional visual artifact introduced to clarify set membership (similar to *LineSets*).

KelpFusion uses continuous boundaries made by lines and hulls in order to make it easier to see category information, which might be more difficult to identify using the thinner lines that are used by *LineSets* and *Kelp Diagrams*. The visual appearance is generally comparable to or better than *LineSets* for many tasks, though this depends on the spatial arrangements of elements. Hulls are used to group together elements that are both spatially close and belong to the same set; they are less effective when applied to node-link diagrams that do not have an intrinsic spatial meaning.

Inspired by these approaches, and especially by *Kelp Diagrams*, we introduce *BranchingSets*, an interactive visualization technique that enables interaction with node-link diagrams overlaid with set membership information. We have evaluated our technique with a controlled user study which measures the performance of *BranchingSets* in comparison to *LineSets*, and *Bubble Sets* for different tasks using non-spatial datasets. Furthermore, we have collected qualitative feedback from users, many of whom indicate a preference for our technique in terms of readability and visual clutter, particularly with datasets involving multiple set intersections. Our contributions are as follows:

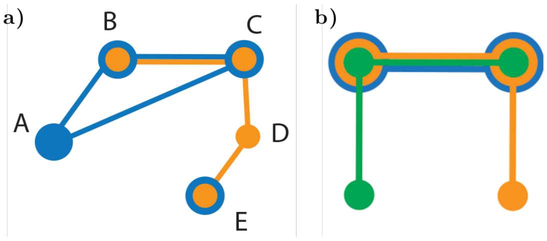


Figure 2: On the left (a), node E belongs to the blue category but is visually disconnected from the other blue nodes (A, B, and C) since no input or output links belonging to the blue category exist. On the right (b), we show *BranchingSets* using the "protruding" links option with a blue, green, orange color ordering, which may improve visual identification of nodes and links that share set memberships.

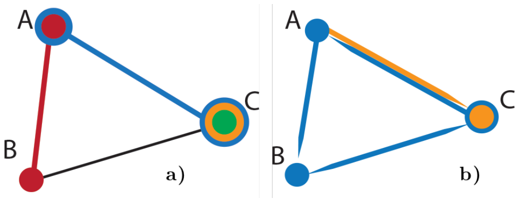


Figure 3: On the left (a), we show links with and without membership data; since the nodes B and C do not share a category, the link connecting them given a thin, black line. On the right (b), *BranchingSets* is applied to a directed node-link graph; the link between nodes B and C is bidirectional, the others are unidirectional.

- We introduce a set visualization technique similar to *Kelp Diagrams* and *LineSets* that is specifically adapted for node-link diagrams with embedded hierarchical data;
- We conduct a user study that evaluates how different techniques facilitate tasks relevant to set membership identification on node-link diagrams, and in particular queries involving multiple intersections;
- We provide a series of interaction techniques for exploring large datasets made possible by our technique, including exploring nested nodes and/or nodes augmented with further information and finding links between disparate nodes;
- We introduce a real-world application as a use case, enabling the investigation of multiple biological pathways with hierarchical data.

2. MOTIVATION AND DESIGN GOALS

The development of the interaction techniques and visual encodings used in *BranchingSets* were motivated by real-world use cases, primarily from projects that make use of very large datasets involving multiple biological pathway networks that contain: complex, hierarchically-nested nodes; redundant information; and links that themselves were usefully identified as members of one or multiple sets. Domain experts had a strong expectation of seeing their datasets using a network representation, yet bemoaned the difficulty of identifying patterns within the "hairball" of links. Based on a task analysis with these domain experts, we first created an interactive prototype specifically for their needs, but we believe that our approach is useful for a wider range of network data containing category information. Section 6 explores a real-world use case that uses *BranchingSets* to facilitate visual analysis tasks. Primary design goals include the following items, which serve as building blocks that enable more complex visual analysis tasks:

Allow users to identify the membership of each node and each link and the subgraphs composed of each node and/or link that belongs to a specified category (G1). We want to represent the membership of both nodes and of links. When the representation combines relational information from multiple data sources, it may be necessary to identify the data sources that contain a given node or link. Additionally, we want to be able to examine how a graph changes over time and to track the evolution of particular subgraphs over time.

Allow users to identify the subgraph resulting from the intersection or the complement of different categories (G2). Given two or more categories, a user should be able to identify the subgraph that makes up the *intersection* of two graphs (the portion shared between the categories) and also the *complement* subgraph (the graph that belongs to a given category but not to others).

Utilize the visual representation of node-link diagrams (G3). We want to design a solution that does not introduce additional visual clutter to the node-link diagram and also that does not require the user to become familiar with an entirely new visual metaphor. In certain application domains users are familiar with node-link diagrams and, for better or worse, skeptical of new data representations [20].

Reduce the complexity of the visualization (G4). We want to enable the user to interactively simplify the representation of a complex network as needed in order to display only information relevant to his or her research. That is, we recognize that even ideal visual encodings might not be sufficient to provide a readable visualization of a large, complex node-link diagram, and thus our technique allows the user to quickly toggle specified graphs or subgraphs on and off, or to interactively collapse a subgraph into a single node.

Enable interactive inspection of nodes and links (G5). Especially in cases in which datasets contain many categories, we want to leverage user interaction in order to facilitate effective exploration. For instance, a user may be interested in inspecting the contents or features of a node belonging to multiple categories, or to find all connections between two nodes that match a particular requirement.

3. VISUAL ENCODINGS

Here we provide a brief overview of the main visual encodings used in the *BranchingSets* technique and identify where they differ from those used in similar techniques. We assign a different color to each category, and additionally we define an ordering of the colors. As explained below, the order of colors is used to simplify the recognition of nodes which belongs to the same categories. If the categories have no intrinsic hierarchy, an ordering is arbitrary, but consistent. Each node is assigned to the color of the category it belongs to. If a node belongs to multiple categories, we use all the respective colors, but give the node the primary color that comes first in the colors ordering. A supplementary colored border is added to the node for every additional category, and the colors are assigned with respect to the colors ordering previously defined. This creates a visual correspondence between nodes which belong to the same categories: nodes that are members of exactly the same categories will present an identical sequence of colors; and nodes that share only a subset of categories will present the shared colors in the same order.

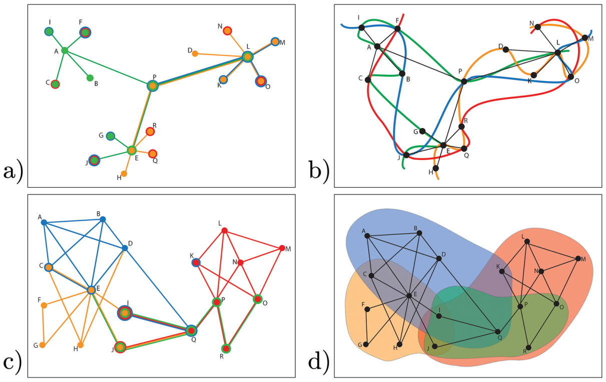


Figure 4: Examples of two datasets used in our study. On the top, (a) and (b) show a representation of the same dataset using both *BranchingSets* and *LineSets*. On the bottom, (c) and (d) each show another dataset using *BranchingSets* and *Bubble Sets*.

Although otherwise similar to *Kelp Diagrams*, *BranchingSets* also shows group information associated with links. Similarly to the strategy designed for visualizing node membership, links are colored with the corresponding color of the category. If the same link belongs to multiple categories, multiple lines are placed side by side, one for each color. This is different than the visual encoding used in *Kelp Diagrams*, which either uses a “nested style” or a “striped style” of coloring links; and of course the *Kelp Diagrams* technique does not aim to depict relationships in node-link diagrams, but instead uses links only to emphasize point set membership. Our technique is also reminiscent to one introduced recently by Lambert et al. [17], but rather than using colored hulls to surround the original links, we use the colors themselves to indicate both the links and their set membership.

Colored links have also been used to indicate connections between protein-protein interaction networks by an application that displays elements in the STRING database [13]. However, this application does not integrate the membership of links with nodes in the network. By default, *BranchingSets* represents links with line segments drawn underneath the nodes. This leads to the line segments “disappearing” behind the circular shape of the node. An alternative is to assign links and nodes of the same color to the same graphical layer [10], resulting in a continuity between the colored border of the node and the line segment that can be helpful for tasks involving distinguishing subgraphs that share categories. Fig. 2 shows a graph with and without the “protruding” links.

Fig. 3 presents an example of how *BranchingSets* can be used to show the grouping of links and nodes in directed graphs. Many techniques for visualizing a directed edge, such as tapered lines, gradients, or arrows [15], can reduce the surface of the link, which makes it more difficult to identify its color. Techniques that make use of color blending or an opacity gradient could similarly be difficult to use in conjunction with a color coding for visualizing categories. The thick tip of regular arrows introduces problems when multiple links must be placed side by side, as our technique does. For these reasons, we have adopted medium-sized line segments with a sharp tip to indicate direction. In case of a bidirectional link, the line segment has a sharp tip on both sides. Holten and Van Wijk [15] also explore the use of curved lines to indicate direction, and our technique also supports the use of curved lines as a way to quickly identify cycles within a graph or subgraph.

4. USER STUDY

We have conducted a user study to evaluate the performance of *BranchingSets* compared with *Bubble Sets* and *LineSets* on node-link diagrams. *BranchingSets* shares similar visual encodings with *Kelp Diagrams*, and our results for *BranchingSets* could also be applied to *Kelp Diagrams* for the tasks we evaluate here (that is, in cases where links are not assigned to categories). As evaluations of the efficacy of visual encodings for cardinality, membership, and intersection tasks have been studied previously (in other contexts), our study replicates aspects of these previous ones in order to confirm results on non-spatial datasets. A quantitative study measures the accuracy and the completion time for a number of tasks across different levels of complexity (defined below); additionally, a qualitative study examines the users' subjective evaluation in terms of ease of comprehension and visual clutter. Although this study does not include the interaction components of *BranchingSets* (which are introduced in Section 5), we wanted to confirm that the fundamental visual encodings were effective for displaying multiple set intersections on node-link diagrams. Moreover this study provides evidence for results that are not directly addressed in previous work. In Alper et al. [1], *LineSets* is evaluated in comparison only to *Bubble Sets*, and is evaluated on both maps as well as social networks. In Dinkla et al. [10], *Kelp Diagrams* explicitly aims to provide a less cluttered visualization than *LineSets*, but with similar functionality, but does not include a quantitative user study. Moreover, although the technique can be applied to arbitrary node-link networks, it is discussed primarily in terms of geospatial datasets. In Meulemans et al. [21], *KelpFusion* is evaluated against both *LineSets* and *Bubble Sets*, but is intended only to augment map visualizations using continuous boundaries. This study thus presents a new evaluation of three distinct set visualization techniques on non-spatial node-link diagrams, and also explores differences between simpler and more complicated queries about intersections.

In order to perform the controlled user study we recruited 17 undergraduate or graduate students with backgrounds in computer science or computer engineering. The age of the users ranged from 20 to 34, with a mean age of 24, and the gender distribution was 9 females and 8 males. For each study we generated node-link diagrams with 17 nodes and assigned category information to each node in the network. Although we used only a small number of nodes, this number was sufficient for exploring how differences in *complexity* affected task completion. The diagrams ranged in complexity along three parameters: the total number of categories a node could belong to (number of intersections); the density of the network (number of links between nodes); and the overall cohesion of the categories (whether or not similarly colored elements appear clustered together or are randomly distributed across the network). The diagrams were randomized across these parameters, with between 21 and 40 links and between 0 and 4 intersections for any give node; each participant saw the same set of diagrams but in a random order. Tasks involving multiple intersections have not been explicitly evaluated previously, and facilitating complex questions on networks with properties of real-world datasets was an important goal of our investigation. It is important to note that *BranchingSets* allows edges to belong to different categories. Since this functionality is not enabled in the other visualizations, we colored edges with the same color as the nodes they connected. For instance, if a vertex A and B were both assigned to the same two categories, we used two colored links to indicate the multiple set membership of the edges. Figs. 4a–d presents samples of the images shown during the user study. Specifically, Figs. 4a and b show one example dataset using two different representations and Figs. 4c and d show another example dataset using two different representations.

Questions for each task (described below) were asked in a randomly shuffled order and were answered using a multiple-choice format. Before starting the test, we made sure that every user had sufficient background knowledge to perform the test. Some students were not at first completely familiar with graph layouts containing set information, and everyone was given a short training session that described the fundamental concepts of node-link diagrams and set membership. Following this short introduction, all the users were introduced specifically to the *Bubble Sets*, *LineSets*, and *BranchingSets* technique by showing and explaining images representing the same dataset with different techniques. Users were allowed to ask questions to solidify their understanding of how set membership was represented on each of the techniques, and the administrator of the test checked their understanding with brief questions. Though we told the users that the study meant to evaluate different techniques, users were unaware that the *BranchingSets* visual encodings were the focus of our research. The study was performed on a 15" monitor, and the figures were each 1000x800 pixels in size. The users took time to get familiar with the interface before starting the study. Furthermore, when a new question was displayed on the monitor the user was able to read the question and the possible answers before starting the task. For each of the 12 questions, the figure was hidden and the measurement of the time taken to complete a task started only when the user themselves chose to reveal the image.

We based our quantitative study on four common tasks that can be performed on graphs with categories. These tasks follow the study found in Alper et al. [1], but differentiates between simple intersection tasks and more complex tasks involving multiple intersections. Table 1 lists the tasks and a sample question for each task; Tasks 1-4 are evaluated over three different techniques: *Bubble Sets*, *LineSets* and *BranchingSets*.

4.1 Hypotheses

We believe that the continuous smooth curves of *LineSets* will perform comparably to the visual encoding of *BranchingSets* for both T1 and T2.

(H1a) No significant difference will be found between the vi-

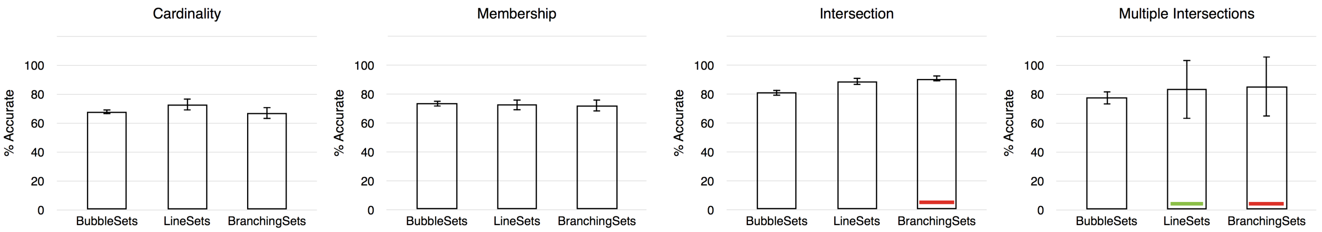


Figure 5: Correctness in percentage (bar charts show mean error and standard error from the mean): results of comparing the three techniques on the four tasks considered. There is statistical significance for multiple intersections and intersection tasks. The horizontal lines encode pairwise statistical significance using a red and green colors. The red line indicates a significant difference between *BranchingSets* and *Bubble Sets*; the green line shows *significant difference between LineSets and Bubble* Sets.

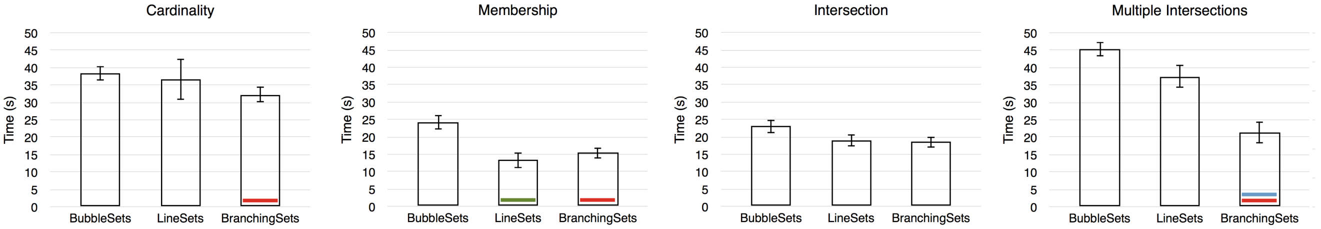


Figure 6: Time in seconds (bar charts show mean error and standard error from the mean): results of comparing the three techniques on the four tasks considered. There is statistical significance for multiple intersections, cardinality, and membership tasks. The horizontal lines encode pairwise statistical significance using a red and green colors. The red line indicates a significant difference between *BranchingSets* and *Bubble Sets*; the green line shows *significant difference between LineSets and Bubble* Sets; the blue line indicates significant difference between *BranchingSets* and *LineSets*.

Table 1: Tasks used in the quantitative experiments

T1:	Cardinality	"Which group contains the most nodes?"
T2:	Membership	"Which nodes are in the red group?"
T3:	Intersection	"Which nodes are contained in both the red and green group?"
T4:	Multiple Intersections	"Which nodes are contained in exactly 3 groups?"

sual encodings used in *LineSets* and *BranchingSets* in terms of either accuracy and completion time for the cardinality task (T1).
(H1b) No significant difference will be found between *LineSets* and *BranchingSets* in terms of either accuracy and completion time for the membership task (T2).

Alper et al. previously demonstrated that *LineSets* is superior to *Bubble Sets* for intersection tasks; our results should confirm their findings. Although *LineSets* has been used to visualize items belonging to large sets, the resulting output can be quite tangled. *BranchingSets*' visual encoding of multiple sets will enable users to more effectively identify set intersections (T3). Specifically, we expect users to more quickly and accurately query the network for questions involving multiple intersections (T4).

(H2a) *BranchingSets* will outperform *LineSets* and *Bubble Sets* in terms of accuracy for intersection tasks (T3, T4).
(H2b) *BranchingSets* will outperform both *LineSets* and *Bubble Sets* in terms of time for intersection tasks (T3, T4). Further, we expect users to qualitatively respond more positively to *BranchingSets*' layout of node-diagrams with multiple intersections than either *Bubble Sets* or *LineSets*.

(H3) *BranchingSets* will rate higher during intersection tasks than both *LineSets* and *Bubble Sets*.

4.2 Results

Several aspects were considered for the statistical analysis of the results of the user study. First, we compared the three methods for each of the tasks by looking into the mean errors over all subjects and all data sets. Second, we did the same comparisons considering the time it took the participants to fulfill the tasks. Third, we compared the three different methods against each other for an evaluation of visual clutter; the participants were asked to state their confidence about the visual clutter task using a Likert scale (1 to 5). For all analyses, we computed means and standard deviation of the errors. To test for statistical significance of the individual results, we first tested the distribution of the error values against normality using the Shapiro-Wilk [25] tests. In case of non-normal distribution, we applied non-parametric Friedman test [14] on K related samples when comparing more than two groups. If the computed differences were significant, we performed pair-wise comparisons of the groups using a Wilcoxon test [27] on non-parametric related samples to be able to report which groups particularly differ from each other. In case of normal distribution, we used an ANOVA test when comparing more than two groups. For pair-wise comparisons in case of more than two groups we ran a series of Tukey's post-hoc tests and Holm's sequential Bonferroni adjustment at the 0.05 level.

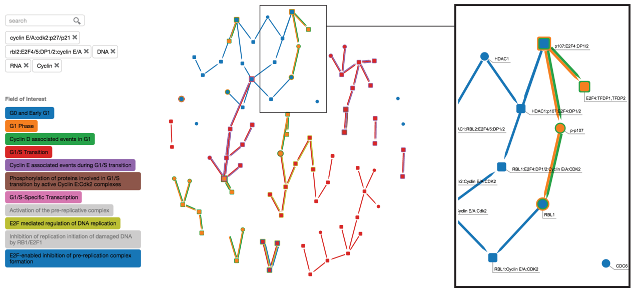
Figs. 5 and 6 summarize the results of the comparative analysis of the three different methods for each of the four tasks. The bar charts show the mean error values and the standard error from the mean. The omnibus tests for significance showed that there is statistical significance in the mean errors for some of the tasks. The outcome of the pair-wise significance test is indicated by the red, green, and blue horizontal lines: if pair-wise comparison between *BranchingSets* and *Bubble Sets* methods showed a significant result, the line is colored *red*; if pair-wise comparison between *Bubble Sets* and *LineSets* showed significance, the line is colored *green*; if pair-wise comparison between *BranchingSets* and *LineSets* showed significance, the line is colored *blue*.

For T1, "Cardinality", the Friedman test showed significant difference ($\chi^2(2, 17) = 6.118, p < 0.047$) among the three methods. Applying the Bonferroni adjustment across pair-wise Wilcoxon comparisons showed significant differences only between *Bubble Sets* vs. *BranchingSets* ($Z = -2.059, p = 0.039$). Therefore, *BranchingSets* is faster than *Bubble Sets* significantly but is as fast as *LineSets* for determining the cardinality of sets and thus **H1a** is confirmed.

For T2, "Membership", the Friedman test showed significant difference ($\chi^2(2, N = 17) = 15.176, p < 0.001$) among the the three methods. Applying the Bonferroni adjustment across pair-wise Wilcoxon comparisons showed significant differences between *Bubble Sets* vs. *BranchingSets* ($Z = -3.385, p = 0.001$) and *Bubble Sets* vs. *LineSets*($Z = -3.101, p = 0.002$). Thus, for this task *Bubble Sets* method is slower significantly than other methods. However, there is no significant difference between *LineSets* and *BranchingSets* and **H1b** is confirmed.

As Fig. 5 shows, for T4, "Multiple Intersections", the Friedman test showed significant difference ($\chi^2(2, N = 17) = 22.194, p < 0.05$) among three methods in terms of accuracy. Applying the Bonferroni adjustment to pair-wise Wilcoxon comparisons showed significant differences between *Bubble Sets* vs. *BranchingSets* ($Z = -3.432, p = 0.001$) and *Bubble Sets* vs. *LineSets* ($Z = -2.795, p = 0.005$). Therefore, *Bubble Sets* has the least significant accuracy compared to the other techniques for this task. However, there is not any significant difference between *LineSets* and *BranchingSets*. For T3, "Intersection", the Friedman test showed significant difference ($\chi^2(2, N = 17) = 3.250, p < 0.05$) among the three methods. Bonferroni across pair-wise Wilcoxon comparisons showed significant differences only between pair-wise comparisons *Bubble Sets* vs. *BranchingSets* ($Z = -2.363, p = 0.018$). Thus, **H2a** is partially confirmed.

As Fig. 6 illustrates, for T4, "Multiple Intersections", the Friedman test showed significant difference ($\chi^2(2, N = 17) = 12.706, p = 0.002$) among the three methods in terms of time. Applying the Bonferroni adjustment across pair-wise Wilcoxon comparisons showed significant differences between *Bubble Sets* vs. *BranchingSets* ($Z = -3.290, p = 0.001$) and *BranchingSets* vs. *LineSets* ($Z = -2.959, p = 0.003$). Therefore, *BranchingSets* is faster significantly compared to the other techniques for this task. However for T3, "Intersection", there was no significant differences between the three techniques. Thus, **H2b** is partially confirmed.


search
cyclin E/A:cdk2:p27/p21 ✕
rbl2:E2F4/5:DP1/2:cyclin E/A ✕
DNA ✕
RNA ✕
Cyclin ✕

Field of Interest

G0 and Early G1
G1 Phase
Cyclin D associated events in G1
G1/S Transition
Cyclin E associated events during G1/S transition
Phosphorylation of proteins involved in G1/S transition by active Cyclin E:Cdk2 complexes
G1/S-Specific Transcription
Activation of the pre-replicative complex
E2F mediated regulation of DNA replication
Inhibition of replication initiation of damaged DNA by RB1/E2F1
E2F-enabled inhibition of pre-replication complex formation

Figure 7: A detail of interactively-selected *Rb-E2F1* subgraphs visualized with our prototype *BranchingSets* application for biological pathway networks.

In addition to these quantitative tasks, we also asked users to answer qualitative questions about each technique involved in the study. The questions measure user preferences in terms of visual clutter using a Likert scale from 1 to 5. We were interested especially to learn whether or not users considered it easy to comprehend the layout and visual encodings used by our technique. Users noted that the visual encoding of *Bubble Sets* is generally easier to understand than *LineSets* and *BranchingSets*, whereas *LineSets* and *BranchingSets* were found to be comparable. Users overall indicated that they found our technique less cluttered than *Bubble Sets*. The average grade was 4.6 with a mode of 5. After having evaluated the correctness of the results and subjects' task fulfilling times, we investigate the subjects' confidence. When looking into the average confidence values for data sets, the Friedman test showed statistically significant difference among three methods ($\chi^2(2, N = 17) = 19.143, p < 0.05$). Applying Bonferroni across pair-wise Wilcoxon comparisons showed significant differences between *Bubble Sets* vs. *BranchingSets* ($Z = -3.542, p = 0.00035$) and *BranchingSets* vs. *LineSets* ($Z = -2.811, p = 0.005$). That is, *BranchingSets* method is perceived as significantly less cluttered for all tasks compared to the other methods and thus **H3** is confirmed.

In summary, our hypotheses about *BranchingSets* proved correct for T1 and T2, but only partially correct for T3 and T4. However, we note that *BranchingSets* was as accurate as *LineSets* for T4, while being significantly faster. Thus we conclude that our technique provides an appropriate visual encoding on which to build our interactive techniques for real-world applications.

5. INTERACTION TECHNIQUES

Our user study confirms that the visual encodings used in *BranchingSets* are more effective for that *LineSets* and *Bubble Sets* for tasks involving multiple intersections. However, we believe that any technique will have difficulty representing large networks featuring a large number of categories. For instance, using color to represent sets may introduce visual confusion when multiple sets are displayed simultaneously. In the remainder of this paper we introduce interaction techniques that mitigate the complexity of representing a large number of categories on large networks. These interaction techniques, coupled with our ability to represent multiple set intersections effectively, facilitate a range of real-world analysis tasks. *BranchingSets* provides user interactions for exploring, extending, and inspecting hierarchical nodes within a complex network. The interaction tasks, discussed below, include:

- Highlighting and labelling categories;

- Hiding and unhiding categories;
- Filtering by keyword;
- Expanding neighbors of a selected node;
- Navigating and rearranging the layout;
- Finding intermediate steps between two nodes.

By default all node labels are hidden. This improves the usability and avoids the inconvenience of too many labels that might overlap and clutter the representation. When the user hovers with the mouse pointer on a node, only the labels of the nodes which are included in the same category are displayed. Moreover, the category which includes the hovered component stands out because all the nodes and links which does not belong to that category change to a gray color. If a node which is included in more than one category is hovered, all the involved categories will be highlighted.

The user can dynamically explore the data by hiding or revealing different categories. A given node or link is visible if it belongs to at least one visible category, and it shows the memberships related to these categories only. We also enable users to search the network by typing one or more words in a text field. The visualization will display only the nodes whose labels matches at least one of the searched keywords. The ability to search dynamically for keywords and filter categories, together with the methods described below, enables an efficient way to progressively explore large datasets.

By clicking on a node the visualization will be expanded to display all the nodes connected to it. This interaction enables the following workflow: the user searches for a set of nodes of interest, then he or she progressively explores the network of interconnections by revealing the neighbors. The visualization of large datasets can easily exceed the boundary of the screen. For this reason we enable the user to interactively zoom in, zoom out and pan across the viewport. Since the representation may lead to edges that intersect, the user can drag and drop any component to a preferred location to improve visibility.

In addition to clicking to reveal connected neighbors, a user can expand the set of visible nodes and links in another manner. When dragging the mouse from one node to another while holding the right mouse button, the visualization updates by showing all sequences of nodes and links that start from the first node and end with the last. In directional graphs, only the sequences of links from the first node to the target node will be revealed, though additional paths could added through subsequent user interaction. Some applications of this interaction could include: revealing the biological elements between two proteins a biochemical pathway, showing the stages of an industrial process in a workflow diagram, or displaying the sequence of web-pages that can be visited in order to navigate from one page to another.

6. EXPLORING BIOLOGICAL PATHWAYS

Biological pathways are used to represent the chain of interactions in a biological process. They describe the biochemical functionality of proteins in a cell. Biologists need a visual representation of biological pathways to support a number of tasks, such as: understanding the network topology of the pathway; enabling the inspection of the hierarchical structure of pathway elements; understanding relationships between different pathways; and making hypothesis about pathways [24]. Different visualization techniques have been adopted to face these challenges. Typically, pathways are represented as directed graphs, where nodes represent biological *participants*, such as *proteins* or *biological complexes*, and edges represent a biological functionality, such as a biochemical reaction. Visualization tools for pathway analysis include *Entourage* [19], *Reactome Pathway Browser* [7], *ReactionFlow* [8], and others that utilize node-link diagrams. *ChiBe* [5] extends the node-link representation by displaying compound nodes that indicate the composition of complexes. Although this approach is similar to our implementation [22], when multiple pathways are displayed *ChiBe* does not show the correspondence between a component and the pathways it belongs to.

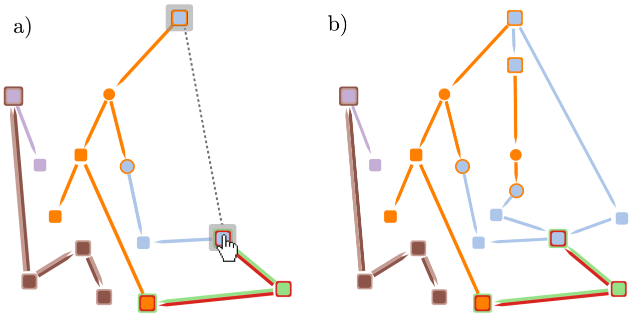


Figure 8: On the left, the user drags the mouse from one biological complex to another. On the right, the visualization is updated with all connecting paths.

We used *BranchingSets* to visualize and enable the interactive exploration of multiple pathways simultaneously [22]. The complexity of these pathways was mitigated using our application, and biologists were able to more effectively filter, search, and compare biological pathway data. Our prototype application can be accessed via our GitHub repository.[1] Fig. 7 shows a subset of participants and reactions included in the *Rb-E2F1* pathway, as found in the Reactome database.[2] The visualization aims to provide a better understanding of the biological components and reactions shared among different sub-pathways included in *Rb-E2F1*. We assigned a different color to each sub-pathway, and circles and rectangles were used to represent *proteins* and *biological complexes*, respectively. The directed links between nodes represent biochemical reactions.

The user can interactively choose to focus on a limited set of sub-pathways to reduce the amount of information and the number of different colors displayed at a given time. We implemented the user-driven interactions described above, enabling a procedural exploration of the network while reducing visual clutter. Fig. 8 shows the user interacting with the visualization to discover all the intermediate steps between two participants. Because our system reduces visual clutter that can occur in other set visualization techniques, we can introduce new visual elements that may be useful for displaying additional information. For instance, *biological complexes* are made up of nested groups of proteins. Double-clicking on a node expands it within the node-link diagram, allowing the user to inspect the hierarchically-nested protein data layered within the complex. Fig. 1 shows an example of what one of these expanded nodes looks like, along with an

[1] https://github.com/CreativeCodingLab/BranchingSets
[2] http://reactome.org/PathwayBrowser/#/453279

accompanying "pruned tree" visualization that locates specific proteins within the hierarchy of the protein complex.

7. CONCLUSIONS AND FUTURE WORK

In this paper we described *BranchingSets*, an interactive visualization technique that shows category information within node-links diagrams. We applied our technique to a dataset containing multiple biological pathways comprised of complex hierarchical data. *BranchingSets* is specifically designed to be integrated with node-link diagrams, which leads to more concise, less cluttered representations compared to other techniques that show the category information within a spatial structure. Additionally, our technique can display the set membership of links, either separate from or in conjunction with nodes. Furthermore, *BranchingSets* includes a set of user-driven techniques for interactively exploring, extending, and inspecting a complex network. These interactions have proven to be helpful when delving into large datasets.

Additionally, we presented a user study that demonstrated that the visual encodings used in *BranchingSets* are effective at helping users recognize similar patterns and to identify differences between graphs as well as to identify nodes that belong to multiple categories. For other tasks, such as simple intersection or membership tasks, our technique was comparable to *LineSets*. Even when there was no quantitative difference is accuracy or speed, users indicated that our solution produces representations which are more easily comprehended and less cluttered when compared with *Bubble Sets* or *LineSets*. As discussed, our results should, in certain cases, apply to *Kelp Diagrams*, which inspired the main visual encodings of our technique. We plan to conduct further studies to explore how our solution scales when we increase the number of nodes, connections, categories, and intersections. Moreover, we plan to evaluate the effectiveness of the user-driven interactions to manage the exploration of large datasets in a range of application domains.

8. ACKNOWLEDGEMENTS

This publication is supported by the DARPA Big Mechanism Program under ARO contract WF911NF-14-1-0395. The authors thank Basak Alper, Paul Murray, and especially Kasper Dinkla for their contributions and helpful feedback.

9. REFERENCES

[1] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski. Design study of LineSets, a novel set visualization technique. *IEEE Trans. Vis. Comp. Graph.*, 17(12):2259–2267, 2011.

[2] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser. Radial sets: Interactive visual analysis of large overlapping sets. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2496–2505, 2013.

[3] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. The State-of-the-Art of Set Visualization. *Comp. Graph. Forum*, 35(1):234–260, 2016.

[4] D. Archambault, H. C. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Trans. Vis. Comp. Graph.*, 17(4):539–552, 2011.

[5] O. Babur, U. Dogrusoz, E. Demir, and C. Sander. ChiBE: Interactive visualization and manipulation of BioPAX pathway models. *Bioinformatics*, 26(3):429–431, 2010.

[6] C. Collins, G. Penn, and S. Carpendale. Bubble Sets: Revealing set relations with isocontours over existing visualizations. *IEEE Trans. Vis. Comp. Graph.*, 15(6):1009–1016, 2009.

[7] D. Croft, A. F. Mundo, R. Haw, M. Milacic, J. Weiser, G. Wu, M. Caudy, P. Garapati, M. Gillespie, M. R. Kamdar, et al. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 42(D1):D472–D477, 2014.

[8] T. Dang, P. Murray, J. Aurisano, and A. G. Forbes. ReactionFlow: An interactive visualization tool for causality analysis in biological pathways. *BMC Proc.*, 9(6):S6, August 2015.

[9] T. Dang, N. Pendar, and A. G. Forbes. TimeArcs: Visualizing fluctuations in dynamic networks. *Comp. Graph. Forum*, 35(3):61–69, 2016.

[10] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg. Kelp Diagrams: Point set membership visualization. *Comp. Graph. Forum*, 31(3.1):875–884, 2012.

[11] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais. PivotPaths: Strolling through faceted information spaces. *IEEE Trans. Vis. Comp. Graph.*, 18(12):2709–2718, 2012.

[12] A. Efrat, Y. Hu, S. G. Kobourov, and S. Pupyrev. MapSets: Visualizing embedded and clustered graphs. In *Proc. of Graph Drawing*, pages 452–463. Springer, 2014.

[13] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering, et al. STRING v9.1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):D808–D815, 2013.

[14] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Amer. Stat. Assoc.*, 32:675–701, 1937.

[15] D. Holten and J. J. van Wijk. A user study on visualizing directed edges in graphs. In *Proc. ACM Conf. on Human Factors in Comp. Sys. (CHI)*, pages 2299–2308, 2009.

[16] T. Itoh, C. Muelder, K.-L. Ma, and J. Sese. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *Proc. of the IEEE Pacific Vis. Symp.*, pages 121–128, 2009.

[17] A. Lambert, F. Queyroi, and R. Bourqui. Visualizing patterns in node-link diagrams. In *IEEE Int. Conf. Info. Vis.*, pages 48–53, 2012.

[18] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. Upset: visualization of intersecting sets. *IEEE Trans. Vis. Comp. Graph.*, 20(12):1983–1992, 2014.

[19] A. Lex, C. Partl, D. Kalkofen, M. Streit, S. Gratzl, A. M. Wassermann, D. Schmalstieg, and H. Pfister. Entourage: Visualizing relationships between biological pathways using contextual subsets. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2536–2545, 2013.

[20] G. E. Marai. Visual scaffolding in integrated spatial and nonspatial analysis. In *Proc. of the EuroVis Workshop on Visual Analytics (EuroVA)*, pages 13–17, 2015.

[21] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer. Kelpfusion: A hybrid set visualization technique. *IEEE Trans. Vis. Comp. Graph.*, 19(11):1846–1858, 2013.

[22] F. Paduano and A. G. Forbes. Extended LineSets: A visualization technique for the interactive inspection of biological pathways. *BMC Proc.*, 9(6):S4, August 2015.

[23] N. H. Riche and T. Dwyer. Untangling Euler diagrams. *IEEE Trans. Vis. Comp. Graph.*, 16(6):1090–1099, 2010.

[24] P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Info. Vis.*, 4(3):191–205, 2005.

[25] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[26] C. Vehlow, F. Beck, and D. Weiskopf. The State of the Art in Visualizing Group Structures in Graphs. In *Eurographics Conf. on Vis. (EuroVis) - STARs*, pages 21–40, 2015.

[27] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[28] P. Xu, F. Du, N. Cao, C. Shi, H. Zhou, and H. Qu. Visual analysis of set relations in a graph. *Comp. Graph. Forum*, 32(3.1):61–70, 2013.