

Group-Level Graph Visualization Taxonomy

Bahador Saket, Paolo Simonetto, and Stephen G. Kobourov

Department of Computer Science, University of Arizona, Tucson, AZ

Abstract

Task taxonomies for graph and network visualization focus on tasks commonly encountered when analyzing graph connectivity and topology. However, in many application fields such as the social sciences (social networks), biology (protein interaction models), software engineering (program call graphs), connectivity and topology information is intertwined with grouping and clustering information. Several recent visualization techniques, such as BubbleSets, LineSets and GMap, make explicit use of grouping and clustering, but evaluating such visualizations has been difficult due to the lack of standardized group-level tasks. With this in mind, our goal is to define a new set of tasks that assess group-level comprehension. We propose several types of group-level tasks and provide several examples of each type. Finally, we characterize some of the proposed tasks using a multi-level typology of abstract visualization tasks. We believe that adding group-level tasks to the task taxonomy for graph visualization would make the taxonomy more useful for recent graph visualization techniques. It would help evaluators define and categorize new tasks, and it would help generalize individual results collected in controlled experiments.

1. Introduction

Graphs are used to describe a set of entities (nodes) and their relationships (edges). Graphs and networks (used interchangeably here) are typically visualized using a node-link diagram, where nodes are depicted as points, and edges as line segments connecting the corresponding points.

Several studies have tested the readability of node-link diagrams. In particular, Purchase [Pur97] examined how graph drawing aesthetics such as edge crossings and display of symmetries impact performance of graph reading tasks, such as path tracing. Huang et al. [HE05, HEH08, HEH14] used eye-tracking and control experiments to understand visual network perception. Perceptual characteristics and memorability in dynamic, animated graphs have also been studied [GEY12, FQ11, AP12].

The results of these studies are difficult to compare, because of the absence of a standardized approach to graph evaluation studies. In an attempt to mitigate this problem, visual data analysis tasks can be organized and categorized in taxonomies. Brehmer and Munzner [BM13] reviewed and compared a large number of earlier studies, and in doing so provided a schema that blends with the existing taxonomies and allow to fully characterize motivation, methods, and input/output information for each task.

Although task taxonomies are available for a broad

range of visualization techniques, including node-link diagrams [LPP*06], none specifically deal with the visualization of graphs with additional grouping information. Graphs where nodes are grouped based on a priori knowledge or structural properties occur in many different domains: from information spatialization [SF03], to recommendations in a recommender system [GHKV09], to self organizing maps of documents, coupled with geographic information systems [Sku02], to maps of general science [BKB05] and maps of computer science [FK14]. It has been shown that augmenting node-link diagrams with spatial features can improve graph revisitation tasks [GE11]. This is used in visualizations that explicitly draw boundaries to indicate the grouping: BubbleSets [CPS09], LineSets [ARRC11], and GMap [GHK10]; see Fig. 1.

We begin with a brief review of existing task taxonomies for graph visualizations, and the multi-level typology of abstract visualization tasks. In Section 3 we augment the task taxonomy for graph visualization by introducing a new set of tasks related to groups of nodes (Group-Level Tasks) and we characterize a couple of the newly proposed tasks using the multi-level typology of abstract visualization tasks [BM13]. In Section 4 we discuss the potential value of the proposed group-level task taxonomies for graph visualization and show how these group-level tasks can be used in future evaluation studies.

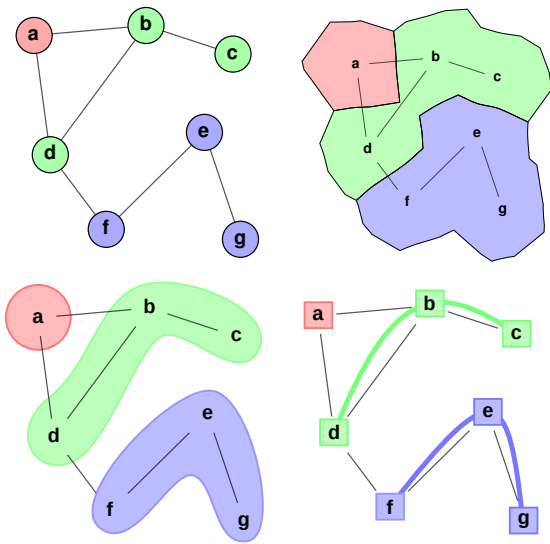


Figure 1: Different group representations. **Top-Left:** in node-link diagrams, groups are often encoded with node colors. **Top-Right:** in GMaps, groups resemble countries of a geographical maps. **Bottom-Left:** in BubbleSets, iso-contours are drawn around nodes belonging to same set. **Bottom-Right:** in LineSets, groups are identified by colored nodes and links. Note that the figures in the bottom row were manually created to mimic the look of these methods.

2. Task Taxonomies for Graph Visualization

Brehmer and Munzner [BM13] organize the vast previous work on task taxonomies in visualization, highlighting their advantages and disadvantages. They point out as the major shortcoming of most approaches, the lack of a global view of the task: high-level categories often ignore how the tasks are performed, while low level categories often ignore why the tasks are performed. In order to close this gap, they develop a multi-level typology that helps create a complete description of a task. Here *typology* indicates a more abstract categorization based on concepts, rather than *taxonomy*, which is a categorization of pre-existing objects (tasks in this case). As shown in Fig. 2, this multi-level typology encompasses three main questions: WHY, HOW and WHAT.

The work of Brehmer and Munzner, however, is not meant to replace model-oriented taxonomies, but rather to “encompass and complement these specific classification systems”. In fact, topic-specific taxonomies provide details about low-level tasks, that are necessarily abstracted in the very general approach of Brehmer and Munzner. Instead, they provide the tools to put these low level tasks in context, guiding the evaluation designer in providing information such as user expertise and motivation.

A couple of existing taxonomies served as foundations for

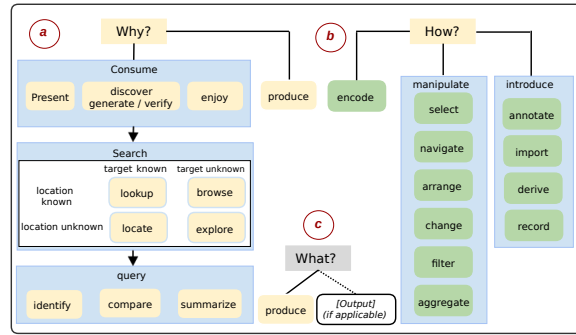


Figure 2: Multi-level typology of abstract visualization tasks. a) WHY is a task performed, from high-level (consume vs. produce) to mid-level (search), to low-level (query)? b) HOW is a task executed, in terms of methods, defined as families of related visual encoding and interaction techniques? c) WHAT are the task inputs and outputs? Figure from [BM13] used with permission.

our work. Amar et al. [AES05] describe a set of ten primitive analysis task types, representative of the kinds of specific questions that one may ask when working with tabular data (e.g., Retrieve Value, Find Extremum, etc.). The task taxonomy for graph visualization developed by Lee et al. [LPP*06] is built upon these tasks, but the authors found that it was necessary to define additional low-level tasks, such as scan and set operations, and the graph-specific low-level task: “find adjacent nodes”. The final set of tasks was then organized into four groups: topology-based tasks, attribute-based tasks, browsing tasks, and overview tasks.

3. Task Taxonomy for Clustered Graph Visualization

We defined a list of 31 group-level tasks both from studying the user interaction with visualizations such as BubbleSets [CPS09], LineSets [ARRC11] and GMap [GHK10], and from interviewing experts in the field. We divided these group-level tasks into four subcategories according to the information required to solve them.

- **Group Only Tasks:** Tasks in this category can be performed by only considering the groups, so that no node or edge information is required. **For example:** Find the set of group-neighbors of a given group.
- **Group-Node Tasks:** Tasks in this category can be performed by only considering group and node information. **For example:** Find the group with the maximum number of nodes.
- **Group-Link Tasks:** Tasks in this category can be performed by only considering group and edge information. **For example:** Count the number of links in a given group.
- **Group-Network Tasks:** Tasks in this category can be performed by only considering group, node and edge information. **For example:** Find the groups which contain cycles.

Group Only Tasks

Find the set of group-neighbors of a given group. — Find groups with specific characteristics (e.g., red background). — How many groups are neighbors of a given group? — Which group has the maximum (minimum) number of neighboring groups? — Find the set of groups that are reachable from a given group — How many groups are reachable from a given group? — Given two groups, find the set of common group-neighbors. — Find a set of groups whose removal would make group Y unreachable from group X. — Find the group with largest (smallest) visual area. — Find a group with largest (smallest) perimeter. — Navigate from group X to another group Y and then to group Z. — Find a specific characteristic (e.g., shape) of group X. — Are the given two groups neighbors? — How many groups are there?

Group-Node Tasks

List groups which contain nodes with specific characteristics (e.g., labels beginning with the letter “A”). — Count the number of nodes in a given group. — Find the group with the maximum (minimum) number of nodes. — Given two nodes X and Y and check whether they belong to the same group. — Given a node V (with specific characteristics), find the group which contains V.

Group-Link Tasks

Count the number of links in a given group. — Find the group with the maximum (minimum) number of links. — Find the most sparsely (most densely) connected group. — List groups which contain a link with specific characteristics (e.g., longest, heaviest). — Find the group with the smallest (largest) number of links connecting it nodes outside the group.

Group-Network Tasks

Find the groups which contain cycles (k -cliques, degree- k nodes). — Find two groups with a link between them, whose removal disconnects the network. — Find the shortest path between a node in group X and a node in group Y. — Find a group which has the node with highest (lowest) degree. — Find the path U-V-W; are nodes U and W in the same group? — Given two nodes in different groups X and Y, what is the smallest number of groups that must be visited on a path from X to Y? — Find the minimum number of nodes (edges) whose removal disconnect two given groups.

Table 1: Group Level Tasks

Table 1 shows several group-level tasks divided by category. Most of the tasks are intuitively clear, but we need to clarify a couple of more subtle issues required in order to accommodate different group visualization metaphors.

- **Group neighboring:** we define neighbors as an abstract concept, which can be made concrete for most visualizations, e.g., in GMaps two groups are neighbors if they share a common boundary, while in BubbleSets two groups are neighbors if they share a common boundary, or if their regions intersect.
- **Group reachability:** we say a group is reachable from another group if it is possible to move from one to the other by moving through neighboring pairs of groups.
- **Group area:** geometric notions such as area and perimeter are clearly defined for GMaps and BubbleSets, but open to interpretation for LineSets and node-link diagrams.

We note that the tasks under consideration can be used as given here, or can be combined in macro tasks. For simplicity, we focus on simple undirected graphs, but most tasks naturally generalize to directed and/or weighted graphs. Still, our list is not exhaustive, meaning that is surely does not cover all the tasks of a given type. However, the tasks that we provide can be used both for the definition of new tasks, and for the definition of more specific taxonomies. For example, Jianu et al. [JRHT14] defined the tasks used in their

evaluation based on the taxonomy of Lee et al. [LPP*06], which in turn is based on the even earlier taxonomy of Amar et al. [AES05].

3.1. Relationship to Graph-Level Tasks

Many group-level tasks can be deduced as an extension of graph-related tasks. In fact, we can consider the groups as metanodes, define metaedges according to the presence or absence of connections between the original nodes, and execute graph tasks on the metagraph generated by this process. Many graph characteristics, such as degree, adjacency or centrality of a node directly translate into relevant group properties. For instance, consider the graph of the connections of a courier company in Figure 3. We can analyze the graph to deduce that the best way to ship a parcel from the US to India require a transit through UK and Italy, but we can also analyze the metagraph to see that all parcels form North-America to Asia need to transit through Europe.

This metagraph can be built in several different ways, leading to different insights on the data. For example, in representations where the groups share their boundaries (e.g. GMap), the metagraph can be built based on the group contacts. This would be useful, for instance, to identify the groups that have stronger interconnections with each other.

Tasks	WHY	WHAT	HOW
Find the set of group-neighbors of a given group.	The purpose of the task is to discover groups that are neighbors of a given group. The target is known (the given group). If the participants are aware of the location of the given group then this is a Lookup task; otherwise this is a Locate task; see Fig. 2a). Then the participant searches for the group-neighbors and then lists these groups. Discover + (Look up + Locate) + Summarize	The input for the task is a given group. The output is a list of all group-neighbors of the given group.; see Fig. 2c). Input: A given group Output: List of groups	The participant must identify all group-neighbor of a given group (to tell groups X and Y apart and to check whether they have a common boundary or another group is in between). Select
Find the group with the maximum number of nodes.	The purpose of the task is to discover the group with the most nodes. Since the group identity (e.g., color) is not given, the target is unknown. This is an Explore task as the participant must search for the group (location is unknown), followed by an Identity task. Discover + Explore + Identify	The input for the task is all the groups, including their nodes and links. The output is the identity of the group with the most nodes. Input: Entire visualization Output: A group	The participant needs to count (estimate) the number of nodes for each group and keep track of the largest one. Derive + Select

Table 2: Examples of group-level tasks described using the multi-level typology of abstract visualization tasks

3.2. Examples of Task Description

In Table 2, we provide a couple of examples of how our tasks can be described according to the typology of abstract visualization tasks [BM13]. We can only provide examples rather than a full description in terms of WHY, HOW and WHAT because this requires information unavailable to us and known only by the evaluation designer. For example, participant who are asked to find a node X might or might not be aware of their approximate position, or might have different motivations for executing the task.

3.3. Limitations and Benefits

The current taxonomy can be extended by considering more general inputs, or by expanding the set of defined tasks. For example, we assumed simple, undirected graphs with non-overlapping, one-level clustering. However, overlapping groups, directed relationships, multi-level hierarchical clustering are common in many applications such as social

networks (where one can participate in multiple communities), organization diagrams and taxonomies (where each group is further subdivided into subgroups and the relationships are directional). Although hierarchical clustering can be considered as a subset of overlapping groups where partial overlaps are forbidden, considering overlapping groups would likely necessitate additional tasks to deal with the increased number of possible operations and the increased drawing complexity.

We believe that our taxonomy has two main benefits: 1) it can help in categorizing possible tasks performed when analyzing a specific visualization; 2) it can help ensure that participants in evaluation studies perform a wide spectrum of tasks. Such a taxonomy would also be applicable to experiments which assess clustered graphs [AES05, APP10] and map-like visualizations [JRHT14]. We have also used this taxonomy in an evaluation study comparing node-diagrams, node-link-diagrams, and node-link-area diagrams [SSKB14].

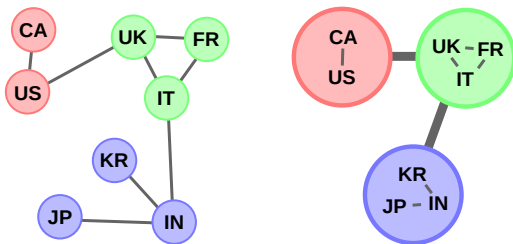


Figure 3: Construction of a metagraph. *Left:* the original clustered graph. *Right:* the metagraph.

4. Conclusions and Future Work

Our primary contribution is in defining a taxonomy of group-level tasks. These tasks can be used to design and describe the operations that can be performed on a clustered graph, and should be used in conjunction with the work of Brehmer and Munzner to fully characterize an evaluation study. We showed this process using two examples. We recently used this taxonomy in a study to evaluate the effectiveness of point-cloud, node-link, and map-based visualizations, and we plan to use it a more general study evaluating map-like visualizations. As future work we would also like to extend this taxonomy to accommodate directed graphs, overlapping groups, and hierarchical clustering.

References

- [AES05] AMAR R., EAGAN J., STASKO J. T.: Low-level Components of Analytic Activity in Information Visualization. In *Symposium on Information Visualization (InfoVis '05)* (2005), pp. 111–117. [2](#), [3](#), [4](#)
- [AP12] ARCHAMBAULT D., PURCHASE H. C.: The Mental Map and Memorability in Dynamic Graphs. In *Pacific Visualization Symposium (PacificVis)* (2012), pp. 89–96. [1](#)
- [APP10] ARCHAMBAULT D., PURCHASE C. H., PINADU B.: The Readability of Path-Preserving Clustering of Graphs. *Computer Graphics Forum (EuroVis10)* 29, 3 (June 2010), 1173–1182. [4](#)
- [ARRC11] ALPER B., RICHE N. H., RAMOS G., CZERWINSKI M.: Design Study of Linesets, a Novel Set Visualization Technique. In *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2011), pp. 2259–2267. [1](#), [2](#)
- [BKB05] BOYACK K. W., KLAVANS R., BÖRNER K.: Mapping the backbone of science. *Scientometrics* 64 (2005), 351–374. [1](#)
- [BM13] BREHMER M., MUNZNER T.: A Multi-level Typology of Abstract Visualization Tasks. In *Symp. Information Visualization (InfoVis '13)* (2013), pp. 2376–2385. [1](#), [2](#), [4](#)
- [CPS09] COLLINS C., PENN G., S. C.: Bubble Sets: Revealing Set Relations with Isocontours Over Existing Visualizations. In *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2009), pp. 1009–1016. [1](#), [2](#)
- [FK14] FRIED D., KOBouROV S. G.: Maps of Computer Science. In *7th IEEE PacificVis Symposium* (2014). To appear. [1](#)
- [FQ11] FARRUGIA M., QUIGLEY A.: Effective Temporal Graph Layout: A Comparative Study of Animation Versus Static Display Methods. *Information Visualization* 10, 1 (2011), 47–64. [1](#)
- [GE11] GHANI S., ELMQVIST N.: Improving Revisitation in Graphs Through Static Spatial Features. In *Proceedings of Graphics Interface 2011* (2011), GI '11, pp. 175–182. [1](#)
- [GEY12] GHANI S., ELMQVIST N., YI J. S.: Perception of Animated Node-link Diagrams for Dynamic Graphs. *Computer Graphics Forum* 31, 1 (2012), 1205–1214. [1](#)
- [GHK10] GANSNER E. R., HU Y., KOBouROV S. G.: Visualizing Graphs and Clusters as Maps. In *IEEE Computer Graphics and Applications* (2010), pp. 2259–2267. [1](#), [2](#)
- [GHKV09] GANSNER E., HU Y., KOBouROV S., VOLINSKY C.: Putting recommendations on the map - visualizing clusters and relations. In *Proc. 3rd ACM Conference on Recommender Systems* (2009), pp. 345–354. [1](#)
- [HE05] HUANG W., EADES P.: How People Read Graphs. In *Asia-Pacific symposium on Information visualisation* (2005), pp. 51–58. [1](#)
- [HEH08] HUANG W., EADES P., HONG S. H.: Beyond Time and Error: a Cognitive Approach to the Evaluation of Graph Drawings. In *conference on BEyond time and errors: novel evaluation methods for Information Visualization* (2008). [1](#)
- [HEH14] HUANG W., EADES P., HONG S. H.: Measuring Effectiveness of Graph Visualizations: A Cognitive Load Perspective. In *Information Visualization* (2014), pp. 139–152. [1](#)
- [JRHT14] JIANU R., RUSU A., HU Y., TAGGART D.: How to Display Group Information on Node-Link Diagrams: an Evaluation. In *IEEE Trans. Visualization and Computer Graphics (TVCG)*, To Appear (2014). [3](#), [4](#)
- [LPP*06] LEE B., PLAISANT C., PARR C., FEKETE J. D., HENRY N.: Task Taxonomy for Graph Visualization. In *the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods For information Visualization (BELIV '06)* (2006), pp. 81–85. [1](#), [2](#), [3](#)
- [Pur97] PURCHASE H. C.: Which Aesthetic has the Greatest Effect on Human Understanding? In *International Symposium on Graph Drawing (GD97)* (1997), Di Battista G., (Ed.), vol. 1353 of *Lecture Notes in Computer Science*, Springer, pp. 248–261. [1](#)
- [SF03] SKUPIN A., FABRIKANT S. I.: Spatialization Methods: a Cartographic Research Agenda for Non-Geographic Information Visualization. *Cartography and Geographic Information Science* 30 (2003), 95–119. [1](#)
- [Sku02] SKUPIN A.: A Cartographic Approach to Visualizing Conference Abstracts. *IEEE Computer Graphics and Applications* 22, 1 (2002), 50–58. [1](#)
- [SSKB14] SAKET B., SIMONETTO P., KOBouROV S., BÖRNER K.: Node, Node-Link, and Node-Link-Group Diagrams: An Evaluation. arXiv:1404.1911 [cs.HC], Apr. 2014. [4](#)