

Visual Querying and Exploring of Large Multilayer Graphs

Erick Cuenca*
LIRMM, University of
Montpellier, CNRS

Arnaud Sallaberry†
LIRMM, University Paul
Valéry Montpellier 3, CNRS

Dino Ienco‡
IRSTEA,
UMR TETIS

Pascal Poncelet§
LIRMM, University of
Montpellier, CNRS

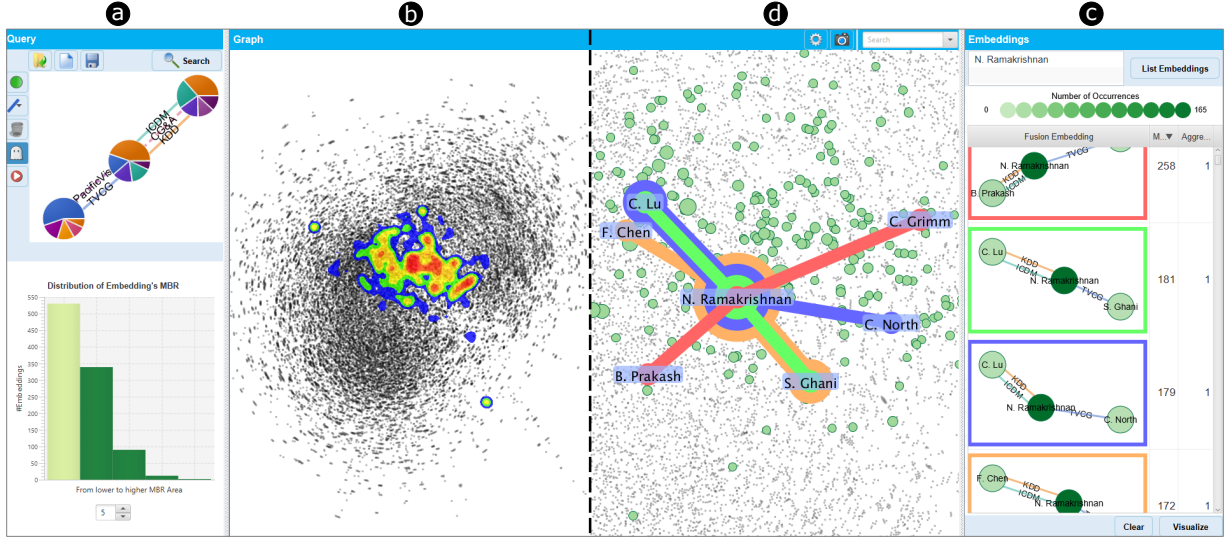


Figure 1: Our approach being used to explore a co-authorship network. (a) The Query view allows to build the query and start the query process to retrieve the results. This view also supports query suggestions represented by pie charts. (b) The Graph view shows an overview of the result locations using a heatmap representation. (c) The Embeddings view shows a list of results for a set of selected entities (e.g. the results involving N. Ramakrishnan). (d) The Graph view displays result relations as Kelp-based diagrams.

ABSTRACT

Many real world data can be modeled by a graph with a set of nodes interconnected to each other by multiple relationships. Such a rich graph is called multilayer graph. We present a novel visual platform to query, explore and support the analysis of large multilayer graphs. Our approach provides coordinated views to navigate and explore the large set of retrieved results at different level of granularities. In addition, the proposed system supports the refinement of the query by visual suggestions to guide the user through the exploration process. A demo video is available at <https://youtu.be/zray-iR8Hz0>.

Index Terms: visual graph querying, visual graph suggestions, multilayer graphs

1 INTRODUCTION

A multilayer graph allows the definition of multiple types of relationships (edges) between vertices [1]. An example of a multilayer graph is a bibliographic network where nodes are authors and the edge type spans over the set of conferences or journals in which researchers have published [1]. Considering such structure to model real world information, designing visual tools to support the analysis of large multilayer graphs is essential. Unfortunately, none of the current visual graph query systems [7] can deal with multilayer

graphs. Additionally, their result exploration are limited since they do not support exploration of the results on the original graph.

In this work, we present a new visual platform that allows to query large multilayer graphs, visualize/explore retrieved results and suggest new query extensions based on the underlying multilayer graph structure and the current query results [2]. Our system not only allows to visually draw the query and browse the corresponding results, but it also supports the interaction between the user and the underlying query process. The user can start, pause, and resume the query engine with the possibility to navigate/explore partially collected results. Finally, the results can be analyzed at different levels of detail.

2 VISUAL MAPPINGS AND INTERACTIVE FUNCTIONALITY

This section describes the three main visual components of our approach: a *Query view*, a *Graph view*, and an *Embeddings view*.

2.1 Query View

With the aim to query the graph, we conceive an interactive graphical interface to visually construct a query (Fig. 1a). This view allows the user to define the query structure (nodes, edges, node-attributes) via interactive functionalities. This interface provides different actions to build the query such as: add a node, add an edge of a specific layer, delete a node/edge, and launch the edge suggestion mechanism. From this view, the user can run the query engine process.

2.1.1 Starting the Query Engine Process

To deal with the different relationships involved in a multilayer graph, we integrate a recent multi-graph query engine named SuMGrA [5]. This query engine exploits specialised indexing techniques in order

*e-mail: erick.cuenca@lirmm.fr

†e-mail: arnaud.sallaberry@lirmm.fr

‡e-mail: dino.ienco@irstea.fr

§e-mail: pascal.poncelet@lirmm.fr

to speed up the backtracking algorithm that is commonly employed to deal with the subgraph isomorphism problem [5].

2.1.2 Visualizing Query Suggestions

Based on the initial query structure, our approach automatically suggests k edges to the user’s query intention by ranking all the adjacent edges for every result node and then suggest the top- k frequent type of edges found. The graph structure is used to obtain the adjacent edges for each result node.

Our approach suggests two types of edges: i) **internal edges** that link nodes that are already in the query structure and ii) **external edges** that link nodes of the query to other ones. Internal edges are represented by dashed lines (Fig. 2(a,b) PacificVis and CG&A links). Since external edges convey extension to an external node, and with the aim to depict a k number of edge-suggestions, we use a pie chart (Fig. 2c). The number of slices in the pie chart is equal to k . The percentage displayed on a slice indicates how many edges in the graph support the extension of the query node by the particular edge type (Fig. 2c indicates that 7% of the results already found can be extended with a SIGMOD edge type).

In order to accept a suggested edge, the user clicks over the internal or external edge. A new internal edge is displayed as a continuous line (Fig. 2(d, e)). A new external edge is displayed as a link to a node in the direction of its slice (Fig. 2f SIGMOD link).



Figure 2: At the left, visual elements on the query depict suggested edges. At the right, the query including newly added suggested edges.

2.2 Graph View

We adopt a node-link diagram with a force direct layout [6] to produce a suitable visualization of the graph (Fig. 1b). We leverage a multilevel technique named Multipole Multilevel Method (FM3) [4] that allow to scale up to large graphs with a good tradeoff between time and visualization performances.

2.3 Embeddings View

This view allows the user to visualize the list of results (Fig. 1d) containing a given set of nodes. The goal of this interface is to allow the user to sample the results for any combination of nodes.

3 INTERACTIONS

Fig. 3 shows the interactions between the visual components in our approach. The user starts the visual query process by building the query structure (Fig. 3a). Next, the user sends the query to the engine (Fig. 3b, with the possibility to start, pause, and resume the query engine). Next, the query engine requests the graph structure in order to retrieve the results (Fig. 3(c,d)) and returns them to the Graph view. Based on the retrieved results and the graph structure, the visual query mechanism suggests k edges in order to refine the previous query and re-execute it (Fig. 3j).

Since the number of results can be large, our tool enables two different levels of analysis: i) an **overview level** (Fig. 3e) in which it sums up the results using a heatmap representation and ii) a **detailed level** (Fig. 3f) in which it allows the user to inspect particular nodes involved in the results (Fig. 3g). From the list of results (Fig. 3h),

the user can select up to five different results that will be visualized on the Graph view by using a Kelp-based approach [3] (Fig. 3i).

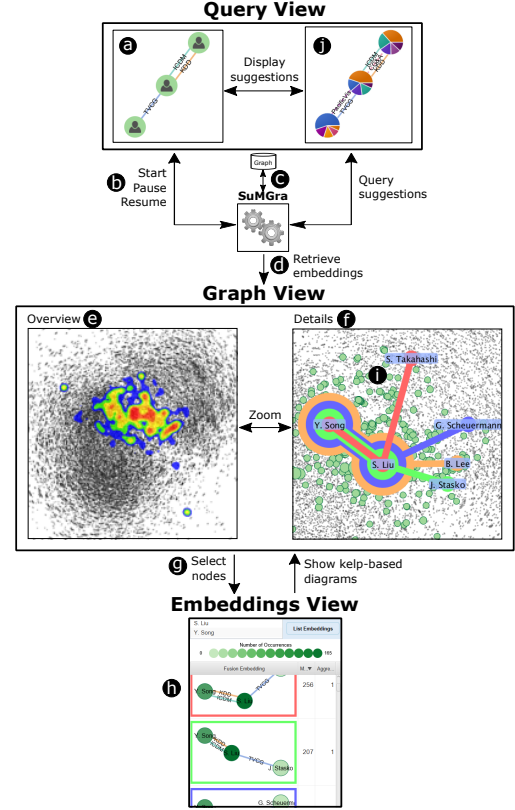


Figure 3: System architecture of our approach. Arrows show interactions between components.

4 CONCLUSION

In this work, we presented a new visual platform that allows to query multilayer graphs, visualize/explore the obtained results and suggest new query extensions based on the underlying multilayer graph and the current query results. The results are analyzed at different levels of detail and the visual components support the user to focus on particular interesting areas for further inspection. The approach also enables the interaction between the user and the query process. The user can start, pause, and resume the query engine with the possibility to navigate/explore partially collected results.

REFERENCES

- [1] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen. Distance oracles in edge-labeled graphs. In *Proc. EDBT*, pp. 547–558. ACM, 2014.
- [2] E. Cuenca, A. Sallaberry, D. Ienco, and P. Poncelet. Visual Querying of Large Multilayer Graphs. In *Proc. SSDDBM*, pp. 32–34. ACM, 2018.
- [3] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg. Kelp Diagrams: Point Set Membership Visualization. *Computer Graphics Forum*, 31(3pt1):875–884, 2012.
- [4] S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proc. GD*, pp. 285–295. Springer, 2004.
- [5] V. Ingalalli, D. Ienco, and P. Poncelet. SuMGra: Querying Multigraphs via Efficient Indexing. In *Proc. DEXA*, pp. 1–15. Springer, 2016.
- [6] M. Kaufmann and D. Wagner. *Drawing graphs: Methods and models*. Springer, 2003.
- [7] R. Pienta, F. Hohman, A. Endert, A. Tamersoy, K. Roundy, C. Gates, S. Navathe, and D. H. Chau. VIGOR: Interactive Visual Exploration of Graph Query Results. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):215–225, 2018.