Layer Definition and Discovery in Multilayer Network Datasets

Fintan McGee* Ludovic Morin† Mickaël Stefas‡ Simone Zorzan§ Mohammad Ghoniem¶

Luxembourg Institute of Science and Technology (LIST)

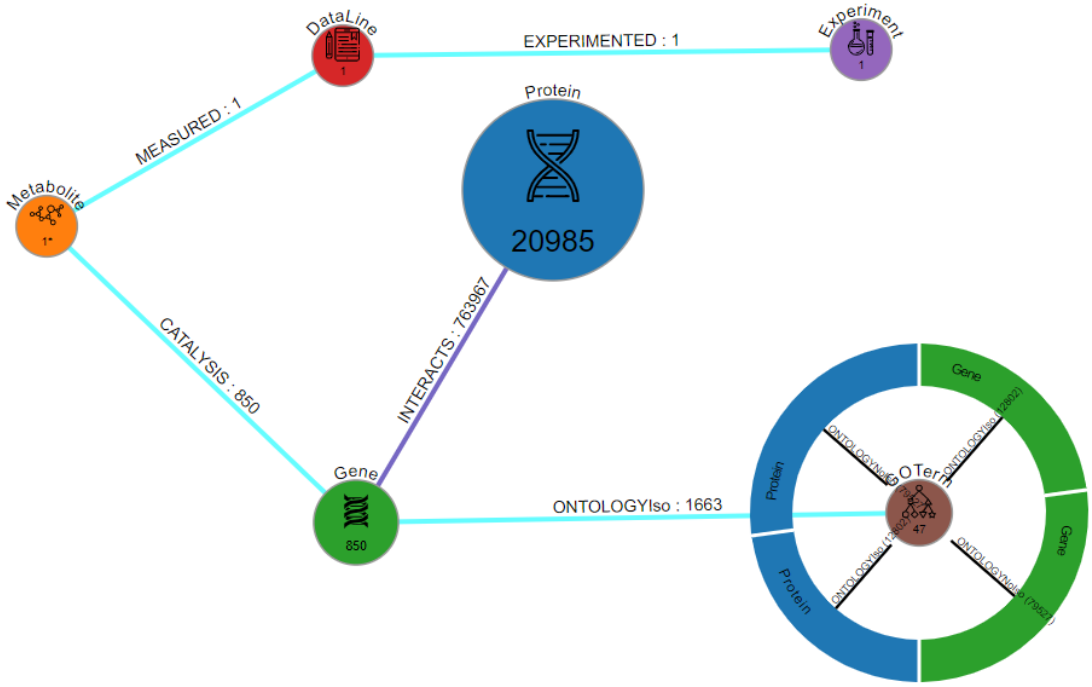

Figure 1: An example of the meta-network of the visual query builder for extracting working datasets from a massive network database. In this case, a biologist has queried for a specific metabolite. She has then queried for experiment data items that relate to the metabolite and then the experiments itself (the top branch). She has also queried for genes related to those metabolites (in the lower branch), and then the proteins related to those genes. She has also queried for Gene Ontology (GO) terms related to the genes. In the above image the radial menu, by which user queries can be easily expanded, has been opened on the gene ontology node. The user can further include 2 different types of node, each with two different types of edge, hence the four visible segments.

Abstract

The real world systems modelled by multilayer networks are frequently characterised by a high level of complexity. Understanding the most suitable definition of layer that can help a user is a significant challenge. Due to the scale of many datasets, even finding the entities and relationships that are of interest to the user is difficult. In this work we describe the issues with working with large multilayer data sources, based on the development of a multilayer network visualization prototype. We present an in-progress prototype and discuss future research directions for creating multilayer networks visualization of data extracted from massive sources.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Graph Drawing; Human-centered computing—Visualization—Visualization design and evaluation methods

*e-mail: fintan.mcgee@list.lu
†e-mail: ludovic.morin@list.lu
‡e-mail: mickael.stefas@list.lu
§e-mail: simone.zorzan@list.lu
¶e-mail: mohammad.ghoniem@list.lu

1 Introduction

Multilayer networks, as a concept, emerged from the field on complex networks as an alternative for modelling real-world systems [15]. They are an approach to dealing with complexity. By introducing the new concept of layer within the graph model, interactions between systems can be studied, and subsystems of interest can be extracted. The complexity of the systems being modelled can result from dependencies between systems, such as power-grids and data networks [6] or from more traditional issues such as scale. The use of layers in a data model can clarify the relationships between sub-systems, and also allow for the division of large data sets into more conceptually manageable sub-systems. As discussed by McGee *et al.* [19] the definition of layers, and the aspects which characterise them, is still an important open challenge. Modern data sets can contain millions of entities and billions of edges (e.g. [25]). To even begin answering a question a user has to extract a *"working"* data set, which is one that does not contain any irrelevant data, and contains all possible data items relevant to the user task at hand. In the case of multilayer networks, the definition of layers is also closely related to the definition of this initial data set. The focus of this paper is on trying to solve this problem of the end user being overwhelmed by a the size and complexity of initial datasets.

We approach this problem by using interactive menus in a visual query interface, that allow the user to quickly determine the size and structure of their desired dataset, before retrieving it. We present an early stage prototype of our approach, and we also discuss how this approach can be a first step for layer specification, a key challenge in the visual analytics of of multilayer networks.

2 Background and Related Work

Within in the field of visualization there are many existing systems which extract data from sources to build large networks (see [19] for an extensive survey of information visualization work that can be considered multilayer network visualization). ORION [14] is an application that can be considered multilayer, as its data is characterised as "heterogeneous" (see [15] for a full list of graph types and synonyms that fall under the multilayer framework). Amongst other features, the system allows users to query tables of a database, via a tabular interface. The user can view the schema of a relational DB and use link tables together to build networks.

The PLOCEUS application of Liu *et al.* [17] also creates networks from tabular data, providing operations such as projection and aggregation that are not available in ORION. The output of this tool is a new network based on the tabular data. As the application source data for both PLOCEUS and ORION is table based (tables are a source of node data), edges are not first class citizens in the application (however, some metrics can be added to them). Cuenca *et al.* [9] take a more visual approach to the querying of large multilayer networks. In their approach the data is already in the form of a network. The goal of their approach is not to generate an initial data set so much as to find matching patterns in a large dataset. A heat map is used to visualise where there are structural matches to the visual query in the target graph.

The Graphiti application of Srinivasan *et al.* [23], is a tool that allows a user to model a network by showing an example of the information they want from a input tabular dataset . This "demonstration based interaction" involves specifying edges between nodes, and then based on attributes (and conditions on those attributes) of the edge end-points (i.e. the nodes), a new network is created based on all the edges that can be induced from the input dataset. The goal of this visual query approach is to simplify the modeling and updating of datasets, removing the need for text based queries and the specification of constraints that require an existing knowledge of a data set. Despite edges being the focus of creation, Graphiti is still very much a node-centric application, as the edges are created based on node properties

Another approach by which networks have been created is the extraction of entities from a text using Named Entity Recognition (NER), as done by the Jigsaw application [24]. In Jigsaw, different types of entities are related based on co-occurrence within the text. The different types of entities can be considered to make up different layers, making this one of the earlier multilayer network visualization applications in the domain of information visualization. None of the approaches above integrates the concept of a layer as an entity into their model creation.

Although not a multilayer technique, the PivotGraph approach of Wattenberg [26] provides a novel way of summarising a network visually. The approach does not scale to large data sets, but the concept of grouping sets of graph vertices and edges in a single visual node or link is a useful technique for visualizing the structure of a dataset.

Heer and Perer [14] report that repeating the level of effort for building a single network to answer questions raised through the exploratory process may undermine and even dissuade analysts from testing all hypotheses. Speed and ease of use are important factors for any new technique or application for creating or extracting networks, especially in the case where multiple networks or layers may need to be extracted. The importance of ease of creation of networks can also be seen in commercial products such as centrifuge [1] or TouchGraph Navigator [3], where network creation and data discovery are seen as major selling points.

2.1 Scale of data sets

As is to be expected for a topic that has emerged from the field of complex systems, multilayer data sets are usually not trivially small. For example, the digital humanities network data set extracted from a research corpus, described by [18], contained over 140,000 nodes and over 1 million edges. The domain of life sciences is coming to recognise the importance of multilayer networks in addressing the many challenges of the field [12]. The data sets within this domain are vast. The STRING database of protein-protein interactions [25] alone contains over 24 million entities and 2 billion interactions [2]. Investigations in the rising field of Systems Biology raise questions spanning multiple such databases, accentuating the challenge of scale. The Tulip application [5] allows some basic support for multilayer networks (in that it allows multiple node types) and is capable of displaying very large graphs. However, even with its focus on large graphs, a database the size of STRING is too much to load and display visually. There is no query interface that allows the data set to be explored within the application, without loading it in its entirety.

2.2 Encoding of multilayer structure

There is no standard format or API for encoding multilayer networks for an application. There are several implementations such as MuxViz [10] and py3plex [22] which provide data visualization functionality and use their own data structure. The underlying data structure to encode a multilayer network, depends very much on the tasks at hand. However, in general an underlying multivariate multi-edge graph structure, will cover most cases.

2.3 Storage of multilayer data sets

The recent arrival of graph databases means that graph data no longer needs to be stored in a tabular format [17] or in relational databases [14]. Such formats can often reduce the importance of edges and not recognise them as important entities in their own right. For example ORION infers edges based on foreign keys between tables. In a multilayer context there may be multiple edges between data, and edges themselves may characterise whether or not entities (either nodes or edges) belong to a specific layer. Graph databases are considered mature enough and suitable for use in multilayer network domains such as biology [13]. Storing graph data in a graph database allows for more simple and direct querying of graph structure and within graph databases both graph nodes and edges are considered first-class citizens. Yet, we are not aware of any graph-oriented database where layers are encoded as an entities in their own right. In current graph database implementations, possible encodings of layers include adding them as nodes with links to their member nodes and edges, or adding them as labels of these nodes and edges.

3 Visual Querying of Large Multilayer Data Sets

3.1 Design motivation

As part of a project working with both large digital humanities and biological data sets, defining the subset of data of interest for visualization was a challenge. Both types of expert users had different sets of requirements and different structures of data.

The digital humanities data set was generated using NER on a large corpus of documents concerning the formation of the European project post 1945. The digital humanities researchers use case focused on browsing a large corpus to understand the relationship between historical entities over time [18]. The biological dataset contained data integrated from several biological sources, as well as the experimental data of expert users, amounting to over 350,000

nodes and 46 million edges. In addition to this publicly available data the biologists also have their own experiment data. The biologists required a system that allowed them to analyse this data in the context of the existing publicly available data, from databases such as STRING [25]. In both use-cases the size and density of the data made it difficult for the users to determine a "working" set of data to even begin their exploration and analysis. The goal of the work described in this paper is to help them easily define their working set of data without being overwhelmed, so they could start to analyse and explore their data.

Defining a network structure is not a concern for either of our datasets. Both data sets were stored in a graph database so, unlike many of the examples in Section 2, the data are already in a network form. Edges in each data set also contained attributes, and they were not simply defined by their end-points. A user will want to define an immediate "working" data set where irrelevant data has been discarded, before they address the specific concerns of their use case and application domain. Our goal was to give expert users self-service access to the contents of the graph database by allowing them to sketch the entities and relationships of interest to the task at hand. And, as they specified their needs, it was important to quickly allow them to get an idea of the size and structure of their potential working dataset.

3.2 The query meta-network

In our approach a user selects a node type from their data set as their starting point in the definition of their working set. This approach aligns with the existing workflow used by expert users in both of our application domains in that the will always know at least what type of entity they are interested in starting their exportation with. All instances of this entity will be represented by a meta-node in a meta-network that is built visually by the user. A query is run on the back end graph DB to immediately provide the system with information about what attributes are available for the particular entity. When a meta-node is added it displays the amount of underlying data nodes represented by the meta-node (as can be seen in Figure 1).

3.3 Query constraints

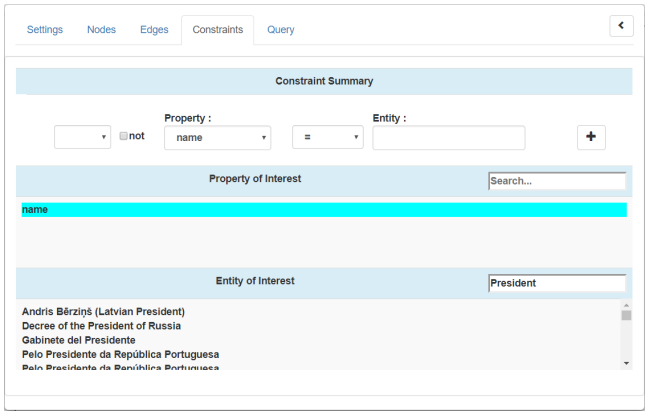

Figure 2: An example of the constraint menu for the digital humanities data set. Here nodes of the type person are being constrained to only include those with "president" in their title.

The user can then apply constraints to the meta-node, via a menu interface to limit the number of underlying data nodes represented by the meta-node (see Figure 2).

3.4 The radial menu

Meta-edges can be added in a similar manner, by clicking on the source and target meta-node and applying constraints. However, it is more intuitive for the user to use the radial menu [7] associated with the meta-node.

Figure 3: The radial menu as it appears for a single document in the Digital Humanities data set.

Clicking on a meta-node entity allows a user to see the other entities related to that original one, as well as the proportion of relations for each type, and the type of relationship. As can be seen in Figure 1 and Figure 3, the edge type and node type are clearly displayed, with different segments of the radial menu depending on the combination of edge and node type. This approach lends itself to the rapid exploration of a database structure, allowing the user to build queries with no prior knowledge of the relationships between entities.

3.5 Current layer specification approach

While the definition of the network structure is not the responsibility of the user of the system in our case, the definition of layers is, as it very often will be dependent of the task at hand. Layers are entities in and of themselves, however they may not necessarily be explicitly defined within the data store. For example in the case of a digital humanities data set with a temporal aspect, the definition of layers based on time period depends on the entities of interest to the user. If a user is interested in events over a five year period, a layer covering a timespan of a decade is of little use.

The current prototype focuses on specifying the working data set, and the specific layering required by each use case has been defined externally, through interviews with the users. When the dataset is retrieved from the back end system, it is automatically divided into sets of layers based on the use case. This approach is not scalable in terms of handling multiple different application domains, as each will have their own structural constraints and data idiosyncrasies.

4 Next Steps

Our next goal, as part of our future work, is to allow users to provide a layering definition as part of their data definition using the meta-network. The meta-network is a visual representation that allows domain experts to reason in terms of the types of entities and relationships of interest. Yet, mapping a schema similar to Figure 1 to a multilayer network structure can be done in various ways. The

most simple approach may assume that layers are defined based on node type. In this case, every meta-node leads to the creation of a layer, and the drawn edges capture between-layer relationships. Yet layers may also be defined by edge type, e.g. types of acquaintance like friends, relatives, colleagues in a social network. In this case, between layer links may not necessarily exist, but comparison tasks will require "identity links" to be visualised between identical nodes in different layers. More elaborate layer definitions might also include multiple node and edge types in the same layer; lasso interactions would be a suitable vehicle for such layer definitions. Besides the previous topological considerations, layers may also be defined based on node or edge attributes [10, 15], whether they correspond to measured/observed data or derived from computations.

4.1 Visualization and layer definition

Set visualization is a highly researched field in information visualization [4], and clearly there is a potential relationship between the definition of sets and the definition of layers using the meta-network. Euler diagram based approaches provide one potential visualization technique. Due to the complexity of layer definitions, this may result in an Euler diagram that is not well-matched, in the sense that is does not exactly align with the set specification, e.g. due to nodes being duplicated in the set. Therefore an interesting approach is that of Simonetto and Auber [20] or Simonetto *et al.* [21] concerning visualization of overlapping sets, since it allows for subsets to be split into disjoint parts. However, as we are also interested in using edges to define layers, some of the more network focused visualization techniques related to Euler diagrams, such as the bubble sets of Collins *et al.* [8] or Kelp [11] diagrams may be a better starting point. Kelp diagrams have already been used by Cuenca *et al.* [9] as part of their overview visualization of multilayer network data sets. Bubble sets have been applied to many different visualization types by Collins *et al*. If they are included as part of the meta-network visualizations, they also need to visualise how the layers are characterised, i.e. the aspects, not just the entities involved.

Faceted network data visualization techniques are also very relevant to our problem of eliciting layer definitions. The pivot graph application of Wattenberg [26] allows the user to explore multivariate networks through a graph-like interface of meta-nodes. Facet Lens [16] allows the facets of a dataset to be explored and compared using the relationships. However, neither application considers a layer as a network entity in its own right, allowing the user to compare and explore them side by side. Additionally, the slicing of data in faceted visualization can be highly arbitrary, however, in multilayer networks the different layers generally model a physical reality of some kind.

5 Conclusion

Sources of multilayer network data can be both highly complex and vast in terms of scale. Providing an intuitive interface to query this data is challenging. End users demand self-service access to data, without the requirement of technical intermediaries unfamiliar with their domain problems. In the case of multilayer network visualization, such an interface offers the opportunity of defining layers intuitively and at an early stage in the process so that users are not overwhelmed with data. In this paper, we described the early stages of a work in progress which allows users to define their working set of data. We highlighted the avenues of future work, to encourage discussion with the community on the fundamental requirements of layer definition, it's limits, and the development of novel visualization techniques.

Acknowledgments

This work was funded by the BLIZAAR project on Multilayer Network Visualization (an international cooperation co-funded by the French ANR grant BLIZAAR ANR-15-CE23-0002-01 and the Luxembourgish FNR grant BLIZAAR INTER/ANR/14/9909176)

References

[1] Centrifuge systems., 2019. http://centrifugesystems.com/, Last accessed: 2019-08-14.

[2] STRING: proteinprotein association networks , 2019. https://string-db.org/, Last accessed: 2019-08-14.

[3] Touchgraph navigator 2, 2019. http://www.touchgraph.com/navigator, Last accessed: 2019-08-14.

[4] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. The state-of-the-art of set visualization. *Computer Graphics Forum*, 35(1):234–260, 2016. doi: 10.1111/cgf.12722

[5] D. Auber. Tulipa huge graph visualization framework. In *Graph drawing software*, pp. 105–126. Springer, 2004.

[6] G. Bianconi. Multilayer networks: Dangerous liaisons? *Nat Phys*, 10(10):712–714, oct 2014. doi: 10.1038/nphys3097

[7] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pp. 95–100. ACM, New York, NY, USA, 1988. doi: 10.1145/57167.57182

[8] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[9] E. Cuenca, A. Sallaberry, D. Ienco, and P. Poncelet. Visual querying of large multilayer graphs. In *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, SSDBM '18, pp. 32:1–32:4. ACM, New York, NY, USA, 2018. doi: 10.1145/3221269.3223027

[10] M. De Domenico, M. A. Porter, and A. Arenas. Muxviz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, 3(2):159–176, 2015. doi: 10.1093/comnet/cnu038

[11] 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. doi: 10.1111/j.1467-8659.2012.03080.x

[12] M. Gosak, R. Markovi, J. Dolenek, M. S. Rupnik, M. Marhl, A. Stoer, and M. Perc. Network science of biological systems at different scales: A review. *Physics of Life Reviews*, 24:118 – 135, 2018. doi: 10.1016/j.plrev.2017.11.003

[13] C. T. Have and L. J. Jensen. Are graph databases ready for bioinformatics? *Bioinformatics*, 29(24):3107, 2013.

[14] J. Heer and A. Perer. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. *Information Visualization*, 13(2):111–133, 2014. doi: 10.1177/1473871612462152

[15] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014. doi: 10.1093/comnet/cnu016

[16] B. Lee, G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan. Facetlens: exposing trends and relationships to support sensemaking within faceted datasets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1293–1302. ACM, Boston, MA, USA, 2009. doi: 10.1145/1518701.1518896

[17] Z. Liu, S. B. Navathe, and J. T. Stasko. Ploceus: Modeling, visualizing, and analyzing tabular data as networks. *Information Visualization*, 13(1):59–89, 2014. doi: 10.1177/1473871613488591

[18] F. McGee, M. During, and M. Ghoniem. Towards visual analytics of multilayer graphs for digital cultural heritage. In 1st *Workshop on Visualization for the Digital Humanities (Vis4DH)*. Baltimore, USA, 2016.

[19] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, and B. Pinaud. The state of the art in multilayer network visualization. *Computer Graphics Forum*. doi: 10.1111/cgf.13610

[20] P. Simonetto and D. Auber. Visualise undrawable euler diagrams. In *2008 12th International Conference Information Visualisation*, pp. 594–599, July 2008. doi: 10.1109/IV.2008.78

[21] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. In *Computer Graphics Forum*, vol. 28, pp. 967–974. Wiley Online Library, 2009.

[22] B. Škrlj, J. Kralj, and N. Lavrač. Py3plex: A library for scalable multilayer network analysis and visualization. In L. M. Aiello, C. Cherifi, H. Cherifi, R. Lambiotte, P. Lió, and L. M. Rocha, eds., *Complex Networks and Their Applications VII*, pp. 757–768. Springer International Publishing, Cham, 2019.

[23] A. Srinivasan, H. Park, A. Endert, and R. C. Basole. Graphiti: Interactive specification of attribute-based edges for network modeling and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):226–235, Jan 2018. doi: 10.1109/TVCG.2017.2744843

[24] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.

[25] C. Von Mering, L. J. Jensen, B. Snel, S. D. Hooper, M. Krupp, M. Foglierini, N. Jouffre, M. A. Huynen, and P. Bork. String: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(suppl 1):D433–D437, 2005.

[26] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 811–819. ACM, Montréal, Québec, Canada, 2006. doi: 10.1145/1124772.1124891