TU WIEN Informatics

Inspired by Biology

Towards Visualizing Complex Networks

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Henry Ehlers, MSc
Matrikelnummer 12131118

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dr. Renata G. Raidou
Zweitbetreuung: Dr. Hsiang-Yun Wu

Diese Dissertation haben begutachtet:

Daniel Archambault	Andreas Kerren

Wien, 11. August 2025

Henry Ehlers

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Inspired by Biology

Towards Visualizing Complex Networks

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Henry Ehlers, MSc
Registration Number 12131118

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dr. Renata G. Raidou
Second advisor: Dr. Hsiang-Yun Wu

The dissertation has been reviewed by:

Daniel Archambault	Andreas Kerren

Vienna, August 11, 2025

Henry Ehlers

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Henry Ehlers, MSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 11. August 2025

Henry Ehlers

Acknowledgements

I would start by thanking my two academic advisors, Assoc. Prof. Renata G. Raidou and Dr. Hsiang-Yun Wu, for their 4-year-long commitment to helping me grow as a researcher and seeing this PhD through to its end. From the many fruitful discussions to their invaluable writing advice, their input greatly shaped and improved my scientific work and thinking. Similarly, I would like to thank Assoc. Prof. Manuela Waldner for her scientific guidance and input when I felt unsure or lost. I would also like to thank Prof. Eduard Gröller for the opportunity to do my PhD at TU Wien's Visualization Group, but also for sharing his experience and expertise on multiple occasions.

While my superiors and mentors were instrumental in helping me shape and realize this project, so were my colleagues and friends at the department and university. Thus, I would like to thank, in no particular order, Daniel Pahr, Diana Marin, Johannes Eschner, Velitchko Filipov, and Sara di Bartolomeo. Beyond making coffee and lunch breaks more enjoyable, their opinions, input, and encouragement pushed me to not only see this project through, but do so to the best of my abilities.

Finally, and most importantly, I would like to thank my parents, Margaret Ward and Christian Ehlers, as well as my brother, William Ehlers, for being by my side and believing in me. The road to here has been a long and arduous one, and while I may have faltered along the way, never once did I doubt their unconditional love and support. I wouldn't be here without them, and I am eternally grateful.

Kurzfassung

Der Begriff "biologisches Netzwerk" umfasst eine große und facettenreiche Menge verschiedener Arten von Netzwerken, wie z.B. genregulatorische Netzwerke, Protein-Protein-Interaktionsnetzwerke, Stoffwechselwege, phylogenetische Bäume und genetische Variationsgraphen. Jeder dieser Netzwerktypen bringt einzigartige Herausforderungen für die Visualisierung und visuelle Analyse mit sich, die beispielsweise von der Größe und Komplexität der Graphdaten, den besonderen Analyseaufgaben der Fachleute oder den multivariaten Attributen der Kanten und Knoten des Netzes abhängen. Um besser zu verstehen, wo die theoretische und grundlegende Forschung gegenüber der anwendungs- und anwendungsorientierten Forschung aufholen muss, um wichtige Forschungslücken zu schließen, geben wir einen Überblick über den Stand der Technik bei der Visualisierung biologischer Netzwerke. Durch diese gründliche Durchsicht der Literatur, die in eine sorgfältig durchdachte Taxonomie (basiert auf der klassischen Visualisierungspipeline) eingebettet ist, sind wir in der Lage, mehrere noch offene Herausforderungen und Probleme zu identifizieren, wie z. B. das Fehlen ausgefeilter Graphenvergleichs- und Analysetechniken oder den übermäßigen Rückgriff auf schematische und geradlinige Knoten-Verbindungs-Diagramme.

Von diesen identifizierten Herausforderungen sind drei für uns von besonderem Interesse, und zwar aufgrund ihrer theoretischen Relevanz, ihrer konzeptionellen Herausforderung bzw. ihres Wertes für den Bereich selbst:

Herausforderung 1: Die Visualisierung von großen und dichten biologischen Netzwerken, die nicht in sogenannte "Hairballs" zerfallen; unlesbare Gewirre von Knoten und Kanten.

Herausforderung 2: Die Visualisierung der Arten von Ungewissheit, die oft mit biologischen Einheiten und Beziehungen verbunden sind.

Herausforderung 3: Die Visualisierung von Gruppenstrukturen, die in zahlreichen biologischen und biochemischen Systemen vorhanden sind.

Um diese drei Herausforderungen zu bewältigen, führen wir fünf Untersuchungen durch. Zunächst untersuchen wir das prinzipielle und algorithmische *splitting of vertices*, um Kantenkreuzungen iterativ aufzulösen und dadurch die Lesbarkeit von Graphen zu verbessern. Unser Ansatz wird in einer Online-Nutzerstudie an kleinen Graphen als

Machbarkeitsprüfung getestet. Zweitens untersuchen wir die Visualisierung sogenannter *ego-networks*, die es erlauben, nur die knotenrelative und knotenrelevante Topologie zu visualisieren, anstatt die Gesamtheit eines Netzwerks. Dazu untersuchen wir den aktuellen Stand der Visualisierung von Ego-Netzwerken und führen eine groß angelegte Online-Nutzerstudie durch, in der vier gängige Ansätze verglichen werden. Drittens untersuchen wir die Visualisierung von *Knotenattribut-Unsicherheit* unter Verwendung von animierter "Wackeligkeit", d.h. animierter Knotenbewegung, und vergleichen diesen neuartigen Ansatz mit drei alternativen Unsicherheitskodierungen auf dem neuesten Stand der Technik, die aus einer groß angelegten Umfrage in der Literatur abgeleitet wurden, in einer groß angelegten Online-Nutzerstudie. Viertens untersuchen wir den aktuellen Stand der Visualisierung von *compound graphs*, d.h. die Visualisierung von Graphen, deren Knoten nicht nur topologische Beziehungen, sondern auch Beziehungen auf Gruppenebene aufweisen. Anhand der Ergebnisse dieser Umfrage formulieren wir eine Reihe von Herausforderungen, die in der zukünftigen Arbeit angegangen werden sollen. Schließlich fügen wir die oben genannten vier Arbeiten zusammen und entwickeln ein *prototypisches Dashboard* für die Visualisierung von zusammengesetzten Graphen, das Elemente der Visualisierung biologischer Netzwerke, compound graphs und von Ego-Netzwerken kombiniert.

Es bleibt noch viel zu tun, um eine effektive Visualisierung komplexer biologischer Netzwerke zu gewährleisten, von der Entwicklung neuartiger Layout-Algorithmen, mit denen große Multi-omics-Netzwerke angegangen werden können, bis hin zur Entwicklung neuartiger interaktiver Systeme, mit denen solche Netzwerke visuell erforscht werden können. Damit solche zukünftigen Bemühungen effektiv sind, müssen sie auf einer soliden Literaturbasis aufbauen, die grundlegende Aspekte der Netzwerkvisualisierung untersucht. Hier argumentieren wir, dass unsere Untersuchungen von Vertex-Splitting, Visualisierung von Ego-Netzwerken, Visualisierung von Unsicherheiten und Visualisierung von zusammengesetzten Graphen für die Zukunft der Visualisierung komplexer biologischer Netzwerke von großer Bedeutung sind und sein werden. Zusammenfassend glauben wir, dafür eine gute Grundlage geschaffen zu haben.

Abstract

The term "biological network" comprises a large and multifaceted set of different types of networks, such as gene regulatory networks, protein-protein interaction networks, metabolic pathways, phylogenetic trees, and genetic variation graphs. Each of these network types brings with it unique visualization and visual analysis challenges, depending on, for example, the size and complexity of the graph data, the particular analysis tasks performed by domain experts, or the multivariate attributes attached to the network's edges and nodes. Inspired by the many challenges and difficulties faced by the field, we tackle (some of) them, not from the perspective of the domain, but abstract (network) visualization. First, in order to better understand where theoretical and fundamental research must catch up to application and application-driven research to fill important research gaps, we survey the state-of-the-art of biological network visualization. Through this thorough review of the literature, embedded within a carefully considered taxonomy based on the classical visualization pipeline, we are able to identify several outstanding challenges and issues, such as the lack of sophisticated graph comparison and analysis techniques or the over-reliance on schematic and straight-line node-link diagrams.

From these identified challenges, three are of particular interest to us, owing to their theoretical relevance, their conceptual challenge, or their value to the domain itself, respectively:

Challenge 1: The visualization of large and dense biological networks that do not devolve into so-called "hairballs", i.e., unreadable tangles of nodes and edges.

Challenge 2: The visualization of the kinds of uncertainty often attached to biological entities and relationships.

Challenge 3: The visualization of group structures common across numerous biological and biochemical (sub)systems.

To tackle these three challenges, we conduct five investigations. To tackle challenge 1, we first investigate the principled and algorithmic *splitting of vertices* to iteratively resolve edge crossings and thereby improve the readability of graphs. Our approach is tested on small graphs as a proof of concept in an online user study. As an alternative solution to challenge 1, we investigate the visualization of so-called *ego-networks*, which allow

for the visualization of only node-relative and node-relevant topology, instead of the entirety of a network. To do so, we study the current state of ego network visualization and conduct a large-scale online user study comparing four common approaches. Third, within the context of challenge 2, we investigate the visualization of *node attribute uncertainty* using animated "wiggliness", i.e., animated node motion, and compare this novel approach to three state-of-the-art alternative uncertainty encodings derived from a larger-scale survey of the literature, in a large-scale online user study. Fourth, in order to tackle challenge 3, we *survey the current state of compound graph visualization*, i.e., the visualization of graphs whose nodes, in addition to topological relationships, share group-level relationships. With the results of this survey in hand, we formulate a series of outstanding challenges to be addressed in future work. Finally, we combine the aforementioned four works together and develop a *prototypical dashboard* for the visualization of compound graphs, combining elements of biological network visualization, compound graph visualization, and ego network visualization.

Much work remains to be done to ensure the effective visualization of complex biological networks — from the development of novel layout algorithms with which to tackle large multi-omics networks to the engineering of novel interactive systems with which to visually explore such networks. For such future efforts to be effective, they must build upon a robust body of literature investigating fundamental aspects of network visualization. Here, we thus argue that, inspired by the many identified biological network visualization challenges, our consequent investigations of vertex splitting, ego network visualizations, uncertainty visualizations, and compound graph visualizations have taken the first steps towards the (more) effective visualization of the kinds of complex network data common to biology and biochemistry.

Contents

CHAPTER 1

Introduction

Molecular and systems biology studies the mechanistic and biochemical underpinnings of cellular (sub)systems through iterative modeling and (experimental) validation. These complex cellular systems comprise numerous different kinds of biological entities, such as genes, ribonucleic acid (RNA) fragments, proteins, or metabolites. Importantly, these biological entities are all connected through a dense web of biological interactions, such as

1. *genetic transcription* which, with the help of transcription proteins, turns (parts of) genes, encoded in deoxyribonucleic acid (DNA), into RNA (fragments) for the ultimate purpose of gene expression,
2. *RNA translation* which, with the help of complex biochemical machinery consisting of messenger RNA molecules, transfer RNA, and ribosome proteins, turns transcribed RNA sequences into amino acid polypeptide chains to ultimately form proteins,
3. *protein-protein interactions* which, through chemical forces and/or other proteins, bring (parts of) two or more proteins together in order to activate, change, or regulate their functions,
4. *metabolic reactions* which, often with the help of enzymes, i.e., functional proteins, chemically transform one set of metabolic intermediaries into another in order to provide the cellular system with chemical "building blocks" it requires to function, e.g., for DNA transcription, RNA translation, inter/intracellular signaling, or cellular replication,
5. *gene regulation* which, with the help of genetic regulator molecules, regulates the (co)expression of a gene during DNA transcription or RNA translation, thereby steering overall cellular function.

This web of biological interactions across biological entity types forms a so-called biological network, a complex network of biological and biochemical entities connected by a variety of interactions that together dictate cellular function.

Such networks are commonly modeled as *graphs* and are common across a wide variety of applications, from social network analysis, through (metropolitan) transportation network visualization, to biological network modeling. Such graphs mathematically describe the relationship, called *edges* or *links*, between entities, called *nodes* or *vertices*. These entities and their relationships can represent a diverse array of data. On the one hand, for example, in a social network, each node might represent a particular person, and each edge between them represents a (particular type of) relationship they share. On the other hand, in a transportation network, each node might represent a subway/tram/bus stop, and each edge a subway/tram/bus connection between them.

Because graphs are an abstract data structure, they do not necessarily have an intrinsic 2D or 3D spatial representation. In the case of a transportation network, for example, nodes can be placed on a (2D) map depending on where they are located geographically, and edges, i.e., subway/tram/bus connections, can be placed along their geographic route (Figure 1.1a). Depending on the user, task, or situation, however, such intrinsic representations of a network may not be desirable or readable. Consider how many metro maps do not necessarily represent space geographically accurately in the interest of better communicating connectivity (Figures 1.1c, 1.1b, and 1.1d). Moreover, many other graphs have no such intrinsic representation to begin with. Returning to the aforementioned social network, while it might be described mathematically as a graph, there is no singular way such a graph can be visually represented. Where (in 2D or 3D space) would one place a node representing a particular person? How would one represent said person visually? And how would one visually represent the relationships between two particular people? It is this very problem, i.e., the visualization of abstract graphs, that also underlies most biological network visualization efforts.

This process of visualizing graphs is called *graph drawing* and is commonly achieved with the help of a *graph drawing algorithm*: For some input graph, graph drawing algorithms commonly map every vertex to a point (in 2D space) and each (undirected) edge to a line or arc between respective vertex points. These so-called *node-link diagrams* are a familiar and powerful approach to visualizing networks. Beyond the simple straight-line node-link diagram (Figure 1.2a), however, these node-link diagrams come in many different variations, each of which with its own set of constraints on node and edge placement: some arrange nodes along the circumference of one or multiple circles (Figure 1.2b), others in layers, while others yet arrange nodes and edges along an ortholinear grid (Figure 1.2c). Here, it is important to stress that no representation is inherently superior to any other representation. Instead, each visualization approach brings with it its own set of advantages and disadvantages, depending on the user group, graph data properties, or the analysis tasks to be performed.

Alternatively, instead of visualizing networks as a type of node-link diagram, one could opt to visualize graphs tabularly, using an adjacency matrix (Figure 1.2d), biofabric (Figure 1.2e), or quilt. Consider the adjacency matrix, i.e., the simplest of the set of tabular representations. Every node is represented twice, once as a column and once as a row in a square matrix. If two vertices are connected by an edge, the corresponding matrix

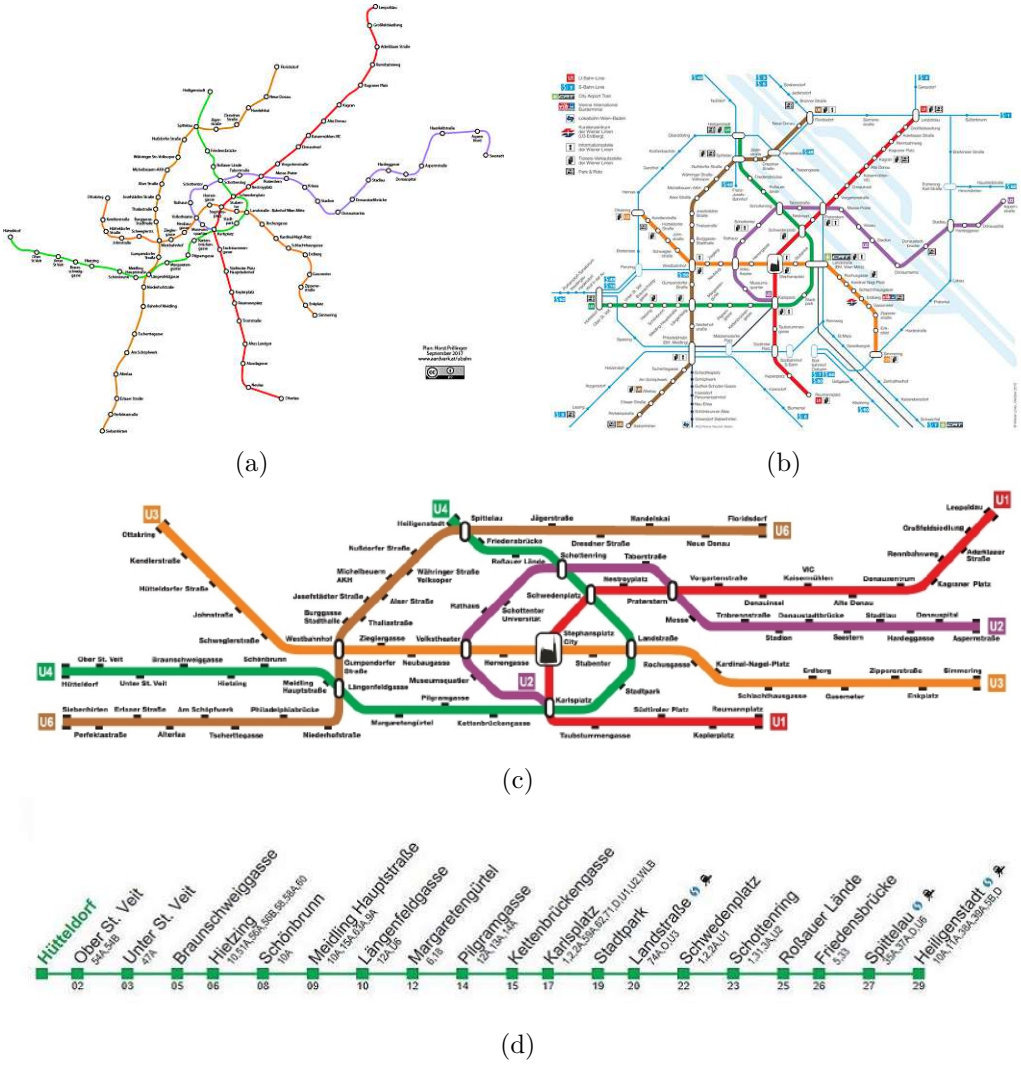
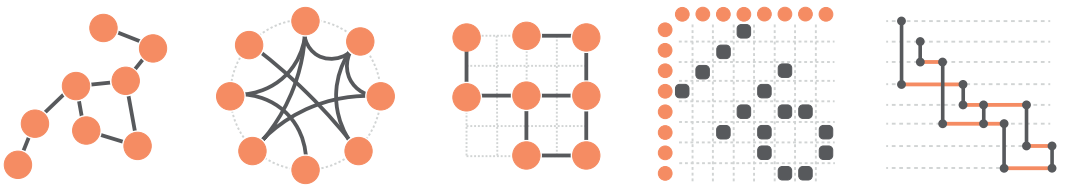


Figure 1.1: The Vienna subway system visualized in three different ways: a) a geographically accurate representation, b) an octolinear, square representation, c) an octolinear, rectangular representation, and d) a linear arrangement of a subgraph view of only one subway line. The complete subway system's graph consists of $|V| = 106$ nodes, i.e., stations, categorized into 5 groups, i.e., subway lines, and $|E| = 101$ edges, i.e., connections. Note that while maps (a) and (c) depict only the connectivity of the city's subways, map (b) additionally showcases the city's train connections in light blue. Moreover, (d) showcases the topology of only one group, i.e., one subway line, arranged linearly. Note also how, depending on the goal and context of the visualization, each of these representations is useful in its own way, depending on the context in which it is presented and the purpose of the visualization.



(a) (b) (c) (d) (e)

Figure 1.2: The same graph G, consisting of $|E| = 8$ edges and $|V| = 8$ vertices, visualized as a a) straight-line node-link diagram, b) radial node-link diagram, c) orthogonal node-link diagram, d) adjacency matrix, and e) biofabric representation.

cells are filled-in. Conversely, if two vertices are not connected, their corresponding cells are left empty (Figure 1.2d). While perhaps not as immediately intuitive as a node-link diagram, these tabular representations offer a number of conceptual advantages, such as better scalability with denser graphs, though at the cost of poorer scalability with increasing numbers of vertices.

Importantly, any domain that deals with the visualization (as well as visual exploration and analysis) of its graph data must wrestle with graph drawing in some form or another. Here, modern biological and biochemical networks are particularly interesting, as they present domain experts with large, dense, heterogeneous, and multifaceted graphs that are highly challenging to visualize. As previously discussed, examples of such networks include protein-protein interaction networks, gene regulatory networks, and metabolic pathway networks, but also gene variation networks, brain connectomics networks, or phylogenetic trees. Here, each of these unique types of networks brings with it unique challenges, depending on the research goals of the domain expert.

For example, within the context of gene expression analyses, researchers measure the abundance of genes (based on the amount of RNA transcripts or protein fragments present in a sample) across experimental conditions (Figure 1.3). These differences in relative gene abundance between conditions can then form the basis of statistical comparison, which can allow researchers to identify genes of potential substantive interest, as they may be crucially linked to observed phenotypic differences. However, domain experts often wish to study these genes not only based on their statistical differences but also within their biological context, i.e., within the context of gene regulatory or metabolic pathway knowledge graphs. This necessitates drawing such graphs and combining them with the statistical and experimental data collected. However, even without the inclusion of such additional data, these networks are highly complex, as they can feature different types of nodes, i.e., biological entities, different types of (directed and weighted) edges, i.e., biological and biochemical interactions, and (hierarchical) group structures, i.e., biological pathways encompassing non-unique subsets of biological entities. Once combined with experimental data, researchers are ultimately faced with the challenging task of visualizing a large, directed, multivariate, compound, and sometimes dynamic network. While we may not be per se interested in such application-oriented problem-solving, understanding

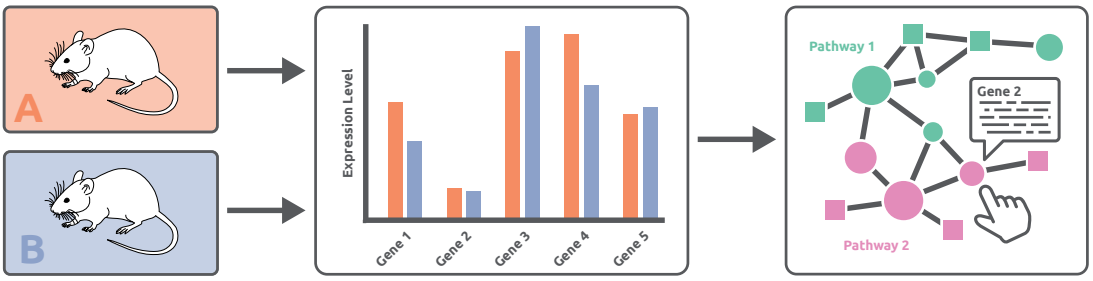


Figure 1.3: A simplified and schematic visualization of the experimental and (visual) analysis process of gene expression studies. For two conditions, A (control) and B (treatment), biological samples are collected; here, from mice. From these collected samples, gene expression levels are measured per gene and per condition. The differences between gene expression levels are shown here as a bar chart and form the basis of subsequent statistical analyses. Statistically significant differences in a gene's expression levels between conditions can indicate substantive relevance, i.e., the involvement of a gene in a crucial disease-promoting or repressing pathway. In order to study the biological context of such substantive genes, this experimental and statistical data is combined with gene regulatory or metabolic pathways knowledge graphs, shown on the right-hand side of the figure. Here, the various types of biological entities (visualized using different shapes), their different types of (here undirected) relationships, the system's group structure (visualized using different node colors), as well as each biological entity's statistical/experimental metadata (shown using different node sizes) must be visually communicated for a domain expert to effectively study such biological (sub)systems.

the current requirements of a domain as broad and challenging as biological network visualization can allow us to identify key areas in which fundamental research must catch up to meet application-oriented demands. In this dissertation, we, inspired by these challenges, thus aim to tackle several fundamental network visualization problems relevant to the current state-of-the-art of biological network visualization in order to take steps towards the (more) effective visualization of the domain's challenging network data.

Research Questions Given the breadth of types of networks that make up biological networks and the many (sub)domains involved in their study, it is unsurprising that we are able to identify several outstanding challenges (Chapter 2). In this dissertation, we are particularly interested in the, in our estimation, three most salient outstanding issues of biological network visualization: improving the readability of conventionally laid-out straight-line node-link diagrams, incorporating uncertainty in biological network visualization efforts, and the visualization of group structure in biological compound graphs. Thus, we define three overarching research questions.

Research Question 1 **How to (more effectively) visualize complex networks common in biological network endeavors?** The readability of a graph drawing can be quantified using so-called

graph aesthetic/readability criteria [Pur02], such as the number of edge crossings, the orthogonality of the drawing, or vertices' minimum incident edge angle ratios. As biological networks increase in complexity, i.e., edge density and number of nodes, the heuristic (stress and force-based) layout algorithms commonly employed in the field produce less and less (graph aesthetically) readable visualizations. We identify three possible conceptual approaches to improving the graph aesthetics of network visualizations. First, one might develop novel graph drawing algorithms that attempt to optimize certain readability criteria directly instead of, as is the case with heuristic approaches, indirectly. Second, one might develop novel methods with which to improve the readability of a drawing *after* it has been drawn by some (arbitrary) user-selected algorithm. Third and finally, one might side-step the problem of visualizing the graph in its entirety by restricting the user's view to a subpart of the network using appropriate and novel *Focus+Context* techniques. Here, given the focus of this dissertation on network visualization, we are particularly interested in investigating the latter two approaches.

Research Question 2 **Which uncertainty is prevalent in biological networks and how should said uncertainty be effectively visualized?** Biological networks feature multivariate attributes attached to both their nodes and edges, such as expression levels/differences attached to genes or relative abundances attached to proteins. These multivariate attributes, however, can be uncertain. Within the context of previously discussed gene expression analyses, for example, individual gene expression differences attached to nodes are actually mean/median values and thus have a certain range of values around them. Typically, these uncertainties are ignored in the visualization itself. In this dissertation, we aim to tackle the problem of actually visually communicating this uncertainty effectively in order to enhance domain expert decision-making.

Research Question 3 **How to effectively visualize compound graph group structure common to biological network visualization endeavors?** Biological networks often feature some form of (hierarchical) group structure, which organizes their edges and/or nodes into (non-disjoint) sets. Crucially, domain experts are simultaneously interested in both the network's biological entities and their relationships as well as the graph's group structures. This complicates the visualization of such

networks, as care must be taken not to visualize one at the cost of the other. In this dissertation, we study how to effectively visualize such compound graphs in order to facilitate better visual analyses and explorations of biological data.

Dissertation Structure To tackle these three research questions effectively, we have compiled six chapters (Figure 1.4), based on six published first/lead-author papers, each of which aims to either directly tackle or contextualize one of these challenges. This dissertation is thus structured as follows.

Chapter 2, based on the published paper "*An introduction to and survey of biological network visualization*" [EBK+25], aims to provide an overview of and introduction to the current state-of-the-art of biological network visualization in order to ultimately identify interesting gaps for future research to fill. These are further investigated in the subsequent chapters.

Chapter 3, based on the published paper "*Improving readability of static, straight-line graph drawings: A first look at edge crossing resolution through iterative vertex splitting*" [EVRW23], aims to tackle *Research Question 1*, i.e., how to improve the visualization of complex networks and their visualizations, by improving the graph aesthetic readability of some given graph drawing. More specifically, we develop a novel approach to iteratively resolving edge crossings (a notoriously important readability criterion) through vertex splitting. Given biologists' and biochemists' conceptual familiarity with (manual) vertex splitting, e.g., in the context of *KEGG* pathways, we believe that, if developed further, our algorithmic approach to vertex splitting might find ready acceptance among said experts. Chapter 4, based on the published paper "*Me! Me! Me! Me! A study and comparison of ego network representations*" [EPF+24], also tackles *Research Question 1*, but does so by studying a (for biological network visualization) novel *Focus+Context* approach in the form of ego networks. In our estimation, such ego network could be of particular interest to biologists and biochemists, as their (visual) analyses of networks often commence with some *a priori* selected entity (vertex) of interest and its immediate neighborhood(s).

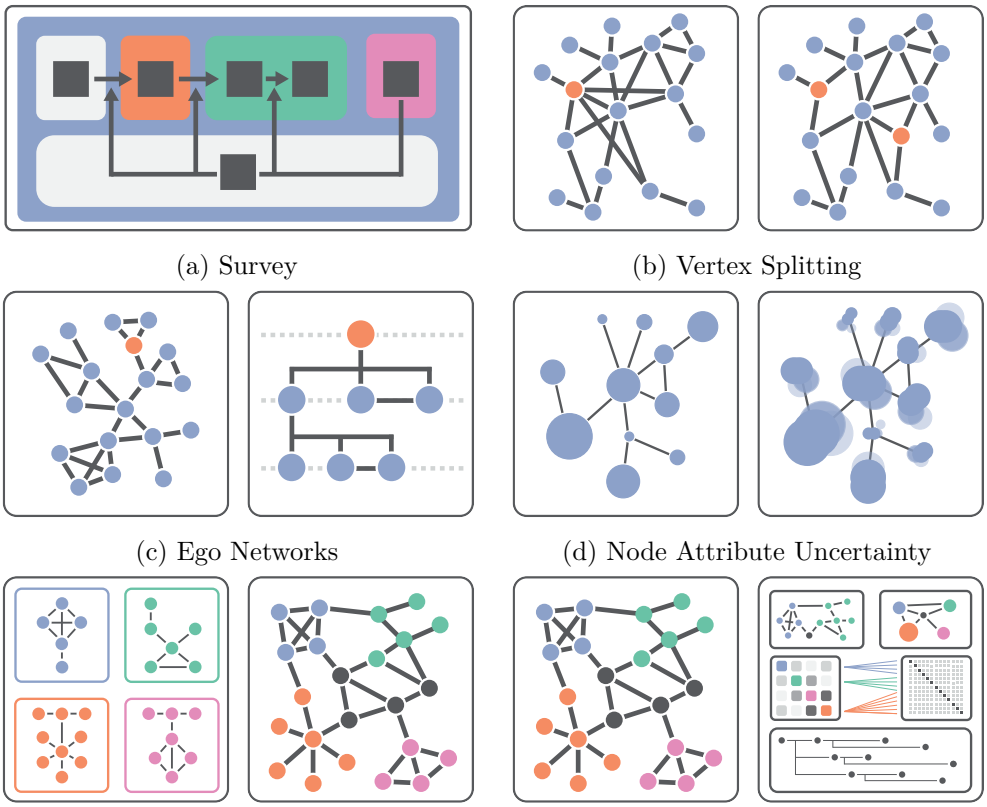
Chapter 5, based on the paper "*Wiggle! Wiggle! Wiggle! Visualizing Uncertainty in Node Attributes in Straight-Line Node-Link Diagrams Using Animated Wiggliness*" [EPdB+25a], aims to tackle *Research Question 2*, i.e., studying and developing novel approaches to uncertainty visualization relevant to biological networks. Specifically, we are interested in developing a novel approach to node attribute uncertainty, as these are, in our estimation, common in and relevant to biological network visualization endeavors.

Finally, chapters 6 and 7 tackle *Research Question 3*, i.e., the visualization of compound graphs. Chapter 6, based on published position paper "*Visualizing Group Structure in Compound Graphs: The Current State, Lessons Learned, and Outstanding Opportunities*" [EMWR24], characterizes the current state-of-the-art of compound graph visualization using a novel taxonomy for both the visual encoding of group structure and the visual

relationship between group structure and graph topology. We deemed this chapter of particular interest to biologists and biochemists, as their network data often features (hierarchical) group structures, be it SNPs that fall within genes, or genes that fall within (metabolic or gene-regulatory) pathways, or proteins that fall within larger protein complexes. Chapter 7, based on the published short paper "*Penta: Towards Visualizing Compound Graphs as Set-Typed Data*" [EKR25], proposes a novel, prototypical dashboard for the visualization of compound graphs, which, inspired by set visualization, decomposes the visualization problem into five separate, but linked, views. While still a prototype, and not (yet) tailored towards the visualization of biological/biochemical data, we believe that the conceptual approaches employed by *Penta* could be useful to the kinds of aforementioned biological compound graphs. The dissertation ultimately concludes with chapter 8 summarizing the work presented, as well as a discussion of future work.

In the remainder of this introduction, we briefly summarize each of Chapters 2 until 7. Finally, we provide a brief enumeration and summary of all additional papers that this dissertation's author was involved in during his PhD, but do not consist of core contributions of this dissertation.

Chapter 2: A Roadmap First, we must understand and characterize the current state-of-the-art of biological network visualization across its many sub-domains in order to identify key outstanding challenges [EBK+25]. Additionally, we provide the reader with an overview of the core techniques underlying (biological) network visualization. Thus, we position this first chapter as not only a survey of but also an introduction to modern biological network visualization. This introduction covers various aspects of network visualization, such as the data structures utilized, the techniques used for network analysis, various common graph drawing approaches, the abstract graph tasks underlying visual analysis endeavors, core interaction techniques, and examples of interactive biological network visualization tools across various domains (Figure 1.4a). Based on the insights of this literature survey, couched in a taxonomic classification following the classical visualization pipeline, we are subsequently able to identify and discuss six outstanding biological network visualization challenges, such as the over-reliance on straight-line node-link diagrams, despite their poor visual scalability, or the lack of acknowledgment of uncertainty in the network's layout as well as node and edge attributes.



(a) Survey (b) Vertex Splitting

(c) Ego Networks (d) Node Attribute Uncertainty

(e) Compound Graph Visualization (f) The *Penta* Dashboard

Figure 1.4: A visual summary of the contributions of this dissertation's chapters, consisting of i) a state-of-the-art report of and introduction to the various facets that make up biological network visualization (Chapter 2), ii) a first look at principled vertex splitting for iterative edge crossings resolution (Chapter 3), iii) a survey of the current state of ego network visualization as well as a comparison of commonly utilized approaches in a large-scale user study (Chapter 4), iv) a state-of-the-art report of compound graph visualization (Chapter 6), v) a survey of the current state of uncertainty visualization approaches as well as a comparison of three carefully chosen approaches to a novel, proposed approach in the form of animated "wiggliness" (Chapter 5), and finally vi) the development of the prototypical dashboard *Penta* for the visualization of compound graphs (Chapter 7).

Chapter 3: Untangling the Hairball One key challenge we note in our survey of the current state of biological network visualization is the frequent use of heuristically laid-out straight-line node-link diagrams, most likely owing to their availability across libraries and packages, as well as their comparative computational scalability. However, these heuristic layout algorithms only indirectly optimize crucial graph aesthetic metrics, which each quantify different aspects of a graph's readability. Thus, they do not necessarily scale well visually with larger and denser graphs. While all aesthetic metrics affect graph readability, one metric stands out as particularly important: the number of edge crossings. Resolving these edge crossings is, however, not trivial, and two possible approaches present themselves in the literature. First, one can minimize the number of edge crossings *during* the algorithmic drawing of the network — a notorious, *NP*-Hard problem. Second, one can resolve edge-crossings *after* some drawing has been produced, regardless of the layout algorithm used, through, for example, the removal of graph aesthetically problematic nodes and edges. However, domain experts are seldom interested in removing biological entities or relationships under study, thus making edge and node deletion approaches theoretically interesting but practically undesirable. Thus, we propose vertex splitting [EVRW23]: Instead of removing a "problematic" vertex and all its incident edges from a graph drawing, we propose splitting it into two separate copies. The original set of incident edges can then be redistributed across these two copies of the original vertex, thereby resolving (a subset of) the edge crossings in which the original vertex was involved (Figure 1.4b), thereby making the drawing more readable.

Chapter 4: Think Globally, Act Locally An alternative approach to tackling — or more precisely, side-stepping — the aforementioned "hairball" problem is to simply not visualize the entirety of the network at once, but only parts of it; specifically, parts relevant to some current visual analysis task. While many different *Focus+Context* approaches present themselves for this purpose, we are particularly interested in a graph visualization technique common in the social sciences: *ego networks* [EPF+24]. Ego networks do not visualize a graph in its totality but only visualize connectivity relevant to a particular (user-selectable) focal node (the *ego*) (Figure 1.4c). However, despite its conceptual power and prevalence in (dynamic) social science network visualization, the technique has not been embraced in other domains, such as biological network visualization. Moreover, we note that little, if any, systematic work on characterizing ego networks, their visualization, and their effectiveness has been done at all. Thus, in this chapter, we first conduct a thorough review of the literature and categorize the collected papers based on both their ego network representations as well as their application domain. From this survey, we identify four representations commonly utilized across domains and applicable to biological networks. We subsequently compare these four representations in a large-scale crowd-sourced user study across six ego-network-specific analysis tasks in a mixed-methods analysis setup.

Chapter 5: Are You Sure? In biology, multivariate attributes attached to a network's nodes and edges can come in a variety of forms and shapes, from qualitative pathway-level group structures to quantitative node-level gene expression differences. Commonly, these multivariate attributes have been treated as certain when, in actuality, these attributes may in fact be *uncertain* as well. For example, returning to the aforementioned example of gene expression analysis (Figure 1.3), each node's expression difference is actually the difference between two mean values. This means each node's difference value has some deviation around it to be considered. Here, including these uncertainties should allow domain experts to perform more holistic analyses and allow for better decision-making, e.g., selecting genes (nodes) for experimental follow-up studies. We are thus interested in investigating how to effectively visualize node attribute uncertainties in networks, as we see the value thereof for biological network visualization efforts. Many different approaches, especially in the context of (3D volumetric) medical data, have been investigated. However, in the context of network visualization, very little, if any, systematic work has been done on the topic. Moreover, we note that one particular channel remains both controversial and under-explored across domains despite its many conceptual advantages: animation. Thus, inspired by relevant medical visualization applications, we set out to investigate animation, specifically in the form of the novel channel of "node wiggliness" (Figure 1.4d), and compare it to meaningfully selected alternative visual uncertainty encodings [EPdB+25a].

Chapter 6: Beyond the Individual One dimension beyond graph topology common to biological networks is group structure: nodes and/or edges that belong to some, potentially hierarchical, set of biologically-relevant groups. Commonly, these group structures are obtained from online knowledge graphs, representing the field's current — and often experimentally acquired and confirmed — understanding of an organism's biochemical inner workings. Consider, for example, the previously discussed gene expression analyses (Figure 1.3). Here, focus lies not only on individual gene expression differences between experimental conditions, but also on macroscopic differences on a pathway level. However, in order to study the biological context of these gene-level expression differences, these knowledge graphs' topologies and group structure must be visualized alongside experimentally collected data. As interest lies simultaneously in both the graph's topology and group structure, care must be taken to visualize each without negatively affecting the other (Figure 1.4e). Here, in order to study and understand current approaches to group structure visualization, we conduct a survey of the state-of-the-art by both combining existing reference sets from previously published reports as well as collecting novel literature [EMWR24]. The collected set of papers is then classified within a novel taxonomy, with which we are able to identify a set of outstanding challenges and gaps that may be addressed in future research efforts.

Chapter 7: Putting It All Together Throughout this dissertation, we aim to investigate fundamental network visualization challenges relevant to biological network visualization. However, we also aim to lay the foundation for an application-oriented solution to the group structure visualization challenge in compound graphs. We argue that the visualization of group structure in graphs can be viewed as comparable to set visualization, where a graph's groups form the sets, nodes form the elements, and nodes that map to multiple sets form intersections. Additionally, we argue that, in order to effectively tackle the various tasks of the visual analysis of compound graphs, each task requires its own visual representation in a multi-view dashboard. Thus, we present the prototypical *Penta* dashboard (Figure 1.4f) [EKR25], designed to decompose the compound graph visualization problem into multiple linked sub-parts. We demonstrate the potential value of *Penta* in three case studies, one of them stemming from the biological domain.

Other Contributions Finally, we also aim to contribute to (applied) network visualization efforts both within and without the field of biological network visualization. To that end, this author was involved in several additional research endeavors. These contributions have been published as follows:

1. *Tailored Mass Spectral Data Exploration Using the SpecXplore Interactive Dashboard*. In this paper, published in the journal *Analytical Chemistry*, Mildau et al. [MEO+24] develop a dashboard for the visual exploration of mass spectral data in the context of untargeted metabolomic networks. The value and utility of this dashboard, *SpecXplore*, is demonstrated using two real-world LC-MS/MS untargeted metabolomics datasets. As the second author, this dissertation's author's contributions to this paper include conceptualization, implementation, and editing. More specifically, this author was heavily involved in the prototyping of the dashboard, both during the conceptualization and initial software implementation stages. Finally, the author had a hand in editing and preparing the final version of the submitted and finally accepted paper, ensuring its correctness and completeness from a (network) visualization perspective.
2. *ConAn: Measuring and Evaluating User Confidence in Visual Data Analysis Under Uncertainty*. In this paper, published in the journal *Computer Graphics Forum*, Musleh et al. [MCER25] explore the role of user confidence in guided visual analysis under the influence of uncertainty. Several (network) metrics with which to gauge user confidence are proposed and evaluated using a lab study studying the impact of guidance on (self-reported) user confidence. As the third author, this dissertation's author's contributions to this paper are conceptualization and editing. More specifically, this author was involved in the conceptualization of the topological graph metrics developed to assess user confidence.
3. *Effective data visualization strategies in untargeted metabolomics.* In this review paper and state-of-the-art report, published in the journal *Natural Product Reports*, Mildau et al. [MEM+25] describe and explore the current untargeted metabolomics

workflow from the perspective of information and network visualization in order to identify gaps for future research. As the second first author, this author's contributions to this paper include conceptualization, literature review, writing, and editing. More specifically, the author was heavily involved in the drafting of the initial idea as well as the researching and writing of sections 2 and 3, i.e., the sections on information and network visualization, specifically. This involved a thorough review of the literature, the writing of the sections themselves, the creation of illustrative figures with which to explain certain (network) visualization concepts to metabolomics domain experts, the unification of the entire paper in terms of language used, as well as the proofreading and editing of the entire document.

4. *HoloGraphs: An Interactive Physicalization for Dynamic Graphs.* In this short paper presented at the *17th International Conference on Information Visualization Theory and Applications*, Pahr et al. [PEF25] present a novel physicalization toolkit with which to represent, explore, and interact with dynamic networks. The utility of this novel approach is demonstrated in a simple case study featuring a dynamic network derived from the *Harry Potter* book series. As the second author, this author's contributions include conceptualization, writing, figure creation, and editing. More specifically, this author was involved in the initial brainstorming of the idea, the conceptualization of the prototype, the creation of the core explanatory figure with which to visually communicate the creation of a *HoloGraph*, the creation of included photos and videos, the writing of the graph theoretic section, as well as the proofreading and editing of the entire submitted document.
5. *NodKant: Exploring Constructive Network Physicalization.* In this paper, presented as best paper at the *27th EuroVis* conference and published in the journal *Computer Graphics Forum* in 2025, Pahr et al. [PDBE+25] propose a physicalization toolkit for the personalized physical construction of graphs. In a mixed-methods user study, the authors analyze the impact of self-construction on user performance and comprehension. As the third author, this dissertation's author's contributions include conceptualization, figure creation, writing, and editing. More specifically, this author was heavily involved during the initial development of the paper's core idea, the development of the evaluation strategy, the creation of several (results) figures, the writing of several sections of the paper, and the proofreading and editing of the submitted document.
6. *BattleGraphs: Forge, Fortify, and Fight in the Network Arena.* In this workshop paper, presented at the *1st EuroVis Workshop on Visualization Play, Games, and Activities* in 2025, Ehlers et al. [EPdB+25b] develop an educational, competitive board game based on the previously discussed *NodKant* toolkit. As lead author, this author's contributions include conceptualization, writing, and editing. More specifically, this author was heavily involved in the initial development of the paper's core idea(s), the delegation of tasks to co-authors, the downstream creation of props, the writing of large parts of the paper, as well as the proofreading and editing of the entire document.

CHAPTER 2

A Roadmap: The Current State of Biological Network Visualization

Biological networks form a large and multifaceted class of networks, from protein-protein interaction networks, through genomic variation networks, to phylogenetic trees. Depending on the particular application domain, these networks bring with them unique visualization challenges, owing to i) their graphs' large sizes and densities, ii) the multivariate attributes attached to nodes and/or edges, or iii) many types of nodes, edges, and layers featured within them. While we are more interested in fundamental network visualization problems than domain-specific application research, understanding the current state of a domain, such as biological network visualization, can help us pinpoint where fundamental research must catch up with application-driven demands. Indeed, it was many of the problems identified here that inspired the work discussed in subsequent chapters.

In this chapter, we thus survey the current state of biological network visualization. This chapter also serves as a general introduction to the basics of (biological) network visualization. The contents of this chapter are based on the state-of-the-art report entitled *"An Introduction to and Survey of Biological Network Visualization"*, published in the journal *Computer & Graphics* [EBK+25], with myself as first author, in collaboration with Nicolas Brich, Jiacheng Yu, Michael Krone, Martin Nöllenburg, Hiroaki Natsukawa, Xiaoru Yuan, and Hsiang-Yun Wu.

2.1 Overview

Biological networks describe complex relationships in biological systems, which represent biological entities as vertices and their underlying connectivity as edges. Ideally, for a complete analysis of such systems, domain experts need to visually integrate multiple sources of heterogeneous data, and visually, as well as numerically, probe said data in order to explore or validate (mechanistic) hypotheses. Such visual analyses require the coming together of biological domain experts, bioinformaticians, as well as network scientists to create useful visualization tools. Owing to the underlying graph data becoming ever larger and more complex, the visual representation of such biological networks has become challenging in its own right. This introduction and survey aim to describe the current state of biological network visualization in order to identify scientific gaps for visualization experts, network scientists, bioinformaticians, and domain experts, such as biologists or biochemists, alike. Specifically, we revisit the classic visualization pipeline, upon which we base this chapter's taxonomy and structure, which in turn forms the basis of our literature classification. This pipeline describes the process of visualizing data, starting with the raw data itself, through the construction of data tables, to the actual creation of visual structures and views, as a function of task-driven user interaction. Literature was systematically surveyed using Application Programming Interface (API)-driven querying, where possible, and the collected papers were manually read and categorized based on the identified sub-components of this visualization pipeline's individual steps. From this survey, we highlight a number of exemplary visualization tools from multiple biological sub-domains in order to explore how they adapt these discussed techniques and why. Additionally, this taxonomic classification of the collected set of papers allows us to identify existing gaps in biological network visualization practices. We finally conclude this report with a list of open challenges and potential research directions. Examples of such gaps include i) the overabundance of visualization tools using schematic or straight-line node-link diagrams, despite the availability of powerful alternatives, or ii) the lack of visualization tools that also integrate more advanced network analysis techniques beyond basic graph descriptive statistics.

2.2 Introduction

Molecular and systems biology attempts to understand the complex mechanistic underpinnings of biological systems by interactively modeling and predicting a system of interest, and finally verifying said model experimentally [CHI10]. These systems are complex, large, and composed of numerous types of interconnected biochemical entities, such as genes, different types of RNA, proteins, or metabolic intermediaries. Commonly, such systems are represented as a graph, in which each biomolecule forms a vertex, and edges represent some form of interaction between two or more such entities [KKPEP20]. These interactions can represent many different types of relationships depending on the network in question, such as an evolutionary link between two genes in a phylogenetic tree, functional (regulatory) relationships between genes, or a protein-metabolite-reaction that forms a new metabolic intermediate [PWS08].

These networks are often best understood as an abstract representation of the community's current state of knowledge about a system of interest [KDSM20]. More specifically, these knowledge graphs represent the accumulated known and putative relationships between certain entities. As such, these graphs are frequently produced by combining existing knowledge with newly acquired experimental data to interactively explore, form, and verify scientific hypotheses. Here, the specific application areas of such visual analyses of biological networks are varied, ranging from i) studying the genetic basis of phenotypic variation based on differential gene (co-)expression, through ii) biomarker discovery for early disease risk classification using metabolomics, to iii) predicting a protein's function based on its interactions with other proteins [AKK+10].

Owing to both technical and numerical limitations, early research efforts focused on individual entities, or small subsets of linked entities, to understand specific relationships between them, such as the role of one particular protein in signal transduction [Nis84] (Section 2.9.1) or the development of (by today's standard) small gene regulatory network maps of 40 genes [DRO+02] (Section 2.9.1). Modern research, on the other hand, aided by the development of high-throughput data acquisition techniques and the availability of increasingly large libraries of previously collected data, enables a holistic understanding of these biochemical networks [GOB+10]. The scale of these networks has increased in terms of the number of entities under study, and also in terms of modeling multiple interconnected biochemical networks simultaneously (Section 2.9.1), sometimes described as a "network of networks", i.e., the interlacing of multiple inhomogeneous networks into a single larger network [SD14]. While beyond the scope of this survey, others go even further than just integrating multiple types of biochemical systems, by seeking to additionally model non-molecular networks. These so-called "network medicine" systems are multilevel "interactome" networks that combine multiple omics networks with, for example, phenotypic similarity or social networks, to better understand and predict disease risk [BGL11].

Earlier biochemical networks, owing to their simpler nature and smaller size, lent themselves more readily to "automated" quantitative analysis. However, owing to the dimen-

sionality and heterogeneity of modern biological network data, such purely quantitative analyses may no longer be possible, or desirable, without an (exploratory) human-in-the-loop visualization. This could involve inspecting particular subparts of the network under study, gaining a big-picture understanding of the network's topology, guiding the quantitative analysis, or even performing an exploratory analysis in lieu of a traditional (statistical) one [Fig15].

Yet, for such human interaction to be effective, the visual representation of the data must be carefully considered. Here, many different visual representations of networks exist and are discussed in literature [KKPEP20], each of which may be appropriate depending on the type(s) of network(s) under study, the network's dimensionality, as well as the study's analytical goals. Consider, for example, on the one hand, a simple (force-directed) planar graph layout, i.e., circles representing nodes connected by straight line segments representing edges between.

Such straight-line node-link diagrammatic representations are intuitive to read and straightforward to implement [Kob14], but scale comparatively poorly (in terms of readability) with increasing numbers of entities, relationships, and layers [KPS14]; only to produce what is often referred to as a "hairball" [KKPEP20, SHW+20]. On the other hand, a more abstract approach, such as Yoghourdjian et al.'s [YDK+18] *Graph Thumbnails*, "icon-like" summary visualizations of a network's higher-level topological structure, may allow for a concise and readable high-level representation of even large networks, but do not allow for any straightforward inspection of individual entities or relationships within such networks. The key challenge lies in finding a trade-off between meaningfully representing the data — or at least the key quantities of interest — while ensuring the data and its context are presented clearly enough to avoid overwhelming the user [GOB+10].

However, sensible representations of high-dimensional data are only one aspect of making such networks understandable. The second, and often underappreciated [YKSJ07] component to assist in making such visualizations readable is the use of effective interaction techniques [Fig15]. This "dialogue" between the user and system is necessary to enable both the effective confirmation of expectations as well as the discovery of novel insight from the data [Kei02]. This back-and-forth between the system and the user is not just a matter of providing effective modes of interaction. While a complete automated quantitative analysis may not be desirable, using such quantitative analysis techniques is invaluable in assisting users to better explore the network under study, as well as refine and validate their experimental hypotheses. In summary, the interactive visualization of biological networks is important, but also incredibly challenging as it sits at the intersection of visualization, graph theory, network analysis, bioinformatics, and biology itself.

Several surveys have been published over the years, from discussions of the data themselves, through the analysis of graph data, to the effective visualization of networks. Examples of published surveys, relating both directly and indirectly to the analysis and visualization of biological networks specifically, include i) a compilation of a list of (now dated) biological

visualization tools and their functionality [PWS08], ii) an overview of the requirements of, and layouts useful to, biological network visualization [BBS14], iii) a state-of-the-art report on multivariate graph visualization and analysis tasks associated therewith [NMSL19], iv) surveys on the visualization of group structure in graphs [VHTW13, EMWR24] v) the development of a general taxonomy to describe the tasks performed in biological pathway visualization [MMF17], vi) a discussion of taxonomies to categorize methods of interaction [YKSJ07], vii) a breakdown of graph theory to assist domain experts in understanding graph data structures and algorithms [KKPEP20, PSM+11], viii) the identification of a number of popular network visualization tools to compare their applicability to high-dimensional data [MFA18], and most recently ix) Filipov et al.'s [FAM23] compilation and unification of graph task taxonomies. We note, however, that no survey or report aims to unify all these individual domains in order to provide a more holistic view of the challenges of biological network visualization.

In this introduction and survey, we build upon this extensive body of literature and extend the preliminary work of Wu et al. [WNV19]. Specifically, while individual review papers have been published that tackle, for example, the topics of graph theory, graph analysis, or (interactive) graph visualization individually, there exists no review that provides an introductory overview to all these topics with a focus on visualization applications in the biological/biochemical/biomedical domains. In this survey, aimed at bioinformaticians, network scientists, and visualization experts alike, we concretely aim to

1. Provide an introduction to the many facets of (biological) network visualization; from graph models (Section 2.4), through the topics common in network analysis (Section 2.5), to various visualization approaches (Section 2.6)

2. Examine the current state and role of visualization in the visualization pipeline of biological networks. Specifically, we aim to highlight gaps between visual tools employed in domain-specific applications and the state-of-the-art developments in visualization literature. We note seven gaps in the literature, namely
   a) the overabundance of schematic (Section 2.6.4) and straight-line node-link diagrams (Section 2.6.1, despite the existence of several powerful alternatives (Section 2.6),
   b) the surprising lack of uncertainty visualization in biological networks, despite its presence in both experimental data, and reference networks and knowledge graphs,
   c) the lack of network analysis techniques (Section 2.5) that go beyond basic descriptive measures with which to rank nodes based on topological importance,
   d) the lack of meaningful network visualization tools for the comparison of biological (sub)networks,
   e) the overabundance of visualization tools for exploratory analysis and hypothesis generation, but not hypothesis verification,
   f) the lack of provenance and user trust tracking, and

g) the lacking availability of dynamic biological network visualization and analysis tools, despite its growing importance across many application domains.

3. Provide an up-to-date overview of the current visualization tools available to different biological domains (Section 2.9). To our knowledge, the last such compilations are now over ten years old [PWS08]. Hence, an up-to-date list should be useful to both the domain and visualization communities.

2.3 Taxonomy

The analysis methods and visualization of biological networks are non-trivial, as i) each (sub-)domain brings with it unique goals, tasks, and challenges [OGG+10], ii) the process of knowledge generation is seldom linear owing to users' simultaneous interactive generation and verification of hypotheses [DP20], and iii) individual technical steps, from the selected graph layout algorithms to the analysis techniques employed, can influence the effectiveness of the (interactive) visualization [SSS+14]. As such, systematically collecting and categorizing literature on such a broad topic is non-trivial as well. Various models have been put forward over the years to capture this complex process of (visually) generating knowledge from data interactively, especially within the context of visual analytics [LGH+17, Mor12, WZM+16]. We instead adapt the comparatively simpler *Information Visualization Pipeline* presented by Card et al. [Car99], as the basis of our own taxonomy (Figure 2.1), as i) it relates to visualization specifically, and ii) it intuitively forms a suitable basis for a linear report such as this, as well as its taxonomy.

This pipeline linearly arranges visualization into four stages, namely *raw data*, which is processed to form *data tables*, which is processed to form *visual structures* (i.e., a combination of spatial substrates, marks, and graphical properties), which produces actual visual *views*, connected by *Task*-driven *Interaction* techniques (Figure 2.1) [DP20]. In essence, the pipeline dichotomizes the visualization process into data and visual components, with an explicit emphasis on the importance of (task-driven) interaction, which aims to complete tasks based on their priority at every step along the way [WZM+16]. We adopt the pipeline from Card et al. [Car99] as the basis of our own taxonomy, which consists of six classes:

1. *Graph Models* which describes the structure of the raw data fed into the pipeline,
2. *Graph Analysis* which summarizes or provides additional insight into the raw graph data to facilitate more effective visualization or interaction,
3. *Network Visualization* which transforms the provided data into a structure more amenable to visualization, subsequently presented in one or more views,
4. *Analysis Tasks* which determine the tasks necessary for the analysis at hand, as well as what modes of *Interaction* are provided to achieve those analytical tasks,
5. *Applications* which motivate and inform every aspect of the pipeline.

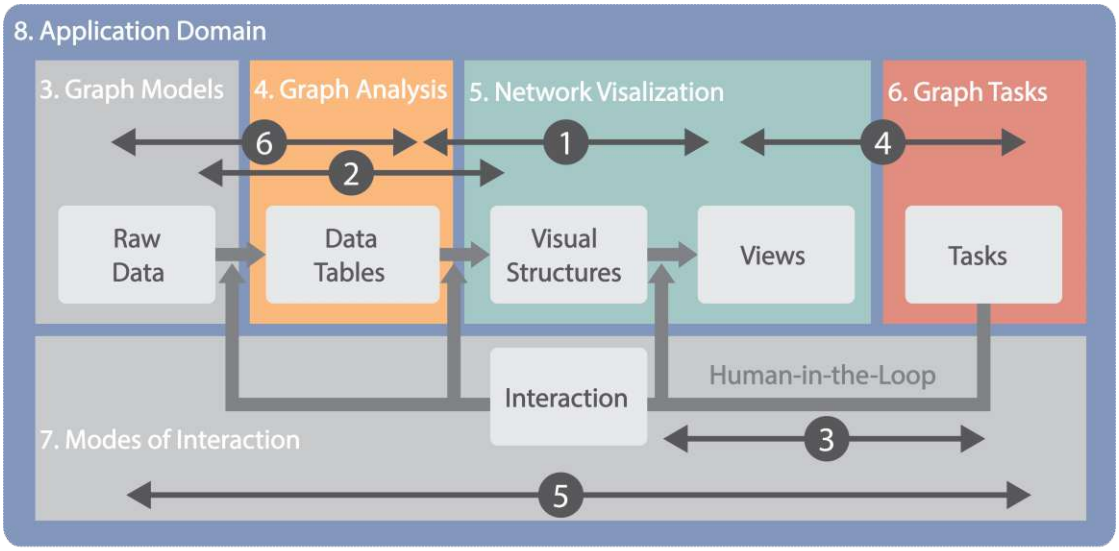


Figure 2.1: The biological network visualization pipelines (extended from Card et al. [Car99]) together with the classes of our taxonomy. This pipeline consists of six elements, namely *raw data*, which are processed to form *data tables*, which in turn are processed to form *visual structures*, which produce actual visual *views*. All these individual pipeline elements are connected by *Task*-driven *Interaction* techniques. To support the target domain, our report groups these pipeline elements into 6 classes, namely i) ***Graph Models***, which describe the structure of the raw data fed into the pipeline, ii) ***Graph Analysis***, which summarizes or provides additional insight into the raw graph data to facilitate more effective visualization or insight generation, iii) ***Network Visualization***, which transforms either the data tables or raw data into a structure suitable for visualization, iv) ***Graph Tasks***, which determine the tasks necessary for the analysis at hand, as well as v) how to best design modes of ***Interaction***, with which to achieve the analysis's goals, and, finally, vi) ***Applications***, which motivate and inform every aspect of the pipeline. Each of these six classes is discussed in its corresponding section, as noted by the numbers next to their section's title. Four of these sections form the basis of our taxonomy (Figure 2.2), namely those highlighted in color, i.e., ***Graph Analysis***, ***Network Visualization***, ***Graph Tasks***, and the various ***Applications***. Finally, the six identified challenges, discussed in Section 2.10, are highlighted in dark grey circles. Each challenge's arrows indicate which part of the visualization pipeline they affect.

We discuss each of these taxonomic classes in its corresponding section in order to provide an overview of core concepts, methodologies, and papers.

We systematically collected and then filtered papers in a step-by-step fashion. Table 2.1 depicts an overview of publication resources we investigated using API-driven queries (where possible). To constrain the focus of this report to visualization examples actually

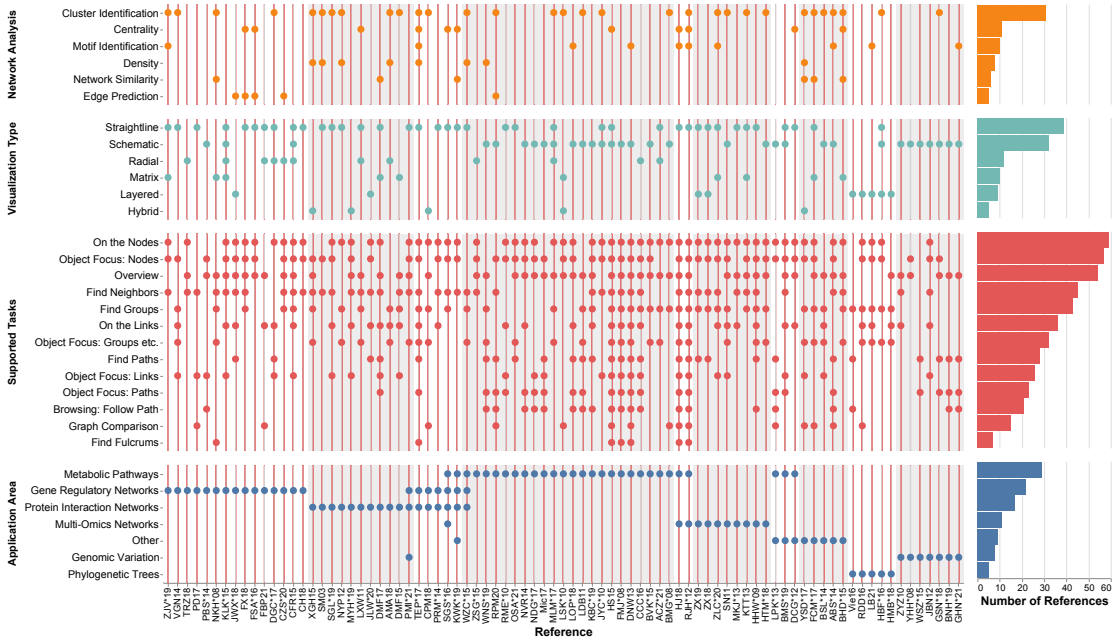


Figure 2.2: Classification of the collected and filtered 83 papers along the x-axis into the four *taxonomy classes*, i.e. i) *graph analysis techniques*, ii) *network visualization approaches*, iii) *graph analysis tasks*, and iv) *application domains*, which form the y-axis's four facets. Each of these categories, and subsequently their y-axis, is broken up further into the various subcategories that make up each of their sections, and ordered by their subcategories' totals. Papers are clustered along the x-axis according to their targeted application areas, marked in alternating bands of gray and white. Totals of each subclassification are shown as a bar chart on the right.

relevant to Card et al.'s [Car99] visualization pipeline, we only adopt full papers that are also relevant in this context. More specifically, we curated a final set of 83 papers (Figure 2.2) as follows: i) an initial list of over 700 papers was collected using API querying, ii) additional publications were manually collected from additional sources, iii) papers were manually refined, only keeping those that featured visualization as a primary focus, and,

Table 2.1: Table representing the literature search and sources for this survey.

Search Domain	**Sources**
Visualization	IEEE TVCG, CGF, IEEE VIS, EG EuroVis, IEEE PacificVis, GD, EG VCBM
Bioinformatics	Bioinformatics, PLOS Computational Biology, Briefings in Bioinformatics, BMC Bioinformatics, Frontiers
Digital Libraries	IEEE Xplore, Wiley DL, EG DL, ACM DL, PubMed DL

finally, iv) this trimmed set of publications was then categorized based on our developed information-visualization-pipeline-motivated taxonomy (Figure 2.1).

2.4 Graph Models

Complex relationships are often formulated using a *network* (often associated with various attributes) in applied areas, while a *graph* is a data structure expressing the fundamental connections between entities in mathematical terms [BGL11, WNV19]. In such a formulation, the vertices would represent biological entities, such as genes, proteins, or metabolites, and the edges connecting them would describe (functional) relationships between them. These formulations can include multiple types of vertices, (hierarchical) clusters or groupings of vertices, and different types of relationships [VM21, HM21]. In addition to the topological data themselves, data attributes, can be attached to provide extra information on certain aspects of the network, such as vertices, groupings, or edges.

In this section, we follow common strategies to use graphs to express the topological structure of networks discussed in the collected literature. The formal definition of a basic graph is defined as i) *simple graphs*, and its variant specialized for biological networks is defined as ii) *substrate graphs*. More sophisticated graphs, including iii) *k-partite graphs*, and iv) *hypergraphs*, provide specific properties to the relationships. v) *reaction graphs* simplify bipartite information to focus relationships of reactions. Finally, vi) *clustered graphs* and vii) *multilayer graphs* introduce simple and nested grouping information of the graphs.

2.4.1 Simple Graph

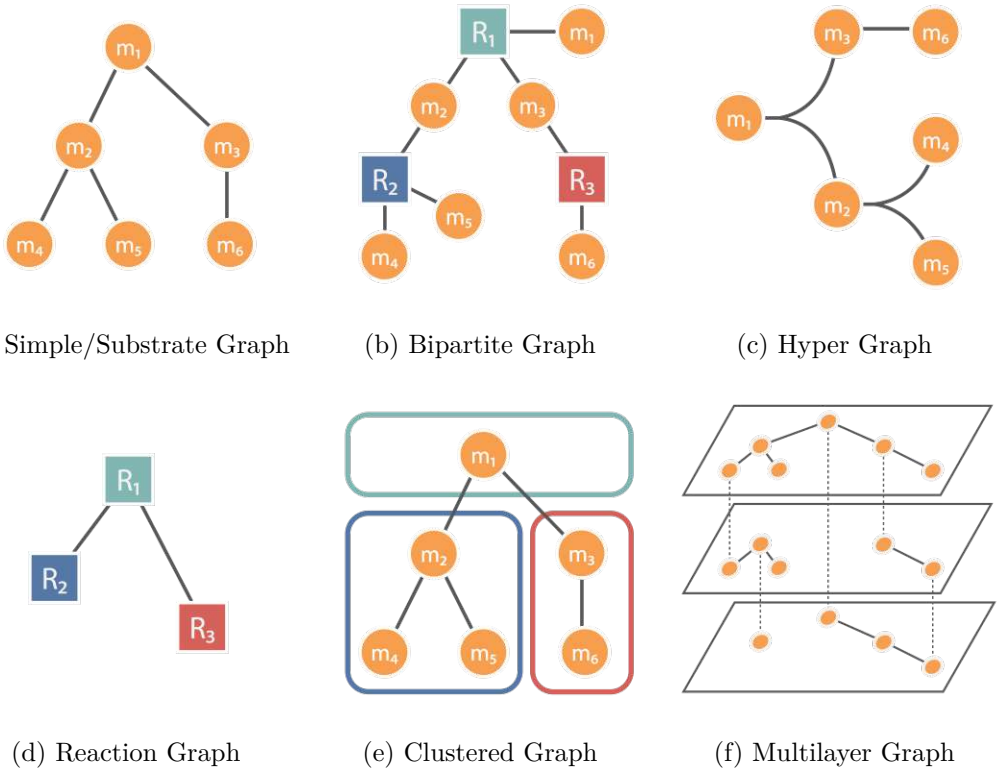
A complex relationship is often formulated as a *graph* to be manipulated mathematically [WDNM17]. The underlying graph data structure facilitates access to the data [SBV+21] so that analysis and visualization algorithms can perform efficiently.

Definition: The simplest graph model can be described as a tuple $G = (V, E)$, consisting of a set of vertices $V = \{v_1, v_2, ..., v_n\}$ representing individual entities. Their mutual connectivity is represented by the edges $E = \{e_1, e_2, ..., e_m\} \subseteq V \times V$.

This is a common definition that allows us to describe the fundamental relationships between entities for analysis and visualization purposes [BVK+15, FSA+16, KWKJ19]. A simple graph such as this can be shown as a linked list or an adjacency matrix [KKPEP20], or some hybrid format. They are often used interchangeably for different task purposes.

2.4.2 Substrate Graph

Although the general definition of a substrate graph/network could refer to its underlying physical infrastructure, in biology, a substrate graph refers to a system of interconnected biochemical reactions. The substrate graph is one of the pioneering graph structures



(a) Simple/Substrate Graph (b) Bipartite Graph (c) Hyper Graph

(d) Reaction Graph (e) Clustered Graph (f) Multilayer Graph

Figure 2.3: Graph models that are commonly incorporated in biological as well as general network analysis, i.e., a) *simple* or *substrate graphs*, b) *bipartite graphs*, c) *hypergraphs*, d) *reaction graphs*, e) *clustered graphs*, and f) *multilayer graphs*. Across all graphs, with the exception of (f), $V = \{m_1, m_2, m_3, m_4, m_5, m_6\}$ which can be placed in one of three groups $\{R_1, R_2, R_3\}$.

that has been used in the early development of biological pathway analysis and visualization [KG00, WDNM17]. Examples can be found later in Figures 2.11 and 2.13.

Definition: A substrate graph is structurally equal to the simple graph introduced previously. Nonetheless, since enzymes binding with chemical reactants are called substrates (denoted m_i in Figure 2.3(a)), each $v \in V$ in a substrate graph can represent a single reactant (e.g., metabolites in pathways) or multiple reactants together with enzymes as a single vertex. Each $e \in E$ in a substrate graph can describe a reaction between the substrates, indicate a regulatory interaction, or present a movement of substrates across cellular compartments (e.g., transport pathways).

Since a substrate graph is a simple graph with different definitions in vertices and edges, topologically, it is straightforward to implement and maintain a substrate graph as many known algorithms can be easily applied.

2.4.3 K-partite Graph

A k-partite graph is a graph whose vertices are partitioned into k different disjoint sets. Bipartite graphs are specific types of k-partite graphs, where $k = 2$, and are common representations for biological pathways. Some biological pathways are formulated as bipartite graph [NDG+17], where, for example, a vertex can be either categorized as a metabolite vertex (m_i) or a reaction vertex (R_i), but not both, as shown in Figure 2.3b. Examples can be found later in Figures 2.15 and 2.14.

Definition: A graph $G = (V, E)$ is k-partite if and only if there exists a vertex partition $V = P_1 \cup P_2 \cup \cdots \cup P_k$ and $P_m \cap P_n = \emptyset$ for any $m \neq n$. Furthermore, for each edge $e = (v_i, v_j) \in E$ we have $v_i \in P_m$, $v_j \in P_n$, and $m \neq n$, which guarantees that the end vertices of an edge do not belong to the same vertex set.

A k-partite graph (often a bipartite graph in the context of a biological network) allows us to highlight groups vertices using colors or positions in visualization since they are disjoint. This facilitates matching and comparison in the analysis.

2.4.4 Hypergraph

In principle, a hypergraph is a more intuitive and direct representation of biological pathways, and several biological network notations, such as SBML [HFS+03], BioPax [DCP+10], and KGML [KG00] support it. A hyperedge in a hypergraph can refer to a single biochemical reaction, in which multiple metabolites are involved (Figure 2.3c). Although hypergraphs can be converted into bipartite graphs, and vice versa, due to the difficulty of data management, only a few tools support hypergraph representations [WDNM17].

Definition: A hypergraph $G = (V, E)$ consists of a set V of vertices, and a set E of hyperedges, which are non-empty subsets of V. Formally, $E \subset 2^V$ is a subset of the power set of V.

The advantage of hypergraphs is that hyperedges can model k-ary relationships, while classical graphs can only model binary relationships. For example, users can spot what enzymes are involved in a reaction by observing a hyperedge connecting all of them. Most formats of biological networks record necessary information for building a hypergraph. However, an implementation of hypergraphs and analysis on a hypergraph are not as intuitive as other data structures.

2.4.5 Reaction Graph

A reaction graph is a simple graph, structurally, but has different semantics in comparison to a substrate graph. Vertices here are distinct reactions, and the edges are metabolites represented in the network (see Figure 2.3d). Reaction graphs have been used in computational models in early research efforts [SJ08].

Definition: Vertices $v \in V$ in a reaction graph G represent reactions in biological networks, edges $e \in E$ stand for metabolites involved.

A reaction graph is predominantly used for topological analysis, such as shortest path analysis or centrality analysis so that users can rank graphs according to important concepts in the field [HS15].

2.4.6 Clustered Graph

In addition to simple relationships, functional groups or categories can be assigned to vertices or edges. Such groups are often categorized hierarchically by domain experts or analysis approaches [NDG+17]. An example can be found later in Figure 2.14, while the color boxes show the gouping information.

Definition: A clustered graph (Figure 2.3e) is a simple graph G with additional grouping information. Each $v \in V$ in a clustered graph belongs to one or more clusters $c \in C = \{c_1, c_2, ..., c_k\}$.

In other words, clusters can also form a hierarchy using a cluster tree T, whose leaf set is V and inner vertices are clusters of all leaves in the subtree. In some definitions of clustered graphs, all clusters in C are disjoint and form a partition of the vertex set V [FT04, SN11]. Nonetheless, in the context of biological pathways, clusters c are not necessarily disjoint. For example, ATP, a universal energy molecule occurring in mitochondria and cytoplasm, is often used to drive several biological reactions. If we consider these compartments as clusters, they are overlapping since ATP can be transported from the mitochondria to the cytoplasm. Another example would be the relationships of ATP in the biological ontology. Since ATP occurs in many categories of biochemical reactions, including *Citric Acid Cycle* and *Urea Cycle*, its representation should be covered by multiple clusters in the model. In some cases, to simplify the visual complexity of clustered graphs, biologists duplicate unimportant vertices (e.g., vertices with high degrees) to create a specific type of clustered graph, where aliases of an identical vertex only belong to a corresponding cluster. In other words, clusters become disjoint in this case [WNSV19].

2.4.7 Multilayer Graphs

We can transform a cluster graph with complex grouping information into a multilayer graph. When it comes to advanced analysis, the term used for describing more nested relationships between entities beyond clusters is often called multiplex, while other terminologies such as *multilevel*, *multivariate*, *multidimensional*, *multirelational*, or *network of networks* that describe similar concepts are also used in current research [KAB+14, MMF17]. As summarized by Kivela et al. [KAB+14], the above terms can be reframed and encapsulated by the definition of a *multilayer network*. Therefore, one can use multilayer networks as an umbrella term to cover the aforementioned graph models in the field of biological networks. A *multilayer graph* is a simple graph G with additional layer information to describe real-world properties of the network in a whole [KAB+14]. Layers in multilayer

(Figure 2.3f) networks are used to describe the corresponding relationships, where each of which records the property of the corresponding relationships. In this state-of-the-art report, we follow the formal definition by McGee et al. [MGM+19, MGO+21].

Definition: Since each $v \in V$ can belong to several layers, we can consider vertices as pairs $(v, l) \in V_M \subseteq V \times L$, where L is the set of associated layers. Edges $E_M \subseteq V_M \times V_M$ indicate the connectivity of pairs (v_i, l_p), (v_j, l_q). An edge is considered as an intra-layer edge when $l_p = l_q$ or an inter-layer edge when $l_p \neq l_q$, respectively.

In biological networks, we could have $L = \{l_1, l_2, l_3, ..., l_p\}$, where l_1 could be metabolites occurring in mitochondria, and l_2 could be metabolites existing in the cytoplasm, and so on. Note that some metabolites, such as H_2O, which occur in both mitochondria and cytoplasm, can be connected using an inter-layer edge. This formulation becomes powerful in the sense that it covers existing concepts and can be further used as an intermediate form to transform one concept to another, not only as a model but also visually [KAB+14]. In practice, we can use multilayer graphs as a unified graph structure because the graphs described in Sections 2.4.1-2.4.6 are also multilayer graphs for a specific layer set L. Thus, researchers can always convert the aforementioned graphs to multilayer graphs and again convert them to the target graph data structures. This scheme allows us to perform a systematically consistent conversion to different graph representations, as well as using the multilayer graph as a standard diagram when compared to other visual representations. Examples can be found later in Figures 2.16 and 2.17.

2.4.8 Graph Data Structure in Practice

Classical biological analytics tools cover the subsets of the aforementioned graph data structures. Cytoscape [SMO+03] is a general-purpose visualization software for complex networks, and several plugins for biological networks have been integrated. The underlying data structure of Cytoscape utilizes the well-known graph editor yFiles [WEK02]. The COBRA Toolbox [HAP+19] integrates MATLAB for quantitative prediction of cellular biochemical networks. The toolkit incorporates CellDesigner [FMKT03], which is a graphical editor designed for gene-regulatory and biochemical networks. Reactome [VDS+07] is an open-source and peer-reviewed knowledge base of biomolecular pathways. The visualization tool ReactomeFIViz [WDD+14] implements several functions for network-based data analysis, and the graphs are extended from Cytoscape [SMO+03]. BioCyc [PK21] is a pathway and genome database, that integrate Pathway Tools [PBH+21], which facilitates genome data management, systems biology, and omics data analysis. WikiPathways [MAR+21] is a community-based biological pathways database, which integrates PathVisio [KVIB+15], allowing visualizing, editing, and analyzing biological pathways. The aforementioned tools and most literature collected in this survey support primarily property graph infrastructure for single graphs, k-partite graphs, and cluster graphs through user intervention. For hypergraphs, the software often requires graph conversions

(e.g., vertex and edge duplication) to simple graphs for maintenance purposes. More advanced structures, such as multilayer graphs, have not yet been widely used.

2.5 Network Analysis

The general goal of network visualization is to convey or extract information regarding the underlying data effectively. However, with biological datasets growing in size and complexity, straightforward visualizations may no longer suffice in aiding researchers. Instead, if visualization tools are to meaningfully assist domain experts, it may be necessary for them to include certain analysis approaches that pre-select, summarize, or analyze the data (semi-)automatically. Different domains, data, and networks bring with them different analytical goals and analysis tasks, each of which may require different analytic strategies to address. In this section, we aim to give the reader an overview of some of these many network analysis tasks, as well as some common approaches and techniques used to tackle them.

Specifically, we first provide an overview of some simple but useful, descriptive metrics commonly used in the visual analysis of networks, namely *graph density*, *vertex centrality*, as well as some common *network similarity measures*. Additionally, beyond such descriptive approaches, researchers are often also interested in investigating groups of vertices and the ways in which they interrelate, often achieved using *motif identification* or *clustering*.

2.5.1 Density

The density of a (sub)graph $G = (V, E)$ quantifies how many edges $|E|$ it has compared to the maximum possible number of edges in a complete, here undirected, graph with the same number of vertices $|V|$, i.e., $|E|_{max} = \frac{|V|(|V|-1)}{2}$ edges [KKPEP20]. By comparing this hypothetical quantity to the actually observed number of unique edges, one can calculate the graph's density, formulated by Pavlopoulos et al. [PSM+11] as

$$density = \frac{|E|}{|E|_{max}}. \tag{2.1}$$

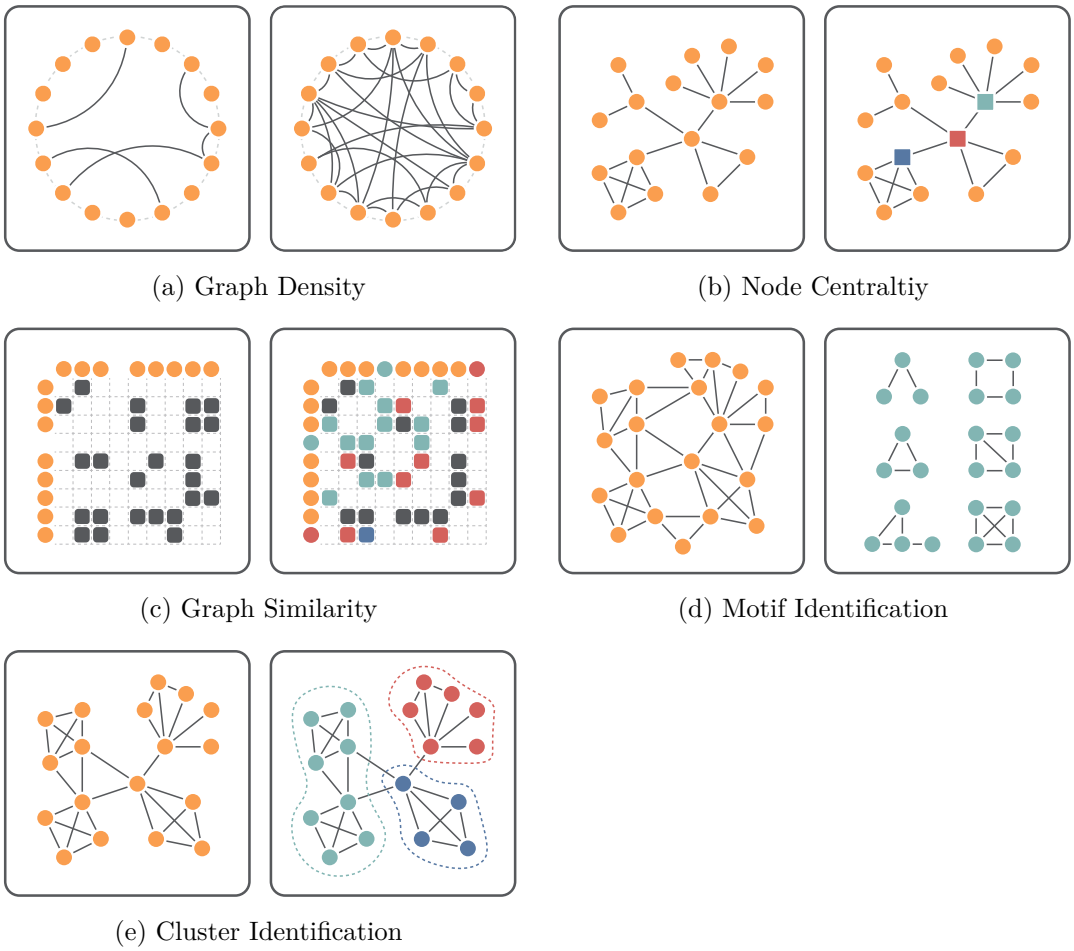
As a rule of thumb, a graph can be considered dense if $|E| = \omega(|V|)$, i.e., it has a superlinear number of edges; otherwise, if $|E| = O(|V|)$, it may be considered sparse [KKPEP20]. To make this concept more tangible, consider Figure 2.4a in which two graphs are shown, one of low density (left), i.e., $G_{low} = 5/|E|_{max} = 0.041$ and one of relatively high density (right), i.e., $G_{low} = 30/|E|_{max} = 0.25$, where $|E|_{max} = 16(16-1)/2 = 120$. The exact interpretation and importance of a graph's density depend on the type of biological network under study and the analysis goals.

(Sub)graph density estimation finds application in both the non-visual analysis and visualization of biomedical network data. First, in analysis, density is a natural choice to compare identified subgraphs. Thus, it finds regular use in vertex clustering applications,

be it as i) a set of weights for each vertex estimated from each vertex's local neighborhood's subgraph [BH03], ii) the actual metric upon which the vertex clustering is based [LLT+17, HHW+09], or iii) a means of comparing and evaluating the identified clusters' structures [LH08]. In visualization applications, however, (sub)graph density has taken on a number of varied roles. In the simplest case, its use can also be as straightforward as a metric to compare clusters and even entire graphs. For example, both Koutrouli et al.'s *NORMA* tool [KKPP21] and Theodosiou et al. [TEP+17] present density among other common topological summaries, such as the number of edges and nodes, the clustering coefficient, or various *centralities*, in order for users to quickly evaluate and compare (sub)graphs. Moreover, as density provides a natural way of summarizing the "quality" of a cluster compared to others, the metric is regularly used to rank identified clusters [SM03], or allow users to filter clusters from the visualization whose density is below some user-set threshold filtering of identified clusters [NYP12, CY06]. Alternatively, density values can also be used to guide user attention. On one hand, this can be as simple as highlighting aspects of the graph based on these metrics. For example, Chang et al. [CZSX20] highlight miRNA "modules" based on their edge density in order to draw visual attention. On the other hand, densities have also been used to produce new, simplified visualizations of some input embedding in order to better guide users to potential regions of interest. For example, Ebbels et al.'s *springScape* [EBJ06] utilizes an embedding's vertices' 2D coordinates as well as its subgraph densities to produce 3D "density landscapes" representations of the microarray data, in order to allow domain experts to better identify regions of interest.

2.5.2 Centrality

Beyond looking at an entire (sub)graph's density, one may also be interested in identifying or ranking its important vertices, be it to select targets of potential biological value [ZLZ+20], or to reduce the dimensionality of the problem [RM16]. This ranking can be based on some selected structural features of the network, by utilizing one (or multiple) of the many available measures of centrality [PSM+11], as exemplified by the *VINCENT* framework [KKZ12, ZJK12]. Depending on the biological question posed, certain structural features are more important than others, and thus different measures of centrality may be of greater utility than others. For example, one may be interested in identifying highly connected hub proteins in protein-protein interaction networks [ZMOP08], genes and motifs important within genetic regulatory networks [KS08b], or proteins crucial for the network's overall robustness to perturbation in metabolic engineering [MZ03]; each of which requires a different type of centrality. Consider, as an example, Figure 2.4b, in which, given some input graph (left) three different nodes of importance are highlighted based on different topological properties (right): the degree of the node, the number of shortest paths that pass through the node, and the topological importance of the node's neighbors. While many forms of centrality have been developed (see [GST+14] for a fairly exhaustive review), we will briefly enumerate and discuss some of the most common and simpler types of centrality to illustrate how varied and useful their applications can be.



(a) Graph Density (b) Node Centraltiy

(c) Graph Similarity (d) Motif Identification

(e) Cluster Identification

Figure 2.4: Visual illustration of the 5 discussed graph analysis metrics and approaches, i.e a) *graph density*, b) *node centrality*, c) *graph similarity*, d) *motif identification*, and e) *(node) cluster identification*. A) showcases two graphs, one of low edge density (left) and one of high density (right). B) highlights, given a simple input graph (left), the three identified nodes with the highest degree, betweenness, and eigenvector centrality displayed as **turquoise**, **red** and **blue** square nodes respectively. C) displays two graphs to be compared, a reference (left) and a second graph (right) in which all node and edge additions and removals are highlighted in **blue** and **red**, respectively. D), given some input graph (left), displays all identified motifs of size three and four (right) in **turquoise**. Finally, e), given some input graph (left) showcases the results of a hypothetical clustering (right) in which the three identified clusters are colored in **turquoise**, **red**, and **blue**.

Degree centrality [Bon87] simply measures how well-connected a vertex $v \in V$ is:

$$C_{degree}(v) = deg(v), \tag{2.2}$$

where $deg(v)$ is the total number of edges connected to vertex v. Thus, the more edges to

other vertices v has, the higher its centrality. Such vertices with high degree centrality are often referred to as *hubs* and can be interesting biologically as their removal can greatly alter the network's overall topology [PSM+11]. For example, Zotenko et al. [ZMOP08] utilized degree centrality to quantify how essential proteins are, in order to investigate the correlation of proteins' "essentiality" with the "lethality" of their removal from the system. Alternatively, Chang et al.'s [CZSX20] *miRNET* and Kuijpers et al.'s [KWKJ19] *DynoVis* utilize degree centrality as a measure of importance to rank and identify vertices of interest.

Closeness Centrality describes the mean distance from a vertex to other vertices by computing the average shortest distances between the current vertex v and all other vertices in $V \setminus \{v\}$ [New10]:

$$C_{closeness}(v) = \frac{1}{\sum_{w \in V \setminus \{v\}} dist(v, w)}, \tag{2.3}$$

where $dist(v, w)$ is the length of the shortest (hop or weighted) path between vertices v and w. Intuitively, the higher a vertex's closeness centrality score, the closer a vertex is to all other vertices in the graph. Commonly, closeness centrality is often used to identify central, and thereby important, metabolites in large metabolic networks [PSM+11, KKPEP20], though it also has seen use in gene regulatory networks [KS08b]. For example, da Silva et al. [dSMZ08] utilized closeness centrality to identify metabolites crucial to the functioning of genome-scale metabolic networks in a variety of organisms. Beyond identifying important vertices, this measure has also been used to analyze the structure [MZ03] and evolution [MBSB10] of metabolic pathways.

Betweenness Centrality [Fre77] quantifies how many of the shortest paths between any two vertices pass through the vertex v of interest:

$$C_{Betweenness}(v) = \frac{\sigma_{wu}(v)}{\sigma_{wu}}, \tag{2.4}$$

where σ_{wu} is the total number of shortest paths connecting all pairs of vertices w, u in $V \setminus \{v\}$, and $\sigma_{wu}(v)$ is the total number of those paths that pass through the current vertex of interest v. In protein-protein interaction networks, this type of centrality is often used to identify proteins that form bridges, as well as bottlenecks, in the network's topology [PSM+11, KKPEP20]. An example thereof can be seen in Joy et al.'s [JBIH05] investigation of high-betweenness proteins and those proteins' evolutionary as well as functional importance. Researchers studying cancer have also utilized betweenness centrality to identify crucial metabolites in signaling pathways [LZ13] and essential genetic drug targets [DPG+19].

Eigenvector Centrality quantifies how connected a vertex is to important other vertices [PWS08], i.e., the more important a vertex's neighbors, the higher its centrality value. Bonacich [Bon07, Bon72] formulates this metric intuitively as a weighted sum of a vertices' direct and indirect connections' centralities. Because of eigenvalue centrality's weighing of information beyond the immediate adjacency of a vertex, this particular centrality

has been used, for example, to identify crucial protein pathways involved in biological processes [NMH+18]. It has also been used to identify as well as predict gene-disease associations [OVER08] as well as study connectivity patterns in human brain fMRI data [LMH+10]. Finally, note that *(weighted) PageRank (centrality)* can be understood as eigenvector centrality for directed graphs specifically [ZWY22, Ber05].

Lastly, ***Eccentricity Centrality*** computes the accessibility of a vertex v from all other vertices in the graph:

$$C_{Eccentricity}(v) = \frac{1}{\max_{w \in V}(dist(v, w))} \tag{2.5}$$

Unlike closeness or betweenness centrality, eccentricity does not consider the sum or average across all vertices in the graph. Instead, it only considers the largest value, which makes it sensitive to outliers [GST+14]. Nonetheless, this measure has been important in identifying essential proteins in protein-protein interaction networks, as an easily reachable protein, i.e., a protein with a high eccentricity score, is sensitive to changes in other proteins' concentrations [JSYG+16].

In summary, visualization tools aimed at biological network analysis or exploration, across biological domains, often feature centrality as a means of screening, ranking, and/or highlighting nodes with exceptional centrality scores. *VANTED* [RJH+12, HJ18], *DynoVis* [KWKJ19], and *OmicsNet* [ZX18], for example, all allow users to rank network nodes using degree centrality to detect network hubs within the context of systems biological data; though *DynoVis* and *OmicsNet* feature multiple types of centrality measures where *VANTED* does not. *miRNET* [FSA+16] features very similar ranking and highlighting functionality to *VANTED*, but tailored to miRNA data specifically. Within the context of protein-protein interaction networks, *cytoNCA* [TLW+15] provides eight centrality measures with which to screen and highlight vertices of interest. Lastly, *CentiBin* [JKS06] allows for the investigation of multiple centrality distributions in order to better select cutoff values and thereby better select nodes of interest, in addition to the aforementioned highlighting and ranking. Some visualization tools go beyond "just" ranking and highlighting. For example, *MetPA* [XW10] utilizes betweenness centrality scores to facilitate pathway enrichment and pathway topology analyses, the results of which are then presented visually in order to ensure their validity.

2.5.3 Network Similarity

Beyond quantifying a network's density or ranking its vertices, we may also be interested in comparing two networks. Consider, as an example, Figure 2.4c: given some input graph displayed as an adjacency matrix (left), we highlight all changes, i.e., additions and subtraction, in turquoise and red respectively. However, such purely visual inspections are, of course, limited and not the only way to compare graphs of interest. One could, for example, compare the difference in two networks' densities. However, more sophisticated (visual) approaches have been put forth over the years to address this challenging topic

[ESDS16], which Sugiyama et al. [SGLLB18] group into alignment-based or alignment-free methods. Given the depth and breadth of this topic, we refer the reader to the recent review by Tartardini et al. [TITP19] for an overview of available approaches. Owing to their predominance in biological applications, we opt to focus on comparing networks with known-vertex correspondences, i.e., two or more networks that share the same set of vertices[SERG14]. More specifically, in this section, we aim to provide an overview of three such known-vertex correspondence approaches discussed by Tantardini et al. [TITP19], namely i) Differences in Adjacency Matrices, ii) the DeltaCon metric, and iii) the Cut distance.

As will be discussed later, an ***Adjacency Matrix*** communicates whether, and (if weighted) with what weight, two vertices in a graph are adjacent to each other. More specifically, each row and column corresponds to a vertex in the graph, and each matrix cell represents the presence/absence or weight of an edge between vertices in the corresponding row and column. For undirected graphs, these matrices are symmetric; for directed graphs, they are not. If two graphs share the same set of vertices, then one can simply calculate the distance between their adjacency matrices based on some matrix norm. While a straightforward and simple technique, such differences can already provide a good first look at a network comparison problem [TITP19].

The ***DeltaCon Measure*** computes pairwise vertex similarities within each network, which can then be used to calculate the similarity between the two networks [KSV$^+$16]. First, the similarity between vertices v and u is defined as $S[vu] = [I + \epsilon^2 D - \epsilon A]^{-1}$, where ϵ is some small, positive constant, I is an identity matrix, D is the diagonal of vertex degrees, and A is the network's adjacency matrix [TITP19]. The element $S[vu]$ quantifies the influence that vertex v has on u, i.e., the more paths connect the two, the higher their influence on one another [KSV$^+$16]. With all pairwise vertex similarities computed in both networks to be compared, one can construct the similarity matrices, S_1 and S_2, of the two networks. Second, using each network's similarity matrix, the two networks' distances from one another can be calculated using, for example, the Matusita distance, though other distance or similarity measures could theoretically be used [KSV$^+$16]:

$$d = d_{Matusita}(S_1, S_2) = \sqrt{\sum_{v \in V} \sum_{w \in V} (\sqrt{S_1[vu]} - \sqrt{S_2[vu]})^2} \tag{2.6}$$

Compared to a simple difference in adjacency matrices, this approach satisfies four desirable properties, namely i) changes that disconnect vertices or sub-graphs are more heavily penalized than changes that maintain connectivity, ii) the bigger the weight of an edge, the larger its influence on similarity should it be removed, iii) highly specific changes in a graph with few edges are more influential than in graphs with many edges, and iv) random deletions or additions to a graph are not as influential on the similarity score, compared to targeted ones [KSV$^+$16].

Alternatively, given two input graphs with the same set of vertices but different edges connecting them, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, and two disjoint sets of vertices,

$S, T \subset V$, the ***Cut Distance***, based on the maximum cut weight [FK99], measures similarity based on the differences in cut edge weights across all possible bi-partitions of S and T in graphs G_1 and G_2 [TITP19]. More specifically, the method attempts to find the non-minimal sets of edges to be removed from each graph, i.e. $C_1 \subset E_1$ and $C_2 \subset E_2$, in order to maximize the difference between each graph's sum of cut edges' weights:

$$d(G_1, G_2) = \max_{S \subset V} \frac{1}{|V|} |e_{G_1}(S, S^c) - e_{G_2}(S, S^c)| \tag{2.7}$$

where $S^c = V \setminus S$, and $e_G(S, S^c) = \sum_{v \in S, u \in S^C} \beta_{vu}(G)$, and $\beta_{vu}(G)$ is the weight of the edge connecting vertices v and u in graph G [LDW18]. A key advantage of the Cut Distance, over other measures of network similarity, is its ability to function with both directed and undirected, as well as weighted and unweighted graphs. On the other hand, a key disadvantage is its computational complexity, making its application to large networks (common in biology) unfeasible [TITP19].

Despite many visualization tools being published for the purpose of comparing biological networks, for example *NetConfer* [NBKM20], *CoExpNetViz* [TDDM$^+$16], or *DynoVis* [KWKJ19], such tools primarily provide a *visual* comparison of networks and their metadata only. Owing to adjacency matrices' straightforward visualization and interpretation, a number of tools, across various biochemical domains, have been published that feature them. Within the context of comparative visualization specifically, such tools most commonly feature side-by-side views of (sub-)graphs. On the one hand, for example, within the context of general time-dependent graphs, Bach et al's *Small Multi-Piles* [BHRD$^+$15] provides an interactive visualization of adjacency matrices; viewable juxtaposed or interchangeably [EMWR24]. On the other hand, for brain connectivity studies, Yang et al. [YSD$^+$17] visualize brain subgraphs across experimental conditions as side-by-side adjacency matrices. Similarly, New et al. [NKHC08] visualize subgraph adjacency matrices side-by-side in order to study differences in genetic co-expression. However, the calculation of adjacency matrix *differences* appears to be less common. One example, Dang et al.'s *BioLinker* [DMF17], visualizes conflicts, i.e., differences, in literature/database-derived interactions between biological entities as a heatmap adjacency matrix in order to facilitate a more complete understanding of possible protein interaction patterns. Almost no published tools were found in this study that employed any of the more complex analytical comparison techniques, i.e., DeltaCon or Cut Distance. One example of the use of the Cut Distance is Theodosou et al.'s *NAP* [TEP$^+$17], which allows users to calculate the minimum *st*-cut between two selected nodes, not for explicit comparison, but for general topological analysis of biological networks.

2.5.4 Motif Identification

Motifs are structures that are statistically over-represented relative to some null model that describe a particular pattern of interaction between vertices in a network [KKPEP20]. Sometimes described as the "building blocks" of larger networks [MSOI+02], these substructures can assist in answering a multitude of biological questions, from identifying and predicting the biological function of a network's subunits to disease discovery based on known and predicted motif function [PM20a]. To appreciate motifs' nature as the building blocks of a network, consider Figure 2.4d: given some input network (left), several motifs (of sizes $|V| = 3$ and $|V| = 4$ nodes) are identified and highlighted (right). While a complete discussion of the many possible approaches to motif identification is beyond the scope of this STAR, we discuss some of the core concepts and challenges. Generally, the process of motif identification can be broken down into three distinct steps; i) the counting of all subgraphs of various sizes present in the network under consideration, ii) the calculation of the uniquely identified subgraphs' frequencies of occurrence, and iii) the determination of those frequencies' statistical significance [WBQH12, PM20a].

The first of these three steps, the identification of unique subgraphs, can already form a problematic computational bottleneck in motif identifications, as the number of possible subgraphs increases exponentially with the maximal motif size [WBQH12]. Generally, subgraph census approaches can be placed into one of two categories: first, exhaustive searches, usually using pattern growth trees, which do not scale well with large networks or subgraph sizes; or second, heuristic or probabilistic searches, such as, but not limited to, probabilistic sampling of edges or vertices, or mapping strategies. The key trade-off lies between the method's computation time and the accuracy of the census.

Once all present subgraphs have been counted, one must ensure they are accurately classified into unique isomorphic motif classes. While some subgraph census algorithms already categorize found motifs by their unique isomorphic forms, not all do [BGP+13, CFSV04]. A discussion of these algorithmic approaches is beyond the scope of this STAR, but the reader is referred to Ehrlich et al. [ER11] for an overview.

Lastly, once all unique motifs' frequencies in the input network have been calculated, one must evaluate whether, and which, of these frequencies are statistically significantly different from those estimated from some random graphs. Doing so requires the repeated simulation of random networks, and repeating the aforementioned first and second steps in order to estimate each randomly generated graph's frequencies of unique subgraphs. This final step, which forms the most computational hurdle in network motif identification [WBQH12]. It also poses the greatest challenge analytically, as the selection of a sensible generative model for these random graphs is non-trivial, as a researcher must consider what assumptions they are willing to make about the underlying generative process, as well as the extent to which these assumptions can be validated. One straightforward approach is selecting an assumed applicable random graph null model, such as permutations of the input network. Two common approaches are the so-called "Switching Method" and "Matching Method" [WBQH12]. The "Switching Method" repeatedly randomly selects

two edges, and subsequently exchanges their ends in order to create a differently connected version of the input graph. The "Matching Method" randomly reconnects all vertices while keeping each vertex's number of incoming and outgoing edges consistent with the original network. Lastly, for particular application areas, one can also consider (non-)parametric resampling approaches [MBAR13]. Often called "bootstrapping" approaches, these rely on repeatedly drawing a random (sub-)set of vertices from the original graph [SBo99], or specific sub-regions of the original network [CGLN19].

Motifs and their identification find use across biological domains, though they are used for a broad range of purposes, from analysis to evaluation. For example, within the context of gene regulatory networks, Zarnegar et al. [ZJVS19] identify motifs in order to better understand both gene expression and functional associations. For metabolic networks, Droste et al. [DNW13] identified so-called "motif-stamps" in order to better guide the automated drawing of the metabolic network. In the context of multi-omics networks, both Rohn et al. [RJH+12] and Zander et al. [ZLC+20] provide motif identification to facilitate more complete topology-based analyses of multi-omics networks. In Protein-Potein Interaction Networks, Spirin et al. [SM03] identify motifs in order to discover novel molecular modules in protein-protein interaction networks. Within the context of genomic variation graphs, Guarracino et al. [GHN+21] utilized motif identification to understand complex pan-genomic relationships between sequences of DNA. Lastly, Al-Awami et al. [AABS+14]'s *NeuroLines* tool identifies repeated connectivity motifs in the brain in order to identify potentially biologically interesting synaptic pathways. Even when motifs are not explicitly incorporated into an analysis or visualization, they are frequently used to evaluate the results obtained [DGC+17, MYH+19, TEP+17]

2.5.5 Cluster Identification

Clusters are vertices grouped together based on common properties [PSM+11]. This grouping attempts to ensure objects within the cluster are as homogeneous as possible while separating objects with different properties into distinct clusters [SA17]. Consider, as an example, the clustering shown in Figure 2.4e: given some input graph (left), we identify three clusters of nodes, shown in turquoise, red, and blue (right). Application areas in biology include, for example, the identification of clusters of proteins in protein-protein interaction networks that may be functionally involved in a similar biological process and thus form a biological complex [MP08]. However, owing to the computational complexity of the clustering problem, as well as the diversity of data and analytical goals, many approaches have been put forth over the years [SA17]; though only a handful are realistically applicable to the large problems encountered in modern biological applications [KKPEP20]. We highlight three noteworthy categories of techniques; i) k-partition approaches, ii) hierarchical approaches, and iii) density-based methods [WBR15, AAWS09]. These approaches do not necessarily provide *exclusive clustering*, i.e., a one-to-one mapping of vertices to clusters, but can also produce *probabilistic*, *overlapping/fuzzy*, or *hierarchical* clusters [PSM+11].

k-Partition Clustering, as the name implies, aims to partition the graph's vertices into k clusters [PSM+11]. Such approaches are useful if a user is seeking a computationally cheap approach to clustering, or has a particular number of clusters, k, already in mind. Starting with some initial assignment of each vertex into one of the k clusters, k-partition clustering iteratively minimizes some dissimilarity measure within each group. The perhaps most well-known of these techniques is k-Means Clustering, which minimizes the within-cluster sum of squares of some distance function, i.e., the mean difference in distance between each vertex in some group, and that group's overall mean [LVJV03]. While simple, this approach does bring with it a number of disadvantages. It is sensitive to its initial random assignment of vertices within clusters, meaning the algorithm will produce a different final clustering depending on the initialization. Moreover, the selection of k presents a challenge, i.e., the method will always produce k clusters, and a vertex's cluster assignment can vary greatly for different selections of k. Additionally, owing to its use of the mean distance measure, it is sensitive to outliers [AAWS09]. To address the shortcomings of conventional k-Means clustering, a number of extensions and alternatives have been developed over the years. Consider, for example, k-medoid clustering, which utilizes the distance between each cluster's vertices and the cluster's *median*, instead of the mean, thus making it less sensitive to outliers. Other examples, which attempt to address other shortcomings of k-means clustering include, but are not limited to, Fuzzy k-Means, Kernel k-Means, or Farthest First Traversal k-Means [AAWS09, Jai10]. Of all the partitioning techniques, k-Means Clustering has found use in biological applications, most likely owing to its conceptual simplicity and availability. For example, within the context of protein-protein interaction networks, Barsky et al. [BMGK08] featured k-Means Clustering in order to identify proteins and/or genes with similar expression profiles across experimental conditions. However, k-medoid clustering has also found application in biochemical application areas, such as in Mildau et al. [MEO+24] *SpecXplore* tool for the interactive exploration of tailored mass spectral data.

Biological graphs may have hierarchical structures of potential interest within them, or a researcher may be interested in evaluating estimated groupings at multiple levels before selecting a single level to investigate [KKPEP20]. In such a case, ***Hierarchical Clustering*** approaches can be useful to create multilevel groupings based on vertex similarities [KN17]. With all pairwise similarities calculated, one can then iteratively group in increasingly large, hierarchical clusters, i.e., agglomerative clustering. Alternatively, these pairwise similarity measures can be used to iteratively break the graph into increasingly small clusters in a "top-down" fashion [PSM+11]. Either way, this step-wise grouping based on some measure of vertex similarity can consider, for example, *Single*, *Average*, or *Complete Linkage*, i.e., the smallest, average, or largest distance between all pairs of objects, respectively; though additional forms of linkage exist [KKPEP20].

Various forms of hierarchical clustering for multiple applications can be found in biomedical (visualization) literature. Generally, such techniques seem to be found frequently in exploratory (visual) analyses, as interactively setting the clustering threshold can reveal different relationships between entities. For example, Cruz et al. [CPAM18] im-

plemented hierarchical clustering to allow users to explore different relationships of a node across clustering hierarchies in dynamic gene expression data. Similarly, Bartell et al. [BYCH+12] and Varemo et al. [VGN14] provide similar clustering, though for gene-set and SNP datasets, respectively. Beyond molecular networks, Riaz et al. [RPM20]'s tool *MAPPS* allows for agglomerative hierarchical clustering in order to group organisms based on their overall metabolic network similarities. These clusters can allow researchers to explore hypotheses relating to metabolic network similarity and specialization.

Based on the input network's geometry, ***Density-Based Clustering*** approaches stratify the input network into groups of vertices of high density, separated by regions of low density [AAWS09]. As illustrated by Kriegel et al. [KKSZ11], these methods "cut" the 3D probability density functions produced by a 2D input graph in order to identify groups of clustered vertices. For networks that do not have some intrinsic spatial interpretation, i.e., the majority of biological networks, one must select an appropriate (dis)similarity function in order to define some data space. This "cut"'s density value must be considered carefully; too high and low-density clusters are lost; too low and only a single large cluster will be identified [KKSZ11]. With the exception of this "cut" value, density-based clustering techniques are (largely) non-parametric. This lack of (hyper-)parameters to specify enables the identification of an arbitrary number of clusters of arbitrary shape [KKSZ11]. The estimation of an appropriate "cut" threshold is, however, non-trivial.

Generally, in biological visualization platforms, clustering can be a useful tool to organize and simplify the produced visualizations. First, (hierarchical) clustering is a common choice to determine the order of rows and columns of matrix representations of graphs [BBHR+16, NKHC08, Lii10, XGH15], which ensures that adjacent elements are more similar than those further from one another, thereby highlighting group associations of nodes. Second, clustering can also be useful to simplify or improve the visualization of a graph. For example, Angori et al. [ADM+19, ADM+22] clustered nodes into separate radial graphs to make relationships within and between clusters clear, as well as minimize edge crossings to make the produced visualization more readable. Similarly, Lambert et al. [LDB11] made use of edge-bundling and clustering of nodes-clustering to "de-clutter" visualizations of metabolic networks. However, clustering is also a useful tool to provide guidance to users. For example, Herandez de Diego et al. [HdDTMM+18] cluster metabolic pathways based on pathway profile similarity, which allows for the coloring of pathway nodes in order to assist users in identifying and comparing similar pathways. Alternatively, consider Lex et al. [LSKS10]'s tool *Caleydo*, which features several different clustering techniques for pre-filtering and highlighting gene expression data.

2.6 Network Visualization

Abstract graph drawing algorithms form the core of network visualization methods, whether they deal with biological networks [BBS14] or networks in other domains. A graph drawing algorithm takes as input a graph $G = (V, E)$ (Section 2.4), potentially enriched

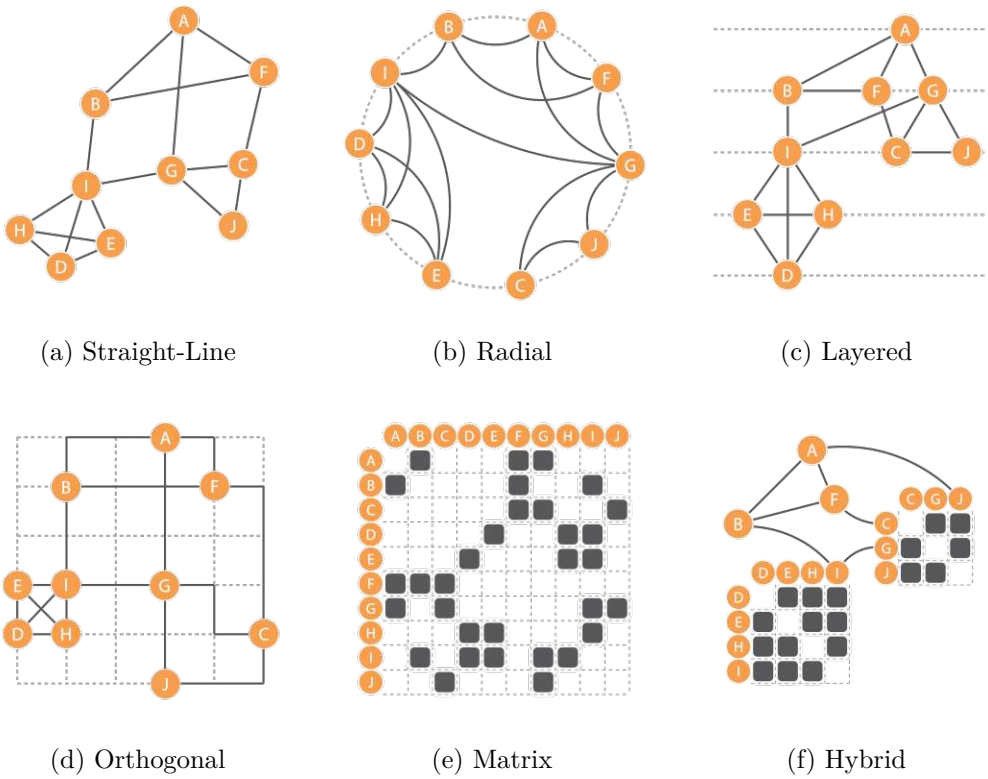
with multivariate vertex and edge attributes or having special structural properties such as being *k-partite*, *clustered*, or *multilayered*. The algorithm then computes a geometric representation of the graph, which, in most cases, maps each vertex $v \in V$ to a point $p_v = (x_v, y_v)$ in the plane (or to a larger vertex symbol such as a disk or box) and each edge $e = (u, v) \in E$ to a curve (or link) connecting the two endpoints p_u and p_v. Such drawings are also known as node-link diagrams, and they come in many different variations depending on optional constraints on the vertex positions and the edge shapes, as well as the chosen quality criteria (also known as aesthetic criteria) to be optimized.

In this section, we discuss and give examples of the most prominent graph layout styles with their corresponding positional and shape constraints, i.e., *straight-line node-link diagrams*, *radial node-link diagrams*, *layered node-link diagrams*, *schematic graph drawings*, *adjacency matrix representations*, and *hybrid graph representations*, as well as commonly applied quality criteria. For more details on graph drawing algorithms, we refer to dedicated books [BETT98, Tam14] and surveys [vLKS+11].

2.6.1 Straight-line node-link diagrams

One of the least constrained and most popular styles is the straight-line node-link diagram [WZC+15, KWKJ19, OSA+21], where vertices can be placed anywhere in the plane and edges are drawn as straight-line segments, see Figure 2.5a. Such graph drawings are usually driven by the idea that adjacent vertices are related and should be close to each other, whereas vertices not connected by an edge should be sufficiently far apart. Additionally, the drawings should have generally few edge crossings, uniform edge length (or proportional to an edge weight parameter), and balanced vertex distribution. Algorithms computing drawings in this layout style often use physical analogies like a system of attractive and repulsive forces [FR91, Kob14], in which we search for a low-energy configuration or the definition of a stress function to be minimized [GKN04]. Typically, such graph drawing algorithms group the vertices of densely connected subgraphs as spatial clusters, but if the graphs get too dense, this may deteriorate and produce so-called hairball drawings with high visual clutter. Various algorithmic approaches have been proposed to improve the computational scalability for large graphs by approximating forces and stress terms while maintaining the general layout quality [OKB17, KRM+17, Hu06].

In recent years, machine learning models have been proposed and trained to generate graph layouts that are visually comparable to typical force and stress-based drawings [KKK17, WJW+20, GMVW24, TCG24]. Machine learning has also been used to compute vertices that can be duplicated in a graph layout in order to reduce the number of edge crossings, e.g., for biological pathway visualization [NOM+21]. Graph layout with the help of machine learning techniques is an area that is still in its infancy and requires more research to understand when it is helpful and when it is not. Force-based graph drawings also form the basis for map-based network visualization, such as GMap [GHK10], visualizing clusters in the graph as countries in a fictional map. The strength of such general-purpose straight-line node-link diagrams is that force and stress-based algorithms and their objectives are intuitive and easy to use, any graph can be drawn, and the user can quickly



(a) Straight-Line (b) Radial (c) Layered

(d) Orthogonal (e) Matrix (f) Hybrid

Figure 2.5: Six typical layout styles of graph drawings are common in both general and biological network visualization, i.e., a) *straight-line node-link diagrams*, b) *radial node-link diagrams*, c) *layered node-link diagrams*, d) *orthogonal (schematic) graph drawings*, e) *adjacency matrix representations*, and f) *hybird graph representations*. All six representations utilize the same graph $G = (V, E)$, where $V = \{A, B, C, D, E, F, G, H, I, J\}$, with the same $|E| = 16$ undirected edges between them.

get a first visual impression of the data. A main disadvantage of unconstrained graph drawings is that there are very few formal guarantees on the resulting drawing quality and its geometric properties. These drawings hence bear the risk that the resulting drawings appear cluttered and hard to understand, in which case more sophisticated analysis and drawing methods need to be applied. Especially for larger graphs, such general-purpose, force-based layout algorithms tend to struggle to produce aesthetically pleasing results [KPS14], owing to the many local minima present in the underlying physical model [Kob14].

Force-directed straight-line node-link diagrams are one of the most popular approaches to network visualization across biochemical domains (Figure 2.2), likely owing to their broad applicability, computational scalability, and intuitive interpretability [KPS14]: from *gene-regulatory networks* [VGN14, CZSX20], through *protein interaction networks* [DMF17] (Figure 2.13) and *metabolic pathways* [OSA+21], to *multi-omics networks*

[KWKJ19]. Likely owing to their poor visual scalability with increasing numbers of edges and nodes [KPS14], such straight-line node-link diagrams often form only one of multiple simultaneous views of the data. For instance, within the context of general systems biological network visualization, both *VizEpis* [KLKP15] and *GeRNET* [DGC+17] (Figure 2.11) feature *radial* as well as *schematic* representations in addition to their force-based straight-line ones. To better explore the results of mass spectrometry, *xiNET* [CFR15] allows domain experts to explore their data in an aggregated node-link diagram, a clustered and non-clustered *radial layout*, a biochemically meaningful *schematic representation*, and a *hybrid layout*. In general, straight-line node-link diagrams are a useful tool, but one that must be used carefully, or in addition to other visualization approaches.

2.6.2 Radial graph drawings

A more constrained layout style is the radial (or circular) drawing style [ZSG+15, KLKP15], which restricts all vertex positions to a given circle, see Figure 2.5b. The edges of the graph are then drawn within the circle as either straight lines or smooth arcs, whereas the outside of the circle is usually not used for the drawing itself, but rather for augmenting the drawing with additional information such as names, vertex attributes, statistical plots etc. In such a radial drawing, one usually aims for a uniform distribution of vertices along the circle and a vertex order that induces a small number of edge crossings in the inside of the circle [BB05]. In some settings, vertices may have different sizes and can also be grouped or ordered by external attributes or an additional clustering hierarchy. Radial drawings are visually appealing, with a popular example of biological networks being the *Circos* system [KSB+09]. While radial drawings work well for small to medium-sized and not-too-dense networks, the restricted space for vertices and edges can become a challenge for large and dense graphs.

Radial representations appear to find frequent use when entity nodes are to be visualized in (disjoint) groups. For example, one of the views provided by *xiNET* [CFR15], is a radial graph drawing that places nodes (residues) dependent on their mapping to different proteins. Two such drawings are offered; one which places all nodes equidistantly along the circle's perimeter, the other which places them proportionally to their positions in each protein's residue sequence. Similarly, *Pathrings* [ZSG+15] arranges genes on a circle's perimeter to make their mapping to Reactome's hierarchical pathway taxonomy clear. However, such grouping can also be communicated by placing nodes on different circles, each with a different radius. For example, *miRNet* [CZSX20, FX18] provides a concentric option in which miRNA fragments are grouped into, for example, functional modules [TRZ18], each of which has its fragments arranged along a different (semi-) circle of varying radii. Additionally, instead of rendering multiple circles of differing radii, each group's subgraph could be drawn as a circle and stacked in a 3D representation, as can be seen in *Mango*'s so-called crown plots [CCC16] or *GeneNetVR*'s time-series view [FBP21]. Beyond representing groups and their entities, radial drawings are often featured as an additional view into (a fragment of) the graph, as it can offer a more organized view into a network [KLKP15, DGC+17]. However, as circular drawings scale poorly with increasing

numbers of vertices and edges [ALMAN18], edge bundling is frequently also employed in order to provide less cluttered drawings and also make group-level connectivity clearer.

2.6.3 Layered graph drawings

For hierarchical and directed graphs, which include in particular all trees, the layered layout style is frequently used. Such a drawing is composed of multiple, usually horizontal layers, onto which the vertices are distributed, see Figure 2.5c. The drawing aims to have all edges pointing upward, i.e., for each edge, its source vertex should be on a layer below its target vertex. This is only possible if the graph has no directed cycles; otherwise, the number of downward-pointing edges is minimized. For directed networks, this means that all or most of the directed relationships or dependencies can be read by following the upward direction, where edge directions are also indicated by arrowheads or similar visual cues [BCD+16]. Further optimization goals include crossing minimization and straightness of edges spanning across more than two layers, as well as minimizing the number of layers and limiting the number of vertices per layer. The main degrees of freedom for such layered drawings include the vertex-to-layer distribution and the ordering of vertices on the same layer. The popular Sugiyama framework [STT81] for layered graph drawing algorithms decomposes the problem into a pipelined sequence of individual layout steps. Most of the corresponding computational subproblems are NP-hard, but several good heuristics and exact algorithms are known [JM97, HN14]. If the graph to be drawn is actually a tree, specific tree drawing algorithms can be applied [Sch11], not all of them computing layered layouts in a strict sense.

Layered graph drawings are less frequently employed and find the most application in visualizing graphs with an underlying tree structure (Figure 2.2). The most common application area of such drawings is the visualization of *phylogenetic trees*. In such applications, each vertex commonly represents a taxonomic classification or organism, and a (directed) edge between them an ancestral relationship [LB21]. Commonly, these vertex layers are arranged horizontally and encode either simply the depth within the tree, seen in *MEGAN* [HBF+16] or *Phylo.io* [RDD16], or, for example, the time domain, as seen in *NextStrain* [HMB+18]. However, these layers need not be arranged horizontally or vertically only. Tools such as the *LifeMap* [Vie16] arrange the tree's layers in a (pseudo)radial fashion in order to make more complete use of the screen space. The *Tree of Life* [LB21] uses this additional space to visualize the network's metadata.

2.6.4 Schematic graph drawings

For more complex networks, where a simple straight-line node-link diagram might be insufficient and too cluttered and radial or layered drawings are not suitable, schematized network layouts placing vertices on 2D grid positions and routing edges as polylines with a restricted set of slopes, e.g., horizontal and vertical, or additionally using the two main diagonals, can be applied [EFK01], see Figure 2.5d for an example. Such layouts are reminiscent of electrical wiring diagrams, grid city maps, or public transit maps.

Typical optimization criteria in schematic drawings include achieving compact grid sizes, minimizing the number of bends, and minimizing the number of crossings. In order to deal with dense and non-planar networks, additional techniques like vertex duplication or edge bundling can be used to reduce visual clutter caused by edge crossings [WNSV19]. Moreover, vertices with more than four incident edges need to be represented as boxes with multiple ports on each side rather than points, since a point on a grid has only four outgoing grid lines. Orthogonal and schematic drawings can also be combined with a layered approach for directed graphs [SSvH14].

Together with *straight-line node-link diagrams*, schematic graph drawings form one of the most common visual representations of biological networks (Figure 2.2). Biochemical and biomedical domain experts are familiar with very specific (often hand-drawn) layouts of (sub)graphs, and thus many tools aim to reproduce, as well as augment, graphs in that schematic embedding that is familiar to them. For example, *Entourage* [LPK+13] aims to produce augmented drawings identical to those found in the *KEGG: Kyoto Encyclopedia of Genes and Genomes* [KG00] in order to allow domain experts to better explore and analyze their data. Similarly, Blucher et al. [BMSW19] provided a tool to allow domain experts to explore the relationship of biological entities within the context of Reactome's pathways and their visual representations. Beyond reproducing existing drawings of networks, schematic representations also create novel drawings using a visual language familiar to the user. For example, within the context of *protein interaction networks*, *xiNET* [CFR15] represents a protein's sequence of peptides as a linear arrangement of rectangles. Differently colored edges connect both peptides as well as entire proteins to communicate different functional relationships within and between proteins. Alternatively, within the context of *metabolic pathways*, Livingi et al. [LOP+18] represent different types of biological entities, their location within the cellular system's compartments, and various types of relationships between them, using a Petri-Net-styled drawing. Other tools, however, utilize automated schematization to better communicate relationships between entities by rendering data in a non-domain-specific style familiar to the user; most commonly inspired by transportation or metro networks. Consider, for example, *NeuroLines* [AABS+14], which represents the brain's 3D topology as a subway map. Similarly, Beyer et al.'s *Sequence Tube Maps* [BNH+19] (Figure 2.19) also make use of a visual metro-map metaphor, albeit to communicate genome sequencing reads more intuitively. Lastly, *Metabopolis* [WNSV19] (Figure 2.14), again inspired by urban maps, represents metabolic pathways as "city blocks", i.e., rectangular areas, and their entities' relationships as an octolinear set of nodes and vertices forming a "grid-like road". In summary, schematic representations, while more complex to draw, provide the user with an intuitive visual representation of complex biological data.

2.6.5 Matrix representations

A very different way to show networks is to use a matrix representation [ZJVS19] rather than drawing the network as a node-link diagram. The adjacency matrix of an n-vertex graph G is an $n \times n$ square Boolean matrix with one row and column for each vertex.

The cell at row i and column j gets the value 1 if the edge (i, j) is part of G; otherwise, it is set to 0. For undirected graphs, both edges (i, j) and (j, i) are represented, turning them into a symmetric matrix. When visualizing an adjacency matrix, cells with value 1 are encoded as colored pixels, see Figure 2.5e. By re-ordering rows and columns, matrix representations can be obtained that group the vertices of densely connected subgraphs and show them as submatrices of blocks of pixels with only few gaps so that high-level patterns of the graph topology become visible. Typical optimization criteria for the row and column ordering include the highlighting of certain structural patterns, e.g., by defining similarity scores on row/column vectors and grouping similar rows/columns [Lii10, BBHR+16]. For certain network reading tasks (e.g., identifying adjacencies, high-level graph comparisons), especially on large or dense graphs, matrices have advantages over node-link diagrams [GFC04], which are usually better at more complex pathfinding tasks.

Given this seeming superiority of matrix representation for certain graph analysis tasks, it is surprising that they are fairly uncommon (Figure 2.2). A possible explanation lies in domain experts' preference for the canonized *schematic* or *straight-line node-link* drawings of their networks; though examples of matrix representations offered alongside [ZJVS19] or instead of [DMF15] (Figure 2.8) such drawings do exist. Instead, matrix representations are frequently utilized to provide a view into the experimental data itself; most commonly in visual gene (co-)expression analysis tools. In addition to these canonical graph representations, tools such as *Caleydo* [LSKS10], *VizEpis* [KLKP15], or *3Omics* [KTT13], provide the experimental-data-derived correlation or co-expression network displayed as a (clustered) heatmap [ZJVS19]; though, similar correlation network visualizations can be found in *genomic variation graph* [BYCH+12] and *brain network* [FCM+17] applications. However, matrix representations have been used specifically to facilitate the comparison of (sub)graphs across conditions. For example, Bach et al.'s *Small MultiPiles* [BHRD+15] offers the ability to browse a "flipbook" of matrix representations to investigate dynamic networks at different time points. Additionally, within the context of gene co-expression analysis, New et al.'s framework [NKHC08] (Figure 2.9) allows for the identification and comparison of multiple subgraphs in side-by-side triangular matrix heatmap representations.

2.6.6 Hybrid graph representations

Finally, the above-mentioned fundamental layout styles have also been augmented and merged into hybrid representations that aim to combine the strengths of two different visualization styles. Two prominent examples of hybrid approaches are *NodeTrix* [HFM07] and *ChordLink* [ADM+22]. The *NodeTrix* idea is a hybrid of node-link diagrams and matrix representations, originally proposed for social network visualization. It uses pixel matrices to represent dense subgraphs, where node-link diagrams would produce too much clutter. Each matrix itself can be seen as an aggregated vertex in a sparser high-level graph, which is displayed as a node-link diagram and thus has an advantage in showing topological connectivity properties, see Figure 2.5f. Edges can either link whole matrices

in an aggregated sense, or they may link individual vertices via the rows and columns of different matrices. *ChordLink* follows a similar idea as *NodeTrix*, but it represents the dense subgraphs of clusters as radial layouts (here called chord diagrams), which are in turn connected in a node-link diagram that shows the global network structure. In the chord diagrams, vertex replication and ordering schemes are used to reduce crossings and improve the visual representation. Both systems provide an interactive interface to define clusters and highlight areas of interest during exploration and analysis.

As these hybrid visualization approaches demand higher implementation efforts, they are used less frequently than any other graph representation (Figure 2.2). However, Henry et al.'s *NodeTrix* [HFM07] representation has found use in the visual comparison of brain networks, as *NodeTrix* allows for a simultaneous coarse and fine-grained view into the global and local topology of the graph, respectively [YSD+17]. Similarly, *ChordLink* [ADM+19, ADM+22] has also found adoption in biological network visualization [MYH+19, XGH15], as it allows for an uncluttered view into the connectivity between groups of vertices to identify broader trends. Lastly, *Caleydo* [LSKS10] provides a 3D view linking multiple *schematic graph representations* with *straight-line node-links* between their nodes.

2.6.7 Discussion

With the different types of network visualizations discussed above (recall Figure 2.5), a key question to a domain expert wanting to visualize their biological network data is to select the most suitable of these visualization styles. This decision depends on multiple aspects of the data, in particular, (i) whether the graph is sparse or dense, (ii) whether it is relatively small (less than 100 vertices) or large, (iii) whether edges are directed or undirected, or (iv) whether clusters in the data should be emphasized. Moreover, certain structural properties in the data are relevant, for instance, the maximum degree of a vertex, or whether the underlying graph belongs to a particular graph class, such as planar graphs or trees, or whether it has no immediate or known structural properties. As a rule of thumb, exploring a new network data set using the force-based straight-line drawings first is usually a good idea. These algorithms do not require any special structural properties of the graph and use the intuitive straight-line node-link diagram style. The mechanisms behind force-based algorithms also group clusters of densely connected vertices and aim to distribute unrelated vertices evenly in space. Finally, force-based layout algorithms are readily available in most network visualization tools and do not require expert knowledge or adaptations of specialized libraries. The many biological network visualization examples listed in Section 2.6.1, which are following this approach, underline their frequent use in the domain.

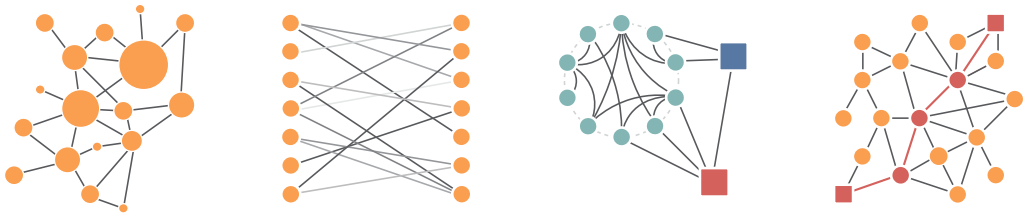
The main disadvantage of force-directed layouts is the lack of quality guarantees coming with a risk of visual clutter and overplotting of features. Resorting to other algorithms and layout styles may improve readability of certain aspects of the data. For instance, layered drawings or schematic drawings have specific constraints that avoid overplotting of vertices and generally create a more orderly appearance of the graph, usually at the

expense of longer or non-straight edges. Likewise, if the network turns out to be tree-like/hierarchical or has generally low vertex degrees, then layered and schematic layouts could also be good options. Lastly, radial layouts are also a way of placing all vertices in a well-structured manner, i.e., in a radial fashion, without requiring specific graph structures. Radial layouts avoid giving some vertices a more central or hierarchically higher position than others, a property that may be undesirable in certain scenarios. Such a constrained placement of the vertices, however, often induces higher number of edge crossings, such that interactive highlighting of edges might become necessary in order to clearly indicate precise connectivity information. Lastly, matrices offer a different visual representation of a network, which can offer insights into the data that are less apparent in node-link diagrams. As there is, by definition, no notion of clutter caused by edge crossings in matrices, they do scale well to dense networks. For instance, densely connected clusters as well as high-degree vertices with many neighbors can be recognized well via visual patterns in a suitably ordered matrix visualization [BBHR+16]. Finally, if several layout styles are of interest for the specific network data at hand, these can be combined either using multiple linked views or in a hybrid style such as those discussed in Section 2.6.6. This, however, usually comes at the cost of requiring significantly more implementation and adaptation effort than using general-purpose layout algorithms.

2.7 Graph Analysis Tasks

In network visualization, as in data visualization in general, the underlying data, and by extension, the associated domain or research question, and consequently the chosen visual representation, dictate what kind of analysis tasks are to be performed. Several taxonomies and typologies of varying degrees of specificity have been put forth over the years. There are general taxonomies applicable to most visualizations [Mun14, Shn96, AES05] and ones tailored to general network analysis [LPP+06b]. Focusing on particular types of networks, there are even more specialized taxonomies detailing, for example, temporal network evolution [APS14, KKC15], networks of overlapping sets [RSA+16], and biological pathways [MMF17].

In this work, we scope the tasks used in the analyzed tools using a taxonomy adapted from Lee et al. [LPP+06b]. More specifically, we select this particular taxonomy as it offers, in our estimation, the right level of abstraction for general biological network analysis and visualization. More general taxonomies [Mun14], on the one hand, are not specific enough to the objective of network visualization. On the other hand, more specific network task taxonomies [APS14] are too specific to their particular sub-domains for a scoping analysis such as this and subsequently do not feature sufficient overlap with the objectives of the papers analyzed here. If applicable, however, the taxonomy by Murray et al. [MMF17] is mentioned as well, as the application domain is of high relevance.



(a) Focus: Nodes (b) Focus: Edges (c) Focus: Groups (d) Focus: Paths

Figure 2.6: Illustrative examples of Lee et al.'s [LRK18] four graph task's objects focus: a) the nodes, b) the edges, c) the groups, and d) the paths. A) displays some hypothetical quantity mapped to the nodes' surface area to guide users to potential nodes of interest. Similarly, b) showcases some conceptual quantity mapped to the opacity of each edge to guide users to potential edges of interest. Similar to the discussed tool *iTol* [LB21], c) shows a group-focused visualization in which each group has been visualized in a different color (**red**, **blue**, and **turquoise**), and one of said groups has been expanded into its constituent nodes. Lastly, d) showcases a path in **red** between two user-selected nodes (shown as squares).

2.7.1 Object Focus of Tasks

Lee et al. [LPP+06b] taxonomy describes the notion of focal objects for the definition of tasks. A focal point describes a central element of the network visualization that plays the most integral role in a given task. We distinguish between four categories of focal objects.

The first, arguably most common, focus is the network's ***nodes*** (Figure 2.6a). Many biological network visualizations feature nodes as their focal object. For example, many applications featuring *simple straight-line node-link diagrams*, such as *STRING* [SGL+19], utilize nodes to display proteins in a *simple straight-line node-link diagrams*, 2.9.1 in which proteins and their associated interaction partners form the focus of their visualization. But nodes as a focus are also featured in other network visualization types like trees. Both, *MEGAN* [HBF+16] and *NeuroLines* [AABS+14] use nodes as focal points. *MEGAN* maps read data to taxonomic or functional classification categories and display these as nodes in a tree. *NeuroLines*, on the other hand, models connections of neurites, i.e., axons and dendrites, as trees with the nodes representing specific synapses.

Equally important in networks are its ***edges*** (Figure 2.6b). Similarly to nodes, edges are frequently used as focal objects in network visualization. Sometimes, visualizations featuring a node focus will also feature edges as focal objects. For example, the already discussed STRING [SGL+19] features a node-focus and an edge-focus. Edges are not only simple connections between nodes but also communicate the level of evidence present for the particular protein-protein interaction the edge represents. Additional applications exist in which edges take precedence over nodes. GeRNet [DGC+17] — an application for

the visualization of gene-regulatory networks — for example, features nodes in addition to edges. They, however, only indicate the object of interaction while the associated edges define the type of interaction in detail. Some visualizations do not even render nodes at all but are focused solely on the connections between them. *NodeTrix* representations, for example, utilize adjacency matrices—encoding edges between nodes—and links connecting those matrices to visualize, for example, brain connectivity networks [YSD+17].

Besides the two basic organizational units of networks, i.e., nodes and edges, other objects can also form a visualization's focus. Applications that aggregate or group similar objects, ***groups***, clusters, and connected components are often the key quantity of interest (Figure 2.6c). *iTOL* [LB21], for example, visualizes phylogenetic trees, displaying different species and their relatedness. While each species is visualized as a node, iTOL defines groups of species, so-called clades, which form an equally important focus and can be interactively collapsed or expanded. For other applications, such as Cruz et al. [CPAM18]'s visualization of RNA-seq clusterings over time, other graph objects are only of secondary importance, and it is the groups, i.e., the RNA-clusters, that form the visualization's object focus.

Finally, sequences of connections, i.e., ***paths***, can also be the focus of an analysis task (Figure 2.6d). This focus type is common in metabolic pathway visualizations, as these aim to visualize the metabolic events inside an organism as a series of reactions, shown as edges between nodes. With *Entourage*, Lex et al. [LPK+13], for example, show important paths in different related pathways based on experimental data. Metabopolis [WNSV19] combines the group-focus with the path-focus to visualize metabolic paths through semantically grouped blocks of metabolic reactions. Besides metabolic pathways, genome graphs are another kind of data visualization with a heavy focus on paths. Visualizing multiple genome sequences as aligned paths, Sequence Tube Maps [BNH+19], for example, displays identical regions as nodes and variations in sequence as diverging paths.

2.7.2 Complex Task Focus

Besides the fundamental objects that are the subject of a given analysis task, Lee et al. [LRK18] further categorized graph analysis tasks into groups of complex tasks.

Attribute Based Tasks

The first category describes ***attribute-based tasks***. Tasks are attribute-based if they are related to attributes of the contained network primitives, most commonly nodes and edges. Tasks concerning operating on attribute data encoded by nodes are categorized as **on the nodes**. Murray et al. [MMF17] also similarly describe *Attribute tasks* for biological networks. *PaintOmics* [HdDTMM+18], for example, colors nodes in metabolic pathways depending on the underlying measurement values of the corresponding entities. This allows users to access these measurement attributes and answer questions such as "Which nodes possess high measurement values?" or "Which nodes possess similar measurement values?". Another common task in this category is the search for nodes with

a specific categorical attribute, often facilitated by assigning categorical colors to nodes. In *OmicsNet* [ZX18], the different biological entities, e.g., transcription factors, proteins, and miRNAs, are given a unique color to allow the user to recognize the type of a given node more quickly. Analogous to attribute tasks on the nodes, there are also attribute tasks ***on the edges***. Attribute data encoded on the edges is the main focus of the hypothetical task. *BioLinker* [DMF17], on the one hand, uses categorical color coding to communicate types of interactions in a protein-protein interaction network. *STRING* [SGL+19], on the other hand, utilizes its color coding to describe evidence types for its protein-protein interactions. Beyond color, tree visualizations like *Nextrstrain* [HMB+18] often use a length encoding to encode the passage of time between measurements or entities. In the case of *Nextstrain* specifically, this allows users to quickly gauge the length of time a particular pathogen has been active.

Topology-Based Tasks

In addition to tasks based on attributes of nodes and edges, tasks can be directed at ***topological*** features of the graph. Such tasks can focus on the graph as a whole, or concern themselves with particular features or subgraphs in more detail.

Fulcrums are articulation points in graphs connecting two components of a graph, and the removal of such a fulcrum would (most often) lead to the generation of two disconnected components. Thus **finding fulcrums** can be an important task as their removal can strongly impact a graph's topology. In *NAP* [TEP+17], for example, fulcrum nodes are identified based on their centrality. While standalone applications to find fulcrums are rare, at least in our corpus of collected literature, more complex frameworks, such as *VANTED* [HJ18] or *Omix* [DNW13] can generate visualizations aimed at finding fulcrums in biological network visualizations.

A common task is ***finding groups***. Groups can play an important role in biological networks, as finding similar entities or grouping entities semantically can help reduce the number of organization units to be analyzed or visualized. Murray et al. [MMF17] define the *Grouping* task as a *Relationship* task. There are several examples of applications, such as *ClusterViz* [WZC+15], that are tailor-made for the task of finding clusters in a biological network, offering various cluster algorithms and side-by-side views of the found clusters with additional statistics. Similarly, the visualization Cruz et al. [CPAM18] focuses on the temporal evolution of clusters in networks. A task that is also reflected in the specialized taxonomies of Ahn et al. [APS14] and Kerracher et al. [KKC15], e.g., the analysis of *Growth and Contraction* of groups or as a *Q4* task, respectively. With *MultiPiles* [BHRD+15] Bach et al. aggregate matrices by similarity to a customizable degree, allowing interactive investigation of the effect of abstracting the data.

Determining the adjacency of nodes, i.e., ***finding neighbors*** of specific nodes, is an equally common task. A prime example of this task is in protein-protein-interaction networks, where the primary objective is to find neighboring nodes, i.e., proteins, that interact. This can be seen, for example, in STRING [SGL+19] or Biolinker [DMF17].

Fining neighbors is also relevant for gene-regulatory networks [PD17] or heterogeneous regulation networks [ZX18, CCC16], in which regulation is indicated by being connected by an edge, analogous to protein-protein-interaction networks. Similarly to the finding of groups, determining neighbors often features in addition to other tasks, where it either plays a secondary role or features alongside other tasks.

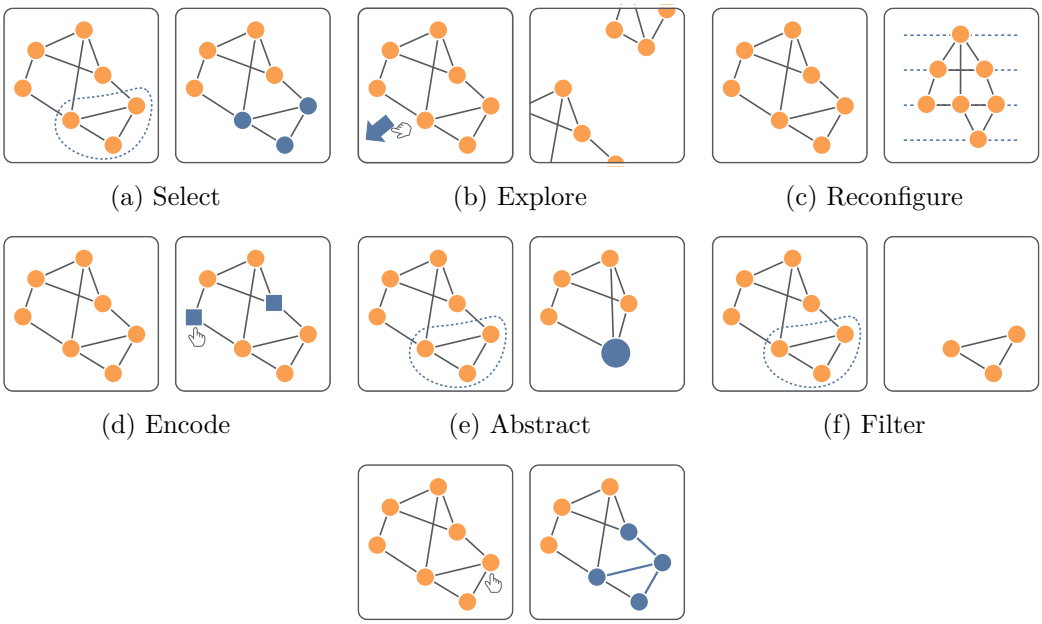
Lastly, we distinguished the task of **finding paths**. As described in subsection 2.7.2, paths are a series of edges connecting two nodes by traversing at least one additional node. In practice, most applications with an object focus on paths also aim at finding paths in a given network. And just as in the path-object focus, finding paths is a particularly common task in metabolic pathways [WNSV19, LPK+13, HJ18] and genome graphs [GHN+21, BNH+19, WSZH15]. Thus, it is not surprising that a multitude of tasks, in more detail than just finding paths, are featured in the taxonomy by Murray et al. [MMF17]. Specifically, tasks about the *Direction* of paths, the *Causality* of changes downstream in a path, or even if paths form *Feedback* loops. However, some protein-protein-interaction graph visualizations like *BioLinker* [DMF17] also feature pathfinding, to locate interactions between two proteins.

Other Complex Tasks

There are, of course, other complex tasks that are neither exclusively based on attributes nor topology. Here, we describe these browsing, overview, and graph comparison tasks.

As the primary browsing tasks, Lee et al. [LRK18] mention the **following of paths** in given networks. For visualizations that feature pathfinding tasks or a path object focus following a path is a very intuitive task.

The **overview task** is quite a broad task category, encompassing everything that does not focus on a single node, edge, or cluster but instead the graph as a whole. Given this wide-ranging definition and the general need for visualizations to represent complex processes, many of the investigated papers feature some kind of overview element (Figure 2.2). The majority of tools opt to visualize networks using some form of node-link diagram (Figure 2.2). Thus, many characteristics and properties, like connected components or hubs, can be identified purely visually. Often, cluster analysis and the additional visualization of its results further amplify this effect. In *miRNet2.0* [CZSX20], different layouts in combination with color are used to highlight different aspects of the network to investigate the effect and regulation of miRNAs. There are however also works that explicitly focus on giving an overview, with *ODGI* [GHN+21] that aims to show the genomic variation in pangenome-graphs, i.e., how genomes differ from each other on a genome level, at an overview stage allowing to identify regions in which there is much (or little) genomic variation. *PaintOmics* [HdDTMM+18], on the other hand, uses an abstracted but otherwise conventional node-link diagram in which each pathway is represented by a single node in combination with node colorings to give an overview of a set of pathways. Another example is the visualization by Cruz et al. [CPAM18] which



(a) Select (b) Explore (c) Reconfigure

(d) Encode (e) Abstract (f) Filter

(g) Connect

Figure 2.7: Human-Computer Interactions in Biological Network Visualization. (a)-(g) Examples for seven general categories of interaction techniques [YKSJ07], i.e., a) select, b) explore, c) reconfigure, d) encode, e) abstract, f) filter, and g) connect. All hypothetical user interactions and changes are shown in **blue**.

shows the temporal evolution of clusters in a network in an abstracted overview time curve, indicating the stability of a cluster over time.

A more specialized task is the one of **graph comparison** in which several distinct graphs are compared. As the comparison of metabolic pathways is of high interest in the domain, Murray et al. [MMF17] also describe *Comparison* task, based on *Attributes* of the graph entities. One such example is Phylo.io [RDD16], which was created explicitly to compare different phylogenetic trees using a side-by-side view. *Cerebral* [BMGK08], on the other hand, uses a combination of small multiples and coloring to allow the comparison of multiple experimental conditions and their effect on metabolic pathways.

2.8 Human-Computer Interactions

To perform the tasks outlined in Section 2.7, many visualizations require interaction techniques transforming the static visualization into non-static, interactive visualizations. Here we describe the commonly used graph interaction techniques identified in the taxonomy by Yi et al. [YKSJ07]. The specific interaction techniques are very general, and in many cases, the specifics of their implementation are strongly data and task-dependent. We thus do not explicitly map them to our taxonomy for network analytics (Figure 2.2),

but instead, point out the connections to the described tasks and mention examples for the corresponding interaction types.

2.8.1 Select

Select is one of the core and most common interaction techniques in both general and network visualization, as it helps users to keep track of data items intuitively [YKSJ07]. A very prevalent realization of this select interaction in network visualization is the simple (cursor-based) selection of vertices or edges (see Figure 2.7a). One such example can be seen in *xiNET* [CFR15], in which users can hover over as well as click proteins or their connecting links to reveal additional (attribute) information. However, such selections of items can often be used in more advanced ways, as well: *Gernet* [DGC+17] allows for clicking and dragging to select a subset of the displayed entities. Fuijiwara et al. [FCM+17] provide a lasso selection tool for the same purpose.

Alternatively, entities can be selected using so-called search masks: Commonly, users can directly select target entities, such as proteins [DMF17] or genes [PRKM+14], via their respective domain-specific accession identifiers. It is important to note that selection techniques are rarely used in isolation. Instead, they are often employed in conjunction with other interactive methods, such as *Explore* or *Abstract/Elaborate* techniques to, for example, show detailed attributes of nodes and links to better support particular *attribute-based tasks*.

2.8.2 Explore

Exploration interaction techniques enable users to navigate varying subsets of data entities, e.g., depending on the *object-focus* [YKSJ07], nodes, edges, groups, or paths. Exploration techniques are often associated with *topology-based tasks* and *attribute-based tasks*, such as finding the set of neighbors adjacent to some node of interest, or locating a set of nodes with a particular attribute value, respectively. As previously mentioned, exploration techniques are often employed in tandem with the *select interactions*. In Bio-Linker [DMF17], for example, users can request the database to load the selected protein and its neighbors of interest by entering its identifier in a search box. This, in turn, expands the current view with new entities and relationships. An additional example is Stem-Cell-Net [PRKM+14], in which users can input a list of gene identifiers. The system will then return a list of genes matching the given identifiers, from which users can select a central focal node for the network. A common standalone exploration technique in biological network visualization is *panning*, which is used to navigate the network's 2D embedding [RJH+12, MYH+19, BMGK08, JYC+10, CFR15, CPAM18] (Figure 2.7b). Generally, clicking and holding the right mouse button while dragging the cursor allows users to move the visualization canvas. This panning technique is helpful as it allows layouts to render nodes off-screen while still allowing their exploration.

2.8.3 Reconfigure

Visualization tools offer the advantage of having a *human in the loop* to tailor the visualization to the researcher's needs. This is primarily achieved through ***reconfiguration*** interactions, in which the users can change the spatial arrangement of the network's representation to obtain different views on the same data [YKSJ07]. *Pathway-Matrix* [DMF15] uses *adjacency matrix* visualization to present the binary relations between proteins in a pathway. As the ordering of rows and columns influences the perception of visual patterns, Deng et al. [DMF15] offer various metrics with which to (re-)order the matrix. This, in turn, can reveal high-level patterns in larger pathways, such as sub-networks and clusters of related proteins. For example, as shown in Figure 2.8, proteins in the same family form clusters when ordering the adjacency matrix by protein name. However, some proteins of different protein families may serve the same or similar function within a pathway. When ordering proteins by similarity, not name, such patterns become apparent and can lead to potential discoveries. Thus, the reconfiguration of matrix representations better supports *topology-based tasks*.

In node-link diagrams, on the other hand, moving vertices in the graph is a widely applied reconfigure technique; see, for example, Figure 2.7c. In addition to the aforementioned *topology-based tasks*, reconfiguration in node-link diagrams is also helpful in facilitating *browsing tasks* like *path-following*. Users can interactively drag vertices around in *Cerebral* [BMGK08], whose initial placement is algorithmically informed. This way, users can manually build a skeleton of important vertices that better match their mental model. Alternatively, in *ClusterViz* [WZC+15], users can explore clusters in biological networks in different orders by sorting them by various attributes, such as score, size, or modularity. Another reconfiguring technique is changing the algorithm underlying the automatic graph layout altogether. Chang et al. [CZSX20] offer different layout algorithms such as Force-Atlas, Fruchterman-Reingold, Circular, etc., for better exploration of miRNA-centric interaction networks. Dragging nodes or switching layout algorithms results in different graph overviews to help us with different overview tasks.

2.8.4 Encode

While the *reconfigure interaction* maintains the data encoding, changing the ***encoding*** adjusts the visual representation, offering a different view of the data (Figure 2.7d) [YKSJ07]. An example of such an *Encode* interaction is the on-demand switching of a biological network's representation from an *adjacency matrix* to a *node-link diagram*. New et al. [NKHC08] utilize this approach to communicate time-dependent changes in gene co-expression networks, as shown in Figure 2.9.

The network is first represented as a matrix [NKHC08]. Brushing selections can then be performed to abstract a selected sub-graph into a hyper-node in the node-link diagram. The resulting, simplified *Level-of-Detail* graph can show the interconnections between groups of vertices. Other encoding techniques have been used to provide alternative encoding representations of the data items in the network. For example, *Cerebral* [BMGK08]

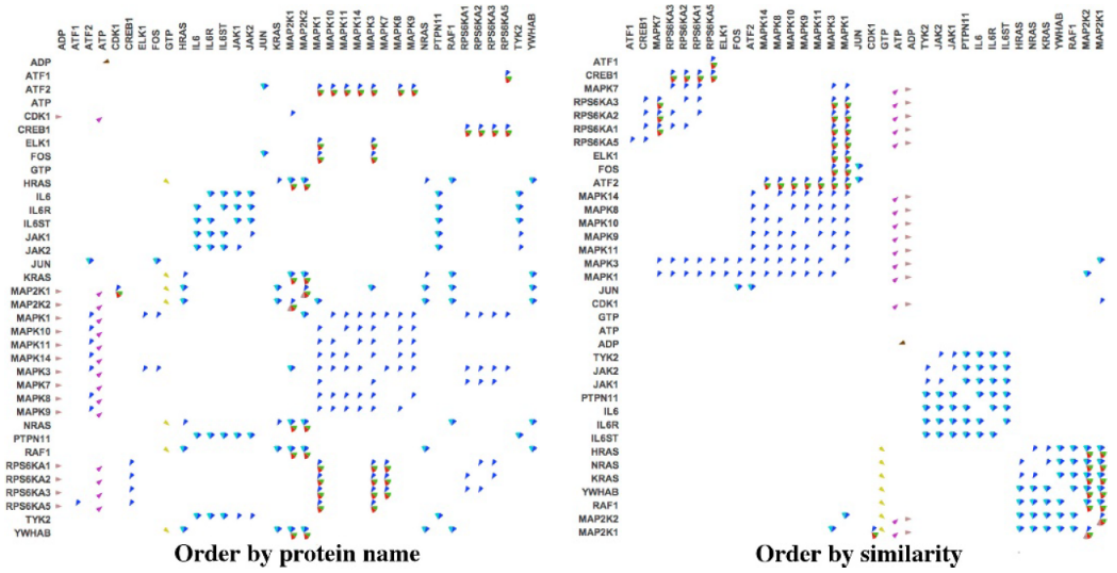


Figure 2.8: Two different protein orderings in the RAF-MAP Kinase Cascade pathway [DMF15]. Reprinted from "PathwayMatrix: visualizing binary relationships between proteins in biological pathways", by T. N. Dang et al., 2015, BMC Proceedings, S6 (9), S3. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

provides sliders for users to adjust edge curviness, label density, and group label size to enable the user to find a proper encoding scheme that best fits their research questions and tasks. *NAP* [TEP+17] allows users to choose between various colors and shapes of edges and vertices. These encoding techniques provide users with the opportunity to get a better visual representation of the whole network, to help them with their *overview tasks*.

2.8.5 Abstract/Elaborate

Another group of techniques bundled under the term ***Abstract/Elaborate*** includes a set of interaction interactions allowing the user to view representations of the input network at different levels of abstractions 2.7e [YKSJ07]. In node-link diagrams, zooming interactions are often supported to show attributes of the biological network, which are more apparent when changing the scale at which the network is displayed [RJH+12, MYH+19, BMGK08, CCC16, TEP+17, BVK+15, DGC+17]. For example, Fujiwara et al. [FCM+17] support zooming in the network view to display regions of interest more clearly by reducing clutter and node overlap. Other approaches use this approach to both declutter their visualization and to utilize the created space by the zoom operation, displaying additional information. In *Caleydo* [LSKS10], for example, zooming into the heatmap visualization displays text labels that would otherwise not be visible. Abstractions often are a cornerstone to facilitate *overview and browsing tasks*.

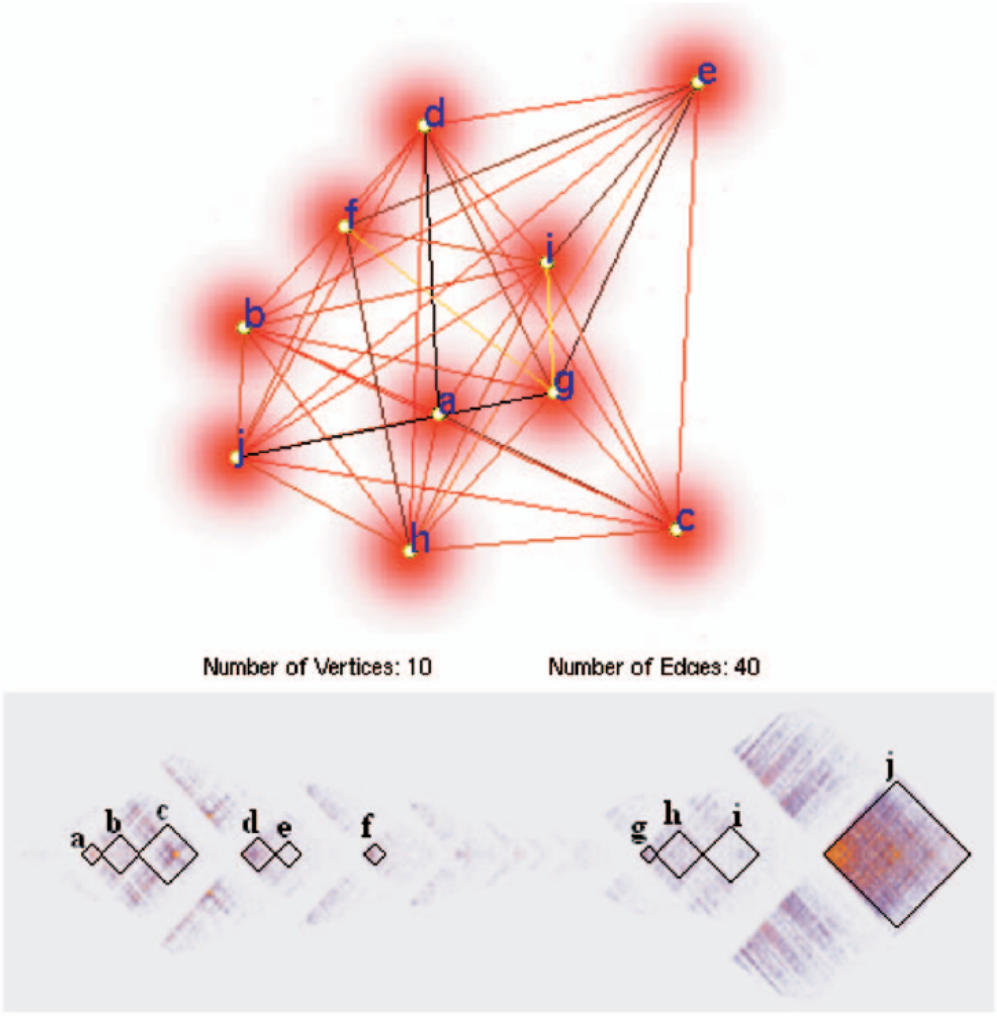


Figure 2.9: A 2D LoD graph created from brushed BTD belt selections to show correlations among BTD structures [NKHC08]. Reprinted from "Dynamic Visualization of Coexpression in Systems Genetics Data", by J. New et al., 2008, IEEE Transactions on Visualization and Computer Graphics, 14 (5), 1081-1095. Copyright 2008 by The Institute of Electrical and Electronics Engineers, Incorporated (IEEE). Reprinted with permission.

*Tool-tip*s are another commonly applied implementation of the abstract/elaborate interaction. They provide detailed information about the biological network facilitating attribute-based tasks. In most applications, tool-tips are shown after hovering over or clicking on a vertex, upon which a pop-up window with additional information is shown. Examples of vertex tool-tips show its name [CPAM18, FCM+17, RPM20], an associated gene annotation [DGC+17], or other biological information [KWKJ19, NVR14]. Other tooltips, like the ones in Bio-Linker [DMF17], show information of the hovered node and

the statistics of its immediate neighbors.

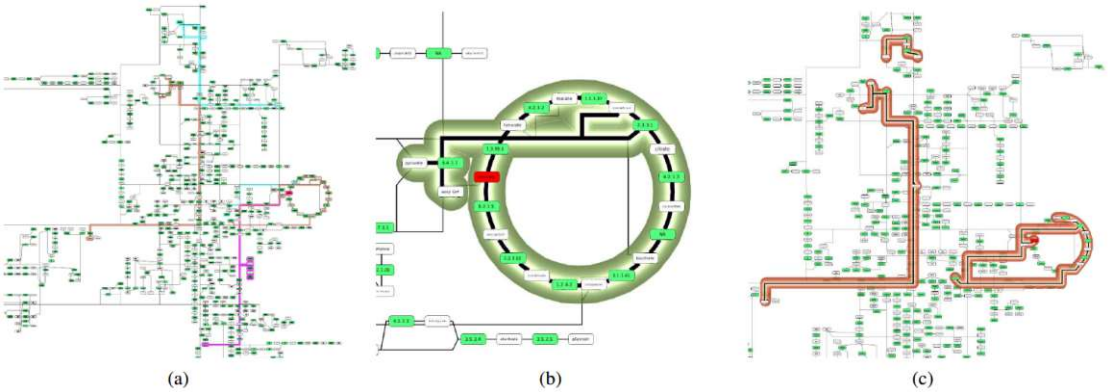


Figure 2.10: Retrieving pathway information using dedicated focus+context interaction techniques [LDB11]. Authors contacted requesting copyright for reprinting.

2.8.6 Filter

Abstraction/Elaboration techniques change how the given data is represented. In some cluttered visualizations, however, this is not desired. Instead, the user may want to remove specific items outright. This is facilitated by ***Filter*** interaction techniques. In general, a filter shows data items that meet specific conditions, for example, to vertex or edge attributes [YKSJ07], as shown in Figure 2.7f. A prime use case for this is protein-protein interaction networks, which often contain thousands of protein species and protein-protein interactions, which, unfiltered, pose a considerable challenge to be visually parsed and understood. Jianu et al. [JYC+10] allow users to remove biologically uninteresting proteins using extendable filters to simplify the visual representation of the network.

In MAPPS [RPM20], users can filter the visualization on pathway length. Other applications offer filtering with multiple options. In Stem-Cell-Net [PRKM+14], the filter options include filtering by species, interaction types, co-expressions, and evidence. Additionally, all unconnected vertices (post-filtering) are removed to avoid orphaned vertices cluttering the networks. Some applications also offer filtering based on graph topology. Zhou et al. [ZX19], for example, allow users to simplify the network by filtering less important vertices or edges based on vertex degree, *betweenness centrality*, or shortest paths. After filtering nodes and edges by their attributes, users can better complete *attribute-based tasks*.

2.8.7 Connect

Finally, we discuss ***Connect*** interactions. This interaction technique highlights the related data items or shows hidden, but contextually relevant, data items [YKSJ07]. These techniques support topology-based tasks such as finding neighbors and connecting

edges. In biological network visualization, we are often interested in related vertices and edges of some selected entity. A common connect technique, in both node-link diagrams and matrix visualization, is to show the neighbors of selected vertices (Figure 2.7g). Jianu et al. [JYC+10], for example, allow users to click a protein in the exploration view, which then highlights both the protein itself and its neighbors to establish a visual correspondence between them. In *Net-Work-Analyst* [XGH15], users can click on a vertex to zoom in to see its position and interconnectivity within the current subnetwork. This, in turn, allows users to better analyze the topology of the subgraph surrounding the selected vertex of interest. In the work of Cruz et al. [CPAM18], hovering over a vertex with the mouse highlights other connected vertices. Fujiwara et al. [FCM+17] react to hovering over a certain matrix cell with highlighting of the corresponding vertices and edges in the 3D graph view in order to provide their anatomical positions as reference. Lastly, a high-level connect technique that can find paths between selected proteins is provided in Bio-Linker [DMF17]: users can specify the source vertex, target vertex, and the maximum number of hops in between the source and target, which will then return all possible paths under the given condition to support both *topology-based tasks* and *attribute-based tasks*.

2.9 Application Areas & Existing Tools

As different biological and medical sub-domains deal with different data, they also deal with different research questions, and challenges [RCM+20]. Consider, for example, on one hand, metabolic pathways analysis which may aim to discover possible drug targets by carefully considering the directed relationships between metabolites, while also considering the possible adverse side-effects on other connected pathways [LPK+13]. On the other hand, a researcher faced with protein-protein and protein-gene interaction networks may be more interested in using the graph to understand experimental results, by considering not only the interconnectedness of entities but also their physical location in the cell [BMGK08].

In this section, we aim to provide an overview of the major biological and medical application domains and the properties of networks under study. Specifically, we focus on the following common application domains: genetic regulatory networks, protein-protein interaction networks, metabolic pathways, multi-omic networks, gene (co-)expression networks, and phylogenetic trees. However, we also briefly mention some other areas in biology where networks are relevant. Here, we provide an overview of the current, domain-specific visualization tools available and highlight their unique set of challenges and potential areas of research to guide subsequent research efforts. A detailed comparison of the tools would be difficult due to the sheer number of tools that often tackle problems in a narrow subfield. Therefore, our aim was to only give an overview of existing approaches. For some of the more feature-rich and established tools, we provide a more detailed description of their features and possible interactions. We refer our readers to the original papers for detailed explanations of the individual features, user requirements, and evaluation results.

2.9.1 Biological Interaction Networks

On the micro-molecular level, many of the processes can be described as interactions between different biological agents, such as genes, proteins, enzymes, or transcription factors. These interactions can thus be described by different interaction networks, e.g., gene regulatory or protein-protein interaction networks, and their study opens up many possibilities to form and evaluate new hypotheses.

Gene Regulatory Networks

Gene expression is one of the most central processes in biological systems. Accessing information stored in the DNA, through transcription and translation, affects nearly every biological reaction inside an organism and is thus connected to a very complex set of regulatory mechanisms. In order to model these mechanisms, gene regulatory networks (GRNs) were developed. These are a specialized type of biological interaction network, whose goal is to describe and gain insight into how gene expression is regulated. When analyzing pathological states of certain cellular processes, GRNs can thus contain helpful information about how these processes are regulated in their physiological state [KS08a]. Additionally, they can reduce the amount of experimental preliminary studies by providing computational entry points for experimental biological research [KS08a].

One example for the visualization of gene regulatory networks is *GeRNet* [DGC+17]. *GeRNet* uses the two algorithms to infer gene regulation rules and draws them in a *force directed node-link* diagram of a *simple graph* (see Figure 2.11). The vertices in this diagram constitute genes, while the edges are the inferred rules, with the type of edge being one of six regulation types. On-demand the user can manually add rules and vertices to curate the generated network (*reconfigure*). Some additional approaches concerning gene regulatory networks are more analytical, featuring time-series visualizations [PD17], while others include matrix visualizations [ZJVS19].

Protein Interaction Networks

While proteins are responsible for performing many of the central tasks in biological systems, they often do not act as an isolated entity but in conjunction with other proteins. Thus, when investigating a protein's function and role in the organism, the need of knowing which proteins might interact with the protein of interest arises. Such knowledge can be extracted from protein-protein interaction networks, which are networks representing the aforementioned interactions [NYP12, SM03]. Additionally, such networks can be analyzed further to find the more loosely defined *modules*, which are used to attribute larger-scale cellular functions [HHLM99, CY06]. The main source of information regarding protein-protein interaction is generated in the wet lab using a wide range of different techniques — a labor-intensive process. Thus *in-silico* predictions of these interactions from a variety of different sources, for example from protein structure, their phylogeny, or the corresponding gene neighborhoods, have become increasingly more important [SSBE08]. Databases like the *STRING* [SGL+19] database, thus not only keep

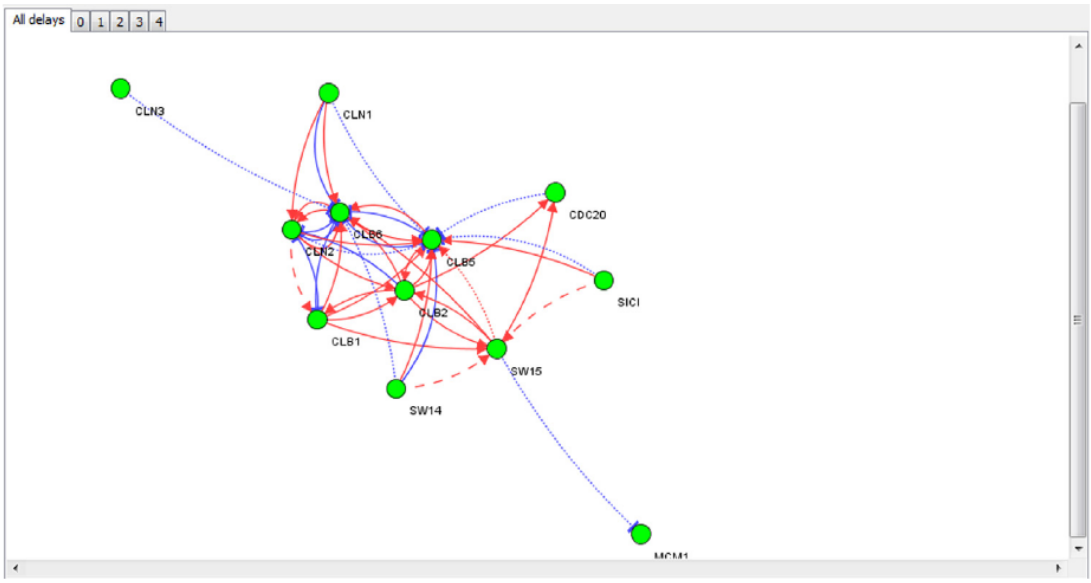


Figure 2.11: *Node-link* diagram of *GeRNet* [DGC+17]. Authors contacted requesting copyright for reprinting.

a record of experimentally shown protein-protein interactions but also record evidence levels indicating the reliability of the given interaction, i.e., if there is experimental or only predictive evidence for a given type of interaction. Producing meaningful visualizations of these dense networks containing such differences in reliability poses an additional challenge for analysis systems.

Indeed, *STRING* itself offers a system for the visualization of its contents [SGL+19]. The user can query the *STRING* database for one of the over 24.6 million proteins from over 5,090 organisms, which is then shown with its association network as a *force directed node-link* diagram of a *simple graph* (see Figure 2.12, top). The query vertex is then shown in red while the edge colors indicate different types of evidence, whose detail can be explored via click interaction on the edges (*selection*). In an interactive mode, all vertex positions can be moved (*reconfigure*). Another option is to utilize k-means or MCL *clustering* to generate a *clustered graph* in which colors signify cluster membership and inter-cluster edges can be visualized by dotted lines. Additionally, a set visualization shows the interactions of the query protein with all interaction partners and its evidence levels sorted by total score (see Figure 2.12, bottom). As the *STRING* database associates this evidence with scores and links them to the corresponding entries, these scores can then be used as a measure of confidence for a given interaction. The composition of those scores can be seen in the click interaction or in the legend accompanying the vertex link diagram. The user can expand or restrict this network by changing a score threshold, changing the number of vertices visualized in first or second-degree neighborhoods, as well as the choice of interaction sources (*filter*).

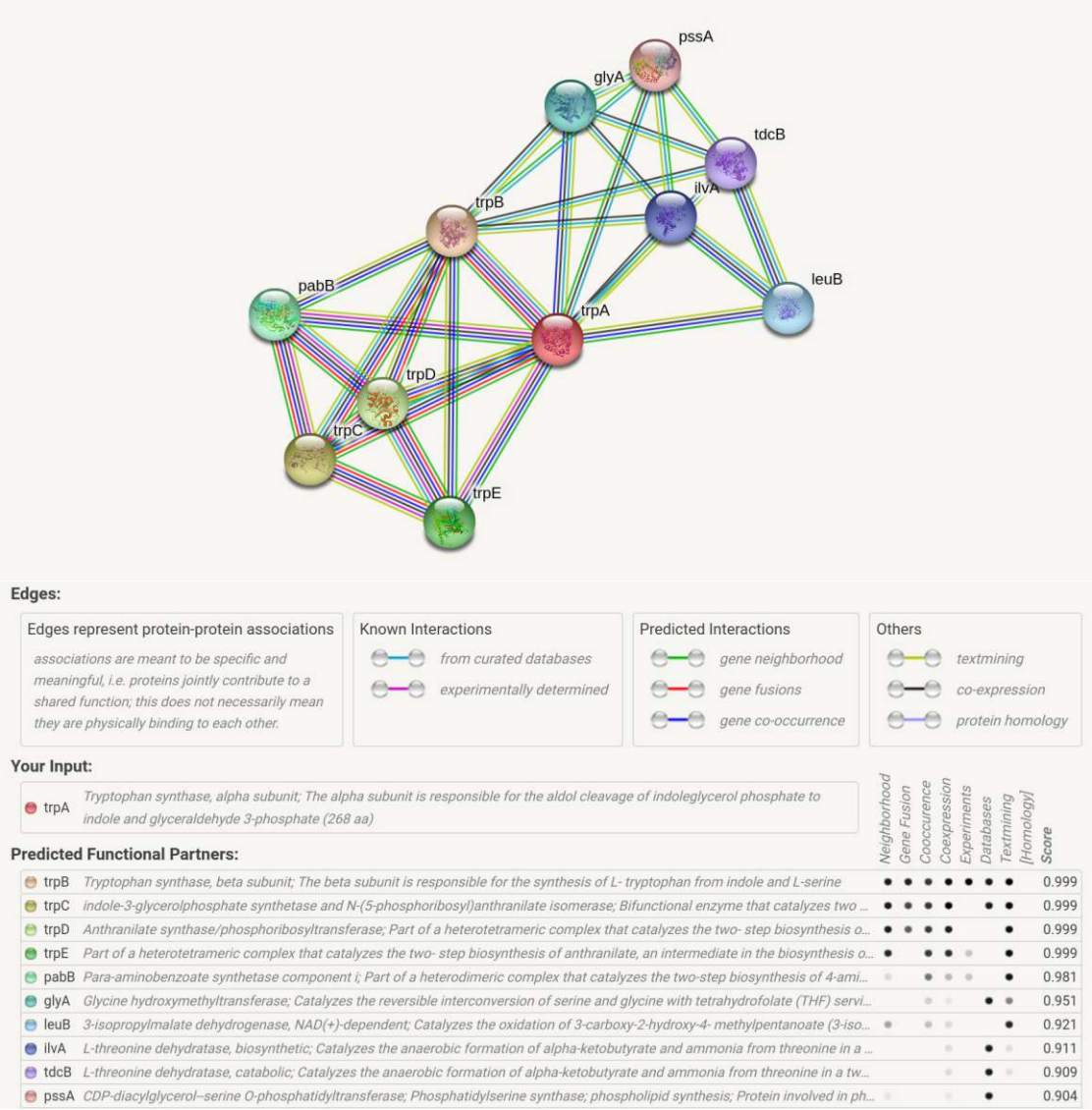


Figure 2.12: An exemplary association network from the STRING homepage [SGL+19]. Reprinted from "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets", by D. Szklarczyk et al., 2019, Nucleic Acids Research, D1 (47), D607-D613. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

Another such system, designed to visualize protein-protein interaction networks, is *Biolinker* [DMF17]. *Biolinker* allows to *select* proteins from a large database and visualizes it and its interaction partners as a *force directed node-link* diagram of a *simple graph* (see Figure 2.13). Here, vertex size is used to indicate the number of interaction partners (*centrality*), while the edge colors indicate types of interaction. The main view

can be *filtered* by selecting categories in the accompanying table. Users can also select two proteins for which all possible paths are then shown (*connect*). In order to gauge the trustworthiness and the underlying evidence, *Biolinker* features an auxiliary visualization showing publications for each of the interactions featured in the main view. Publications are shown as arcs, colored according to the respective interaction type, and placed at the year of publication of the evidence. The publications are also shown in the *conflict matrix*, which indicates if different publications record different interaction types for a given protein (see Figure 2.13).

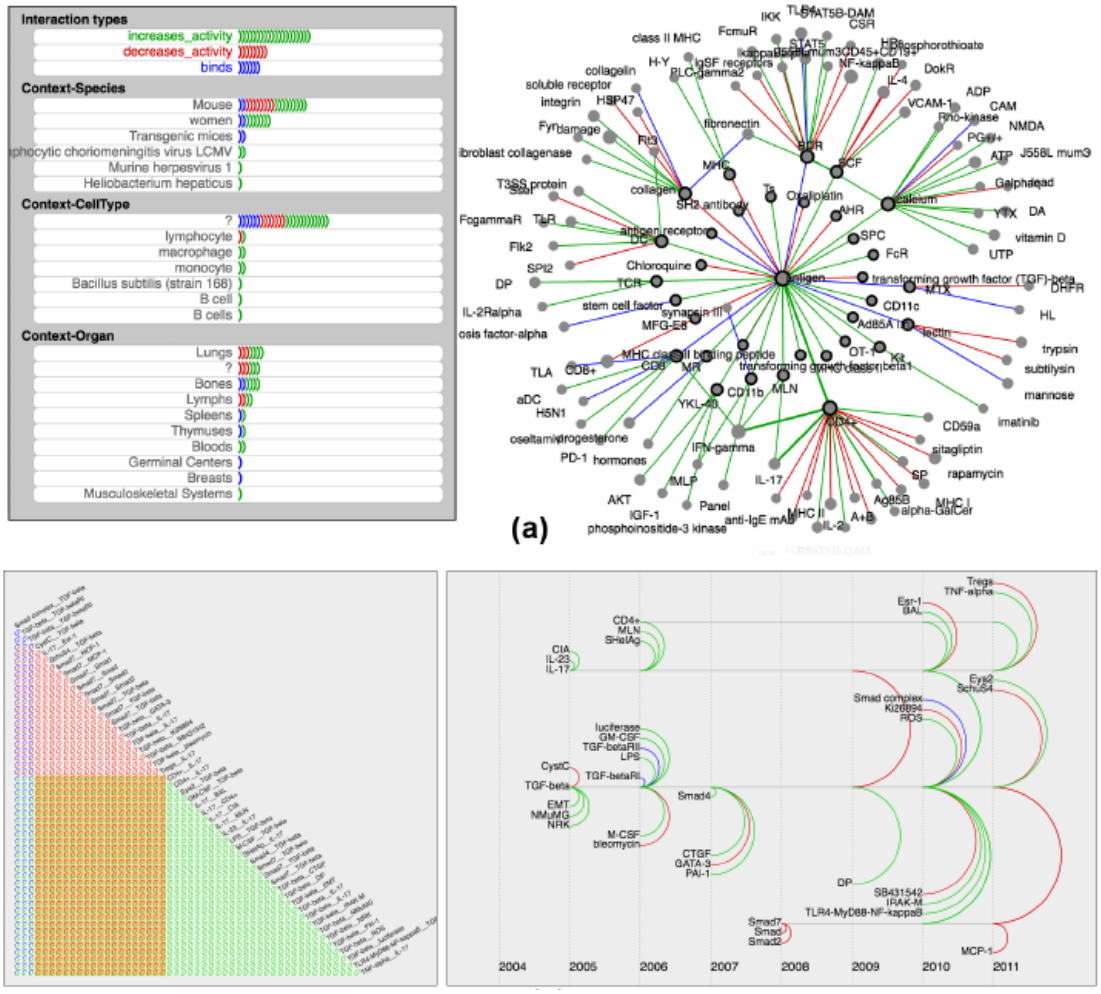


Figure 2.13: Main view of Biolinker, Conflict matrix, and literature arcs of Biolinker [DMF17]. Authors contacted requesting copyright for reprinting.

In part, owing to the rather simple structure of protein-protein interaction networks, many visualization systems with different approaches exist for their visualization. There are approaches aimed at visualizing the underlying geometry in PPI-Networks [ALMAN18],

inferring PPI-Networks from existing data [JLW+20], showing *node-link* ego-graphs for comparing the similarity of PPI-Networks [BHP+24], or approaches using *matrix* views instead of *node-link* diagrams [DMF15].

Metabolic Pathways

Knowledge of metabolic pathways and their reactions is important for a multitude of research questions across domains, be it molecular biology, biochemistry, or biomedical research. Thus, metabolic pathway networks are among the most well known and most widely applied biological networks types. Most molecular biologists or biochemists will be familiar with static visualizations like *Biochemical Pathways* [Mic17] or, with added interactivity, *ReconMap* [NDG+17]. Additionally, databases like KEGG [KG00], BioCyc [KBC+19] or Reactome [BWK+16], along with vast amounts of data, also offer manually curated network visualizations.

As manually generated pathway visualizations cannot be tailored to specific use cases, automated approaches have been developed. These automated approaches, however, often cause poorly readable layouts. Recently, different approaches have specifically targeted this drawback. *Metabopolis* [WNSV19], for example, addresses this problem by utilizing techniques employed in city planning to generate a more readable and familiar layout of a *clustered graph* for automated metabolic pathway visualizations. This graph is shown in a *schematic and orthogonal node-link* diagram. In detail, *Metabopolis* only displays user-defined categories as *urban blocks* and lays them out by solving a constrained *floor-plan problem*. Afterward, intra- and inter-block connections are drawn, finalizing the visualization (see Figure 2.14). Users of *Metabopolis* can also interact in various ways with the visualization. They can, for example, *reconfigure* the layout city blocks or *select* and *connect* start and endpoints of routes which are then highlighted.

Moreover, metabolic network visualization plays an important role in systems biology, where cellular metabolic processes are simulated to gain insight into these processes. *VisANT* [GCW+16] is such a tool, purpose-built for simulating cells, organisms, and their interaction. In the latest iteration of *VisANT*, particular emphasis was given to modeling and visualizing interaction between different cells or event-complete, distinct organisms like bacteria. In general, *VisANT* uses a *bipartite graph* of metabolites and reactions, in a *schematic and orthogonal node-link* diagram, to show the involved metabolic pathways. Additionally, in the so-called *metagraph*, a *clustered graph*, organizational units, organelles, or whole bacteria, can be aggregated in so-called *metanodes* (*encode*, *abstract/elaborate*). Users can collapse or expand these *metanodes* on demand (*reconfigure*) Between these *metanodes*, expanded or aggregated, *exchange*-vertices indicate metabolite exchange between the respective organelles/organisms (see Figure 2.15). *VisANT*, however, is not the only platform offering systems biology functionality, with many tools offering avenues to generate hypotheses (e.g. [NVR14, FMJ+08]).

Another task, common for metabolic pathway networks, is the comparison of different experimental conditions or the comparisons of similar pathway segments in the

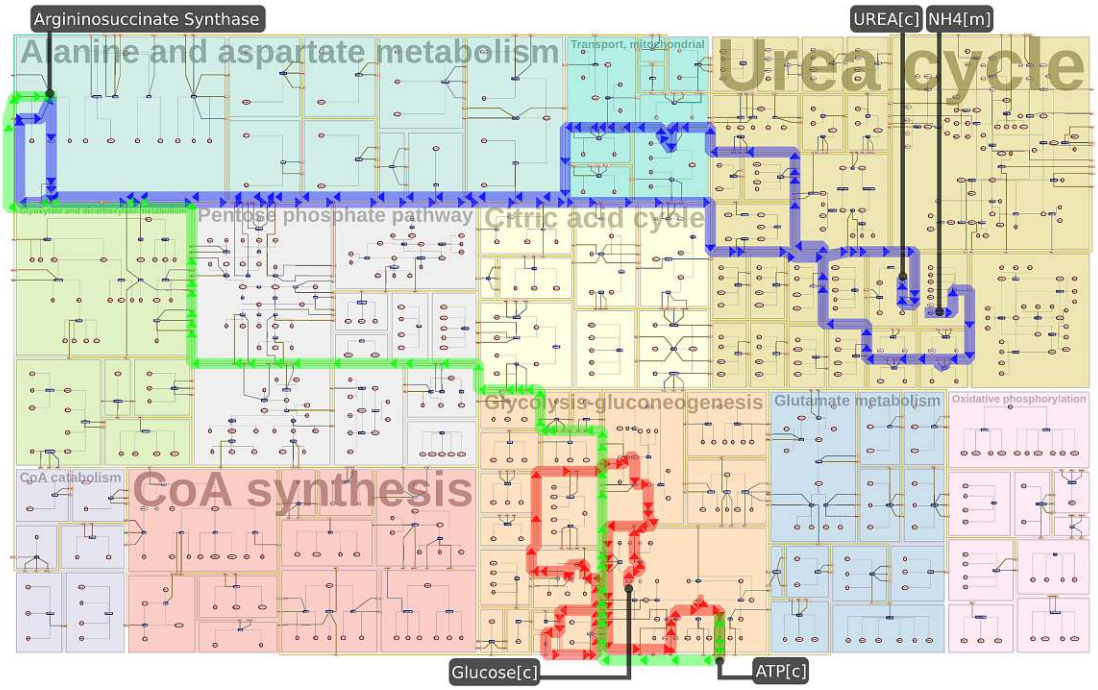


Figure 2.14: Metabolic pathway visualization in Metabopolis [WNSV19]. Reprinted from "Metabopolis: scalable network layout for biological pathway diagrams in urban map style", by H. Y. Wu et al., 2015, BMC Bioinformatics, 1 (20), 187. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

grand ensemble of metabolic pathways. This task introduces an additional challenge for graph visualizations, as they require visualization to facilitate graph comparisons. Entourage [LPK+13] was developed to compare different sub-graphs of the complete metabolic pathway graph as laid out by KEGG. They mainly target the process of drug design or drug repurposing. Their tool aids in this effort by offering a linkage of compounds/drugs in a given sub-pathway to other sub-pathways in which a given element appears, intelligently showing the surrounding metabolic processes. Cerebral [BMGK08], on the other hand, focuses on the visualization of many different conditions for a given metabolic network. The primary use case of Cerebral is to compare multiple experimental conditions, like time points in a time course experiment, in automatically generated but familiar, e.g., sorted by cellular location, and layouts.

Due to the importance of metabolic pathways, many tools offer the visualization of such metabolic pathway networks (e.g. [RPM20, DNW13]). Also, many tools for biological networks feature add-ons allowing the visualization of metabolic pathways (e.g., [ACZ+21]). Additionally, some of the general-purpose network visualization frameworks like yEd [yWo22] and Cytoscape [SOR+11] can be used to draw metabolic pathways (e.g.[LOP+18, BMSW19]).

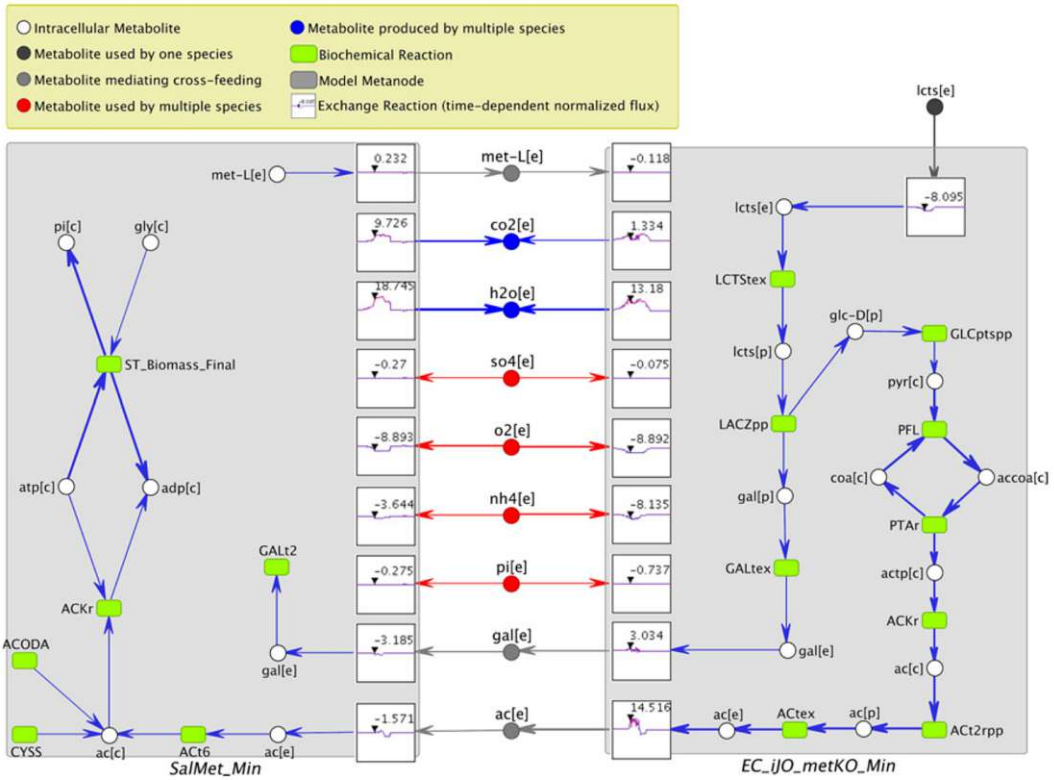


Figure 2.15: Visualization of two interacting bacterial metabolic pathways in VisANT [GCW+16]. Reprinted from "Visualization of Metabolic Interaction Networks in Microbial Communities Using VisANT 5.0", by B. R. Granger et al., 2016, PLOS Computational Biology, 4 (12), e1004875. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

Multi-Omics Networks

For specific biological experiments or research questions, individual analysis of specific classes of molecular-biological entities can play a pivotal role. Considering a more comprehensive set of molecule classes for analysis, however, might be more appropriate for more complex research questions, as the actual biology oftentimes does not allow for a simplified view, restricting itself to the analysis of a single omics-type. Thus, multi-omic network models, networks combining more than one specific class of molecular-biological entities, and corresponding visualizations were developed to aid in such tasks. While biomedical research questions are one of the primary focus areas of multi-omics research [HSL17, SVK+20, HCG17], other biological domains, for example phylogenetics [HWM+15] or plant physiology [ZLC+20] profit from multi-omics approaches. One key challenge in multi-omics visualizations is the integration of heterogeneous data sources into a coherent model. A possible approach is the use of multi-layer networks in which each data source corresponds to a distinct layer of the network. *OmicsNet* [ZX18] features

the application of such a multi-layer layout. Using gene, protein, miRNA, and metabolite lists, users can construct up to three networks, of which two are used to add additional information to the main network, generating a *multilayer-graph*. This, e.g., allows for the generation of a protein-protein interaction network, in which transcription factors and miRNAs targeting proteins can be visualized as additional layers (Figure 2.16).

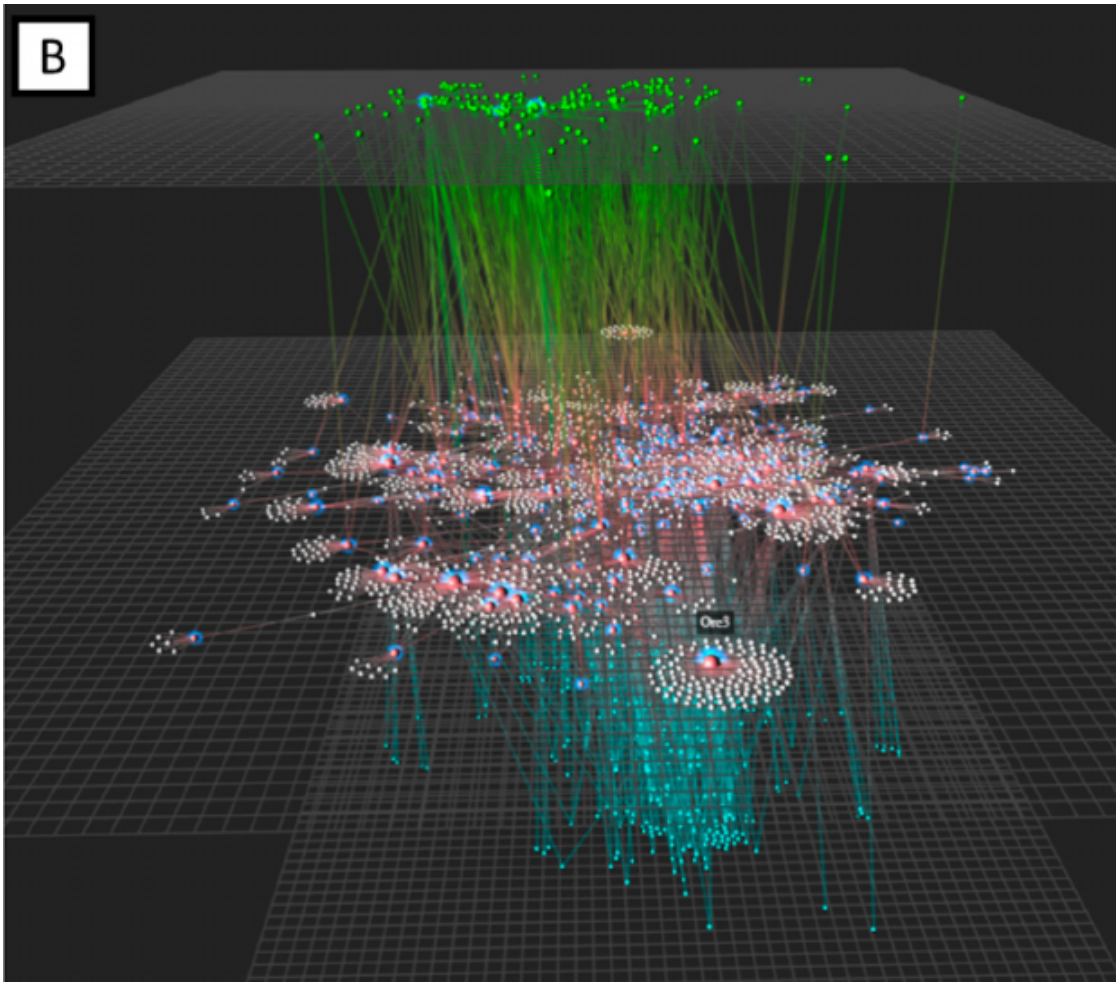


Figure 2.16: Multilayer network of transcription factors, proteins, and miRNAs (top to bottom) in *OmicsNet* [ZX18]. Reprinted from "OmicsNet: a web-based tool for creation and visual analysis of biological networks in 3D space", by G. Zhou et al., 2018, Nucleic Acids Research, W1 (46), W514-W522. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

As this kind of multi-layer visualization often introduces the need for 3D visualizations, it also introduces its drawbacks ([Mun14, MGM+19]). Thus, other multi-omics applications aim to integrate the different omics layers in a two-dimensional visualization [HdDTMM+18, KTT13]. *PaintOmics 3* [HdDTMM+18], for example, can visualize

complex experimental setups involving multiple types of omics-data, namely transcriptomics, proteomics, metabolomics, modifications, and regulatory elements like siRNA and transcription factors. Given a set of matching omics measurement-sets *PaintOmics 3* generates an overview graph (a force-directed node-link diagram of a simple graph), in which each vertex corresponds to a pathway of the manually curated metabolic pathways from *KEGG* (*encode*). Each pathway can be selected individually such that the manually generated pathway networks from *KEGG* (a schematic node-link diagram) are displayed (*abstract/elaborate*). Inside this individual pathway, each omic type measurement is mapped to its corresponding pathway vertex as a colored bar (see Figure 2.17).

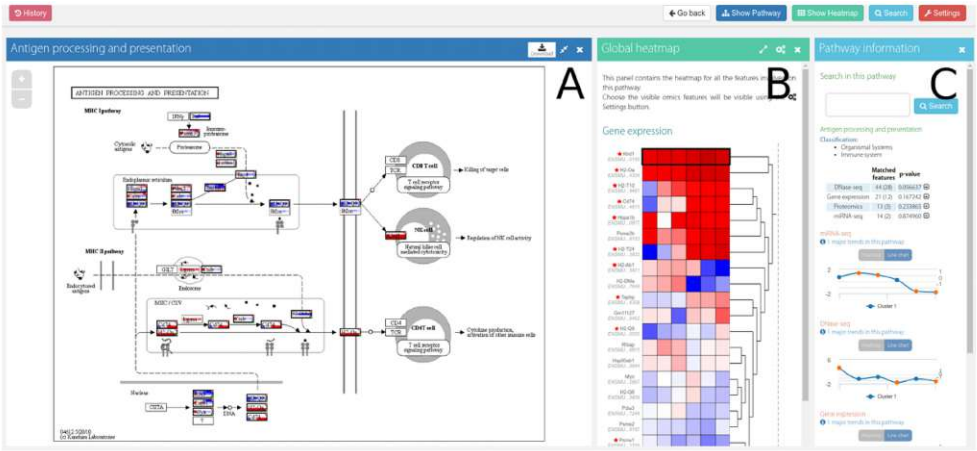


Figure 2.17: Multilayer network of Transcription factors, proteins, and miRNAs (top to bottom) [HdDTMM+18]. Reprinted from "PaintOmics 3: a web resource for the pathway analysis and visualization of multi-omics data", by R. Hernandez-de-Diego et al., 2018, Nucleic Acids Research, W1 (46), W503-W509. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

In general, many frameworks are not tailored to a specific type of biological interaction network but can, in fact, be used to analyze a wide array of different networks and use cases. One of those frameworks is the popular *VANTED* framework [RJH+12]. *VANTED* offers a combination of manual and automatic layout functions (e.g., schematic node-link diagram as seen in Figure 2.18) and features a complete customization of vertices and edges. Experimental conditions, for example, can be mapped as bars, or as color, to the vertices, allowing different analysis approaches (*encode*) (see [RJH+12]) An additional feature of *VANTED* is the openness of the system, allowing the integration of most omics-types and for many use cases, including systems biology simulations and experimental analyses. Additionally, many of the more general-purpose tools, like Cytoscape, offer add-ons to visualize generic biological networks (e.g. [WZC+15]).

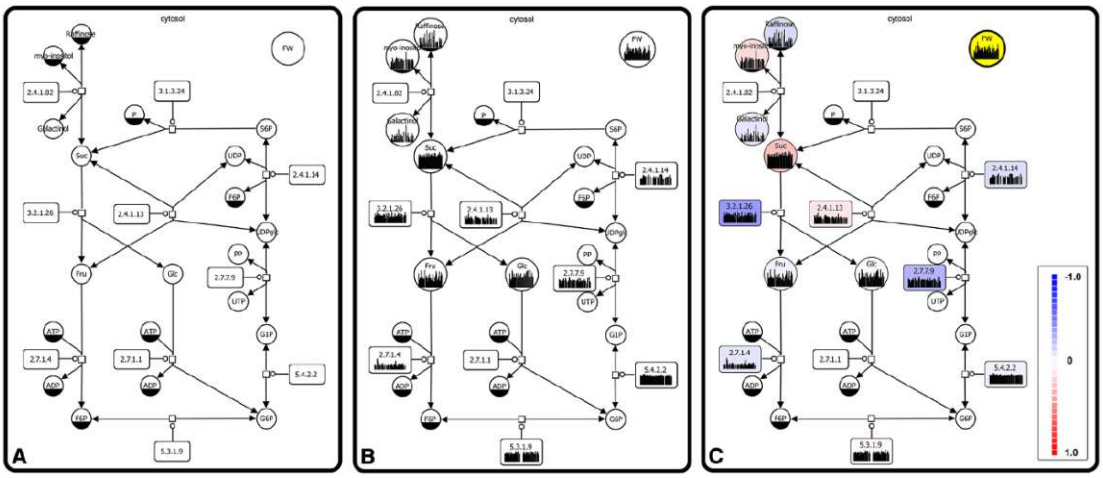


Figure 2.18: Metabolic pathway visualization [RJH+12]. Reprinted from "VANTED v2: a framework for systems biology applications", by R. Rohn et al., 2012, BMC Systems Biology, 1 (6), 139. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

2.9.2 Biological Networks based on Genomic Variations

While different species can share large parts of their genetic code, the differences in this code determine the particular species. However, the genetic code of two individuals of the same species is also not exactly the same: the differences determine the individual phenotype (i.e., observable traits of an organism like sex or eye color). Therefore, genomic variations are studied intensively. These variations include the deletion, insertion, inversion, and repetition of (usually smaller) parts of the genetic code. A graph can be constructed that captures all this information about the variations and allows to visually analyze them. Common tasks include the analysis of shared parts of the genetic code as well as studying the differences that lead to specific traits.

Network-based visualizations of genomic variation are, thus, useful for the analysis of *assembly graphs* and *genomic variation graphs*. The *VG* toolkit [GSN+18] provides a common data model to describe such variation graphs built from multiple variants of a DNA sequence. As the name implies, the *Sequence Tube Map* [BNH+19] tool uses a tube map metaphor for visualizing such variation graphs, where all sequences are shown as parallel tracks and homologous regions are marked. The *Bandage* [WSZH15] tool results in more compact layouts, as it does not adhere to the strict "train track" metaphor but allows for arbitrary, curved layouts. *ODGI* [GHN+21] includes a similar visualization. It is optimized for scalability and can create layouts for whole pangenome graphs. The recent review by Eizenga et al. [ENS+20] provides further details on pangenome graphs and compares the abovementioned visualization approaches (Figure 2.19).

A *Quantitative Trait Locus* (QTL) is a region of DNA on a chromosome that is associated with a specific phenotypic trait that is measured on a continuous scale, e.g., the growing

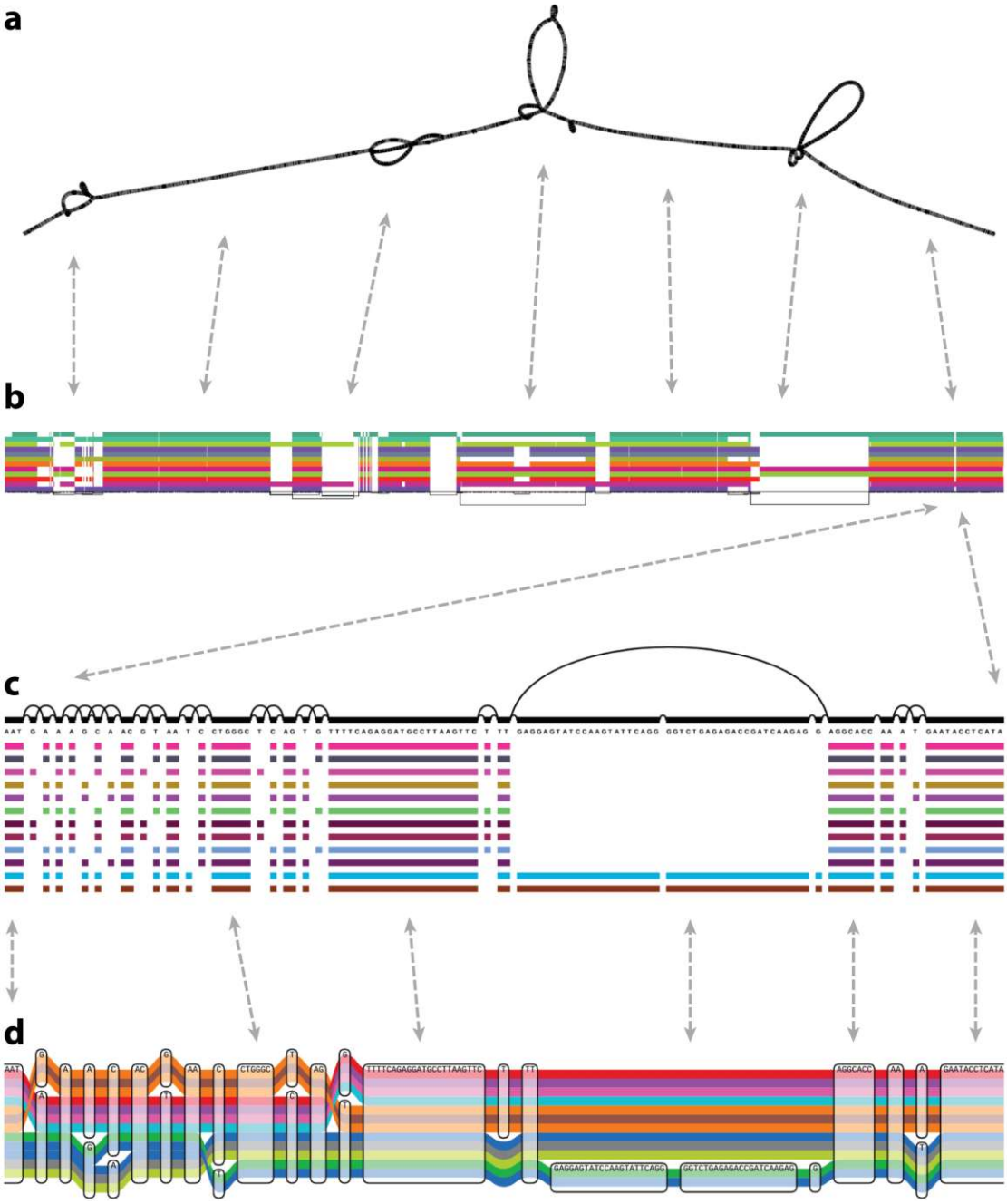


Figure 2.19: Genomic variation graph visualizations from different applications [ENS+20]: (a) Bandage, (b) ODGI viz, (c) VG viz, (d) Sequence Tube Map. The gray arrows highlight the correspondence and scalability of the different approaches. Reprinted from "Pangenome Graphs", by J. M. Eizenga et al., 2020, Annual Review of Genomics and Human Genetics, 1 (21), 139-162. Copyright: Annual Reviews Reprinted with permission.

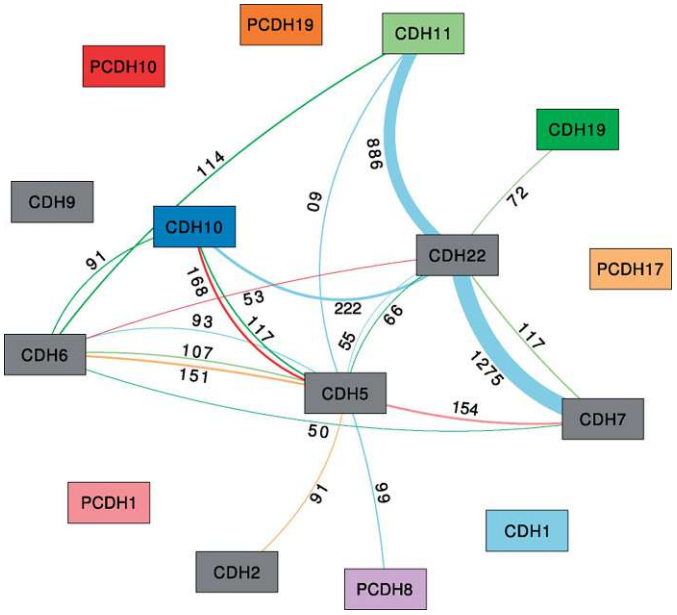


Figure 2.20: The Reveal tool derives gene association graphs from eQTL data and visualizes them as node-link diagrams [JBN12]. Reprinted from "Reveal—visual eQTL analytics", by G. Günter et al., 2012, Bioinformatics, 18 (28), i542-i548. Copyright: Open Access - Creative Commons CC BY 3.0. Reprinted with attribution.

height of a plant or human skin color. Such traits are usually determined by two or more genes. Further information can be found, e.g., in the book by Rifkin [Rif12]. To facilitate the analysis of QTLs, specific tools for the visualization of QTL networks have been developed. A common visualization of QTL interactions is to draw edges between the sequences of the respective chromosomes, as for example used by *QTLNetwork* [YHH+08], *QTLNetworkR* [ZYZ10], or *solQTL* [TMB+10]. However, this simple visualization often does not capture the full interaction network. Jiang et al. [JSS+19] presented a computational model for inferring QTL-QTL interaction networks and visualized them as simple *node-link* diagrams, similar to the interaction network tools discussed in subsection 2.9.1. The *Reveal* tool by Jäger et al. [JBN12] visualizes the genes associated with the QTLs as vertices of a *force directed node-link*-diagram (*simple graph*). Edges signify single-nucleotide polymorphism (SNP) pairs in two genes that significantly influence the expression of another gene (see Figure 2.20). The tool was originally developed for the BioVis Challenge 2011, which focused on QTL expression data [BYCH+12]. For this challenge, Paquette and Lum clustered SNPs and visualized the results as *node-link*-diagrams using the *Iris* tool (Ayasdi, Inc.) [BYCH+12]. Here, the vertices represent clusters, which are connected by edges if they share at least one SNP.

2.9.3 Phylogenetic Trees

Trees are acyclic graphs, that is, two vertices are connected by exactly one edge. In biology, *phylogenetic trees* are one of the most common uses of trees. In general, phylogenetics is concerned with the study of the interrelationship of different species. Today, phylogenetic trees that show the relatedness of different species are often constructed based on genomic information, e.g., by comparing differences in the sequence of proteins or genes. These trees are usually visualized as dendrograms, where the leaves are the species, the inner vertices are the (hypothetical) ancestors, and the distance between vertices shows the evolutionary difference. Besides the analysis of different species, phylogenies are also used to derive taxonomies of different species. The online tool *Lifemap* by de Vienne [Vie16] allows us to interactively explore the tree of life, i.e., the taxonomy of all known species provided by the NCBI. Another example is the *NextStrain* project [HMB+18], which, for example, offers an interactive visualization of the phylogenetic tree of all currently known SARS-CoV-2 virus variants. The *FastTree 2* tool by Price et al. [PDA10] can compute phylogenetic trees for hundreds of thousands of samples. The results can be visualized using various applications such as the online tool *iTOL* [LB21], which, for example, offers radial and linear layouts.

Besides such relatively simple tools that only show the phylogenetic tree in an interactive view, more advanced visual analysis tools have been proposed, which are tailored to specific application cases. One example is *TreeJuxtaposer* by Munzner et al. [MGT+03], which allows for the comparison of the structure of large phylogenetic trees with hundreds of thousands of vertices. The trees are visualized as a dendrogram, and a specialized *focus+context* technique is used to ensure the visibility of differences. Similarly, the online tool *Phylo.io* by Robinson et al. [RDD16] also uses juxtaposition of dendrograms for a comparative visual analysis. Another example is the metagenome analyzer *MEGAN6* by Huson et al. [HBF+16], which also offers a *layered* dendrogram view that shows the extracted phylogenies. It can use data from a given taxonomy, e.g., NCBI, to construct the tree. To compare the number of different species found in a metagenomic sample, the leaf vertices of the tree *encode* the number of reads for that species as a bar chart.

2.9.4 Other Biological Networks

In addition to the most common biological networks mentioned above, there are other examples such as brain networks, ecological networks, and networks that capture the topological structure of high-dimensional data of single-cell RNA-sequences. Although our survey does not focus on such networks, We will discuss some exemplary applications for these types of biological networks for the sake of completeness.

A *brain network* is a network of neurons that represents the functional connectivity of neurons in the brain, as measured by various methods such as functional Magnetic Resonance Imaging (fMRI), electrophysiological measurements, or calcium imaging in humans and model organisms. The brain network is visualized to understand its system and to formulate and validate hypotheses. In particular, the brain network is dynamic

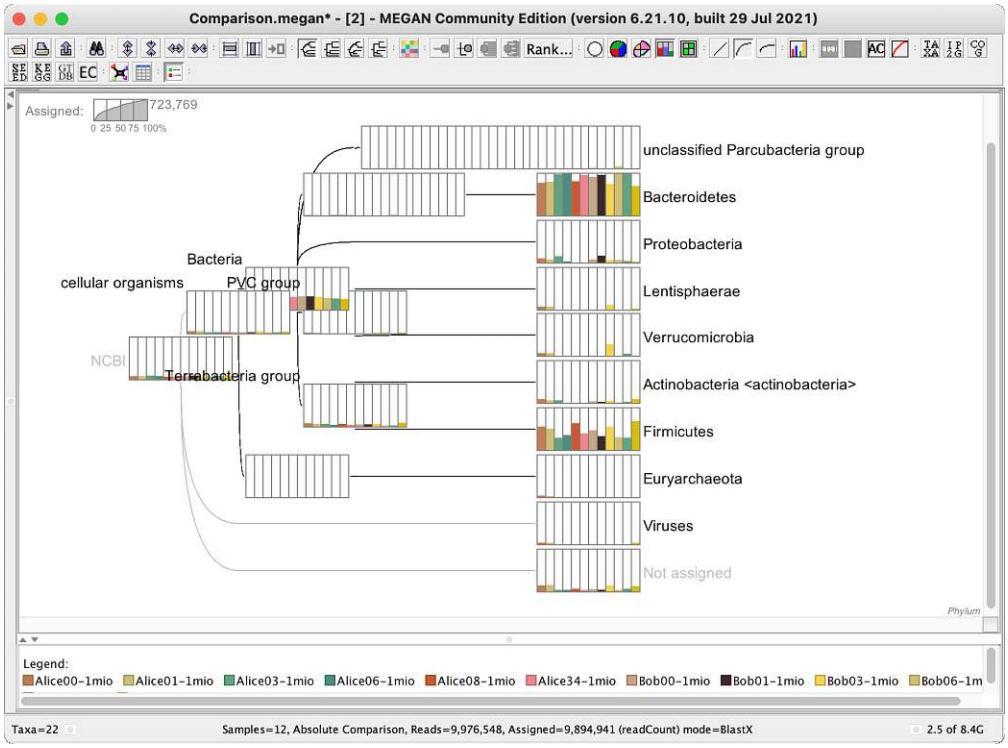


Figure 2.21: Screenshot of the comparative phylogenetic tree visualization offered by MEGAN [HBF+16]. Reprinted from "MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data", by D. H. Huson et al., 2016, PLOS Computational Biology, 6 (12), e1004957. Copyright: Open Access - Creative Commons CC BY 4.0. Reprinted with attribution.

and has a large and dense structure. Therefore, hybrid visualizations have been proposed to handle these features effectively. Since animations and small-multiples do not scale to large networks in some tasks, Bach et al. [BHRD+15] proposed a hybrid visualization method to summarize dynamic features by displaying them as so-called multi-piles, that is, piling *adjacency matrices*, enabling detection of high-level temporal patterns (see Figure 2.22(a)). In addition, various visualization techniques are used, such as node-trix [YSD+17] (see Figure 2.22(b)), which utilizes a *hybrid matrix-node-link* layout, a visual metaphor called NeuroLine [AABS+14], and edge bundling [BSL+14] to analyze large-scale and locally-dense structures of the brain. It is also effective in supporting the comparison and exploration of networks by combining machine learning with interactive visual interfaces [FCM+17].

Ecological networks represent food webs and interspecies interactions in an ecosystem. Although these networks usually have dynamical properties, they are more difficult to measure and quantify than those in neuroscience since they are spatiotemporally extensive and difficult to control experimentally. Therefore, it is more important to

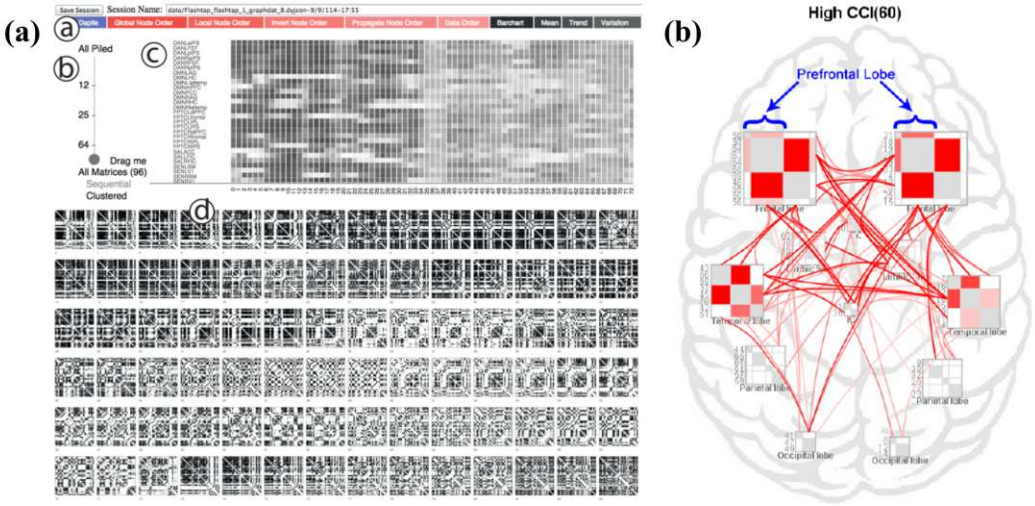


Figure 2.22: (a) Screenshot of the **MultiPiles** application by Bach et al. [BHRD+15]. Reprinted from "Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks", by B. Bach et al., 2015, Computer Graphics Forum, 3 (34), 31-40. Copyright: The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.. Reprinted with permission. (b) **NodeTrix** visualization of the brain network by Yang et al. [YSD+17]. Reprinted from "Blockwise Human Brain Network Visual Comparison Using NodeTrix Representation", by X. Yang et al., 2017, IEEE transactions on visualization and computer graphics, 1 (23), 181-190. Copyright: The Institute of Electrical and Electronics Engineers (IEEE). Reprinted with permission.

combine visualization techniques and visual analysis approaches with mathematical methods for the calculation of interspecies interactions in ecology [NDP+21].

Finally, in the field of biology, sc-RNAseq has made it possible to examine the sequence information from individual cells with optimized next-generation sequencing technologies, causing an information explosion. In addition to mapping and comparing gene expression at the single-cell level, biological *networks of cell differentiation* can be visualized by topological data analysis (mapper) that analyzes high-dimensional data structures as a graph [PSGG+19]. Because of the need to interpret graphs along with heterogeneous data, Zhou et al. [ZCR+21] presented the Mapper Interactive framework, which combines mapper and attribute data through an interactive visualization system.

While the above are just a few selected examples, they show that there is an increasing need for applications that integrate novel data, analytical models, and visualization techniques to facilitate the visual analysis of biological networks.

2.10 Challenges

The outlined *Graph Models*, *Graph Analysis Methods*, *Network Visualizations*, as well as *Graph Analysis Tasks* and their *Interaction Methods* make up the interconnected building blocks of the information visualization pipeline [Car99] underlying biological network visualizations across domains (Figure 2.2). While the discussed *exemplary applications* make the importance of these building blocks clear, individually, these cannot shed light on the overall state of the larger field. Hence, *as discussed previously*, all collected tools and papers were systematically collected from relevant journals, filtered, and categorized within our developed taxonomy. Based on these findings (Figure 2.2), as well as individual observations filtering and reading this corpus of literature, we identify several open research opportunities of potential interest to both the communities of network and information visualization, as well as the many subdomains that make up biological network analysis and visualization.

2.10.1 Beyond Schematic and Straight-Line Node-Link Diagrams

Our collection of papers (Figure 2.2) indicates that most biological graphs are either (manually) drawn schematic representations, as seen in *NeuroLines* [AABS+14], *ODGI* [GHN+21], or *Sequence Tube Maps* [BNH+19], or drawn as force-directed node-link diagrams, as seen in *miDerma* [CH18], *xiNET* [CFR15], or *DORMAN* [OSA+21]. This can likely be attributed to i) the computational complexity and poor scalability of more advanced drawing algorithms, such as octo/rectilinear or layered layout algorithms, ii) their high availability and ease of implementation in common graph visualization tools and libraries, and iii) the subjective preferences of domain experts for these schematic representations, as biologists and biochemists prefer these manually-created, octo/rectilinear layouts that they have grown familiar with after years of use [KG00, NDG+17].

Although the force-directed layout algorithm is considered one of the more scalable approaches, its layouts become harder to interpret with increasing graph density and size [Kob14], forming so-called "*hairballs*" [YDK+18]. This deterioration of visual quality and readability has been noted and measured, not only using quantitative graph aesthetic metrics, such as the number of edge crossings [KPS14] or edge-angle-ratios [Pur02], but also empirically evaluated in several comparative, quantitative user studies [OJK18, GFC04]. Generally, it would appear as though node-link diagrams are, in comparison to, for example, adjacency matrix representation, more sensitive to the size and density of the input graph. Most likely this is owing to only indirectly optimizing important graph aesthetic metrics [Kob14], which in turn impact user performance [Pur97, EVRW23].

Beyond their larger sizes and higher densities, many biological networks also feature node groupings, based on their cellular location [BMGK08], (predicted) function [MEO+24], biochemical pathway association [LPK+13], or some data-driven clustering [KBT+10]. Force-directed, straight-line node-link diagrams, beyond node attribute encodings, do not effectively allow such groups to be visually discerned. Indeed, to effectively visualize such group structure, a plethora of different visual relationships between graph topology and

node groupings, and visual encodings of said group structure are available [EMWR24, VHTW13]. At least within the context of biological network visualization, these options seem not to have yet been effectively explored and utilized.

Overall, we see ample scientific opportunity in the development of biological visualization approaches and tools beyond domain-standard, hand-drawn schematic representations or commonly available force-directed node-link diagrams. While the former may always play a part in tools aimed at domain experts, more effective alternatives are available For example, radial layouts offer advantages for communicating group structure and could be used more frequently for such purposes. Alternatively, given the large quantity of metadata associated with both nodes and edges, application-specific hybrid approaches (Figure 2.2) could allow for a more effective visual exploration of networks and the multivariate data attached to them [NMSL19].

2.10.2 Incorporating Uncertainty in Network Visualizations

Uncertainty can arise at many stages in biological network visualization, from data collection to the conducted automated numerical analyses [KDJ+21]. Especially important are the uncertainties and weights attached to networks obtained from online knowledge graph repositories, such as *KEGG* [KG00] or *Reactome* [VDS+07], as these graphs, integrated with newly collected experimental data, often form the basis of biological network visualizations. However, while topology is naturally always communicated in such applications, the uncertainties and weights often attached to such graphs' edges are commonly ignored for simplicity. Even outside of the context of network visualization, the visualization of uncertainty remains uncommon [Hul20], despite an understanding that it can improve awareness, analysis results, and thrust [SSK+16]. Instead, the various interactions of these networks are often depicted as deterministic and equal, even though this may not be the case. Some approaches, such as *STRING* [SGL+19], do include these uncertainties in their visualization, but do so only as a filter applied to the input data. Here, we see ample opportunity for future work, as the visual communication of this information could provide more insight into the various biological networks under study. Care, however, must be taken when implementing visualizations of uncertainty, as inappropriate or poorly designed implementations may cause misunderstandings or confusion.

2.10.3 Incorporating Graph Analysis in Network Visualization

While we found many visualization tools designed for a variety of different domains, types of graphs, and analysis goals, only a few employed or supported numerical graph analysis (Figure 2.2). The most common quantitative analysis supported by biological network visualization tools was the identification of (node) clusters. As discussed previously, looking at these papers in closer detail, clustering is specifically employed to provide

1. a means of organizing the network's visual representation, commonly correlation or adjacency matrices, as seen in *PathwayMatrix* [DMF15] or *Kiwi* [VGN14], in

order to guide users to biologically meaningful parts of the data,
2. a means of simplifying a network's visual representation, such as i) the coloring and grouping of nodes, e.g., Cruz et al.'s visualization of dynamic time-series data [CPAM18], ii) the breaking up of a graph into simpler sub-components, as seen in *ClusterViz* [WZC+15] or New et al.'s visualization of dynamic co-expression data [NKHC08], or iii) the bundling of edges in node-link diagrams, e.g., Lambert et al.'s visualization of metabolic networks [LDB11],
3. a mechanism for data-driven hypotheses generation, as seen in *Caleydo* [LSKS10] and *Mapps* [RPM20].

Even straightforward and comparatively simple graph descriptive metrics, such as vertex centrality, or graph density were surprisingly uncommon (Figure 2.2. When utilized, these metrics are primarily used either for the straightforward ranking/filtering of entities [DCG+12, FSA+16] and clusters [NYP12, YSD+17], or the visual highlighting of nodes [KWKJ19], to guide domain expert users to regions of potential graph-theoretic, and potential substantive, interest. More involved or complex graph analysis approaches, with the possible expectation of motif identification, are even less common, likely owing to the complexity of implementation.

Perhaps because of the lack of more advanced quantitative analysis approaches, most visualization tools surveyed appeared to be designed for exploratory analyses and hypothesis *generation*. Indeed, only very few tools were designed with hypothesis *verification* explicitly in mind. Two key examples that were, however, are *MAPPS* [RPM20] for edge prediction in metabolic pathways, and *VANTED* [HJ18] for motif identification in omics networks. Both are positioned as *integrated* visualization and analysis platforms to allow domain experts to both generate as well as (statistically) validate topological and substantive hypotheses.

This lack of more advanced topological analyses, in our opinion, provides an opportunity for the field of biological network visualization. While, many visual tools are, and will continue to be, needed for the straightforward visual exploration of biological networks, we also see value in including more sophisticated analysis techniques, such as edge prediction, motif identification, or graph comparison, as such approaches could allow for both a complete generation and verification of domain expert hypotheses.

2.10.4 Towards Better (Visual) Network Comparison

As shown earlier, network similarity is an analysis metric not used often in the collected set of papers. Similarly, only a few tools support the *graph comparison* task. However, since experimental data collected in biology is often heterogeneous, researchers would like to compare [SN11, ABHR+13], for example, the similarity of biochemical (reaction) topologies across different databases [DMF17], individuals/samples [FCM+17], biological compartments [AABS+14], pathways [RPM20, LPK+13], time points [CPAM18, PD17], or experimental conditions [BMGK08] in order to investigate potential solutions for research questions, such as drug target identification. Outside of metabolic pathway

visualization, graph comparison as a task is even less supported, even though visually comparing gene regulatory networks or protein-protein-interaction networks are biologically relevant application areas.

A number of visual approaches to (graph) comparison present themselves. Two or more (sub)graphs can be compared through juxtaposition, partitioning, or superimposition. [VRW14, EMWR24]. Within the collected set of literature, most modern tools opt for a simple side-by-side juxtaposition of (sub)graphs, as seen in Cerebral [BMGK08]'s visualizing of graphs across experimental conditions. An approach that utilizes both *network similarity* and supports *graph comparison* is the *MultiPiles* approach by Bach et al. [BHRD+15]. Here, the networks in their matrix representation are grouped into piles. A *cover matrix* — a representation of the whole stack — acts as a visual representation of all matrices, by displaying different metrics of the pile. *MultiPiles* is joined by other matrix based approaches [YSD+17, DMF17, FCM+17, NKHC08]. Other approaches, with the exception of *DynoVis* [KWKJ19], do not utilize similarity metrics when comparing networks.

This indicates two key open challenges and research opportunities in the visual comparison of graphs. First, as outlined earlier, there is clearly a gap in the analysis methods available in network visualization tools. Especially noteworthy is the lack of graph comparison tools, such as the previously discussed *DeltaCon* or *Cut Distance* measures (Section 2.5.3). Second, most visual graph comparison approaches identified make use of interchangeable or juxtaposed adjacency matrices. For one, non-matrix-based visualizations could also be investigated. Additionally, non-juxtaposition or interchangeable approaches present themselves for future research. In addition to the gap in visualization techniques, the gap in specific application domains becomes apparent — besides general-purpose frameworks like *VANTED* [RJH+12, HJ18], there appears to be little support for visual graph comparison in protein-protein-interaction-networks, genomic variation graphs, or multi-omics networks. Multi-layer networks could be used to facilitate a less side-by-side-centric approach to comparison.

2.10.5 Provenance and User Trust

Related to the aforementioned challenges of uncertainty as well as provenance visualization is the question of user trust, i.e., the degree to which a user trusts either the information presented or the conclusions they reach [MHSW19]. Indeed, there is increased appreciation and awareness of the importance of trust in visualization and the factors that influence it, such as the accuracy, currency, completeness, objectivity, validity, and predictability of data presented [KFW08]. Awareness of trust is especially important when users are expected to evaluate or work with automated (AI/ML-based) analysis methods. Visualizations that aim to explain such automated (AI/ML-based) methods' inner workings and results to increase user trust in such methods are currently gaining a lot of attention [SSSEA20, CMJ+20].

As discussed previously, the application of automated (topological) analysis methods within the context of (exploratory) biological network visualization is still in its infancy. However, their inclusion would also bring the challenge and opportunity of effectively communicating these methods' inner workings and results to domain experts, ensuring higher levels of user trust. An integrated system with faithful data and visual representations could allow users to visually probe and explore the limits of their data and analysis reliability, perhaps coupled with provenance analysis, in order to improve user trust.

2.10.6 Dynamic Biological Network Visualization

Dynamic network visualization focuses on the visual communication of a network's topological evolution over time. While the topic of dynamic network visualization outside of the biological context has already received a fair amount of attention [BBDW14, BBDW17, LPP+19], the visualization of dynamic biological networks is still in its infancy. While some examples thereof exist, such as Kuijper et al.'s [KWKJ19] *DynoVis* framework for the visualization of dose-over-time effects in biological networks, Perkins and Daniels' [PD17] efforts to visualize dynamic gene expression data, or Hartman and Schreiber's [HS15] work on visualizing metabolic models.

We see ample opportunity for future biological and biochemical network visualization research to embrace the challenge of making sense of time-dependent evolution dynamics. Taking inspiration from the work of Beck et al. [BBDW17], a multitude of possible approaches present themselves, grouped within animation and timeline-based techniques, i.e., animated and static representations, respectively. Developers and researchers of biological network visualization tools could take inspiration from existing approaches, such as Bach et al. [BHRD+15] *Small Multiples* approach, which opts to visualize dynamic networks as an interactive stack of adjacency matrices, or Peng et al.'s [PTLZ18] *DMNEVIS* framework, which opts to split the various aspects of dynamic networks into separate views in a larger interactive dashboard.

Biological networks, depending on the particular context, bring with them very particular challenges that require careful consideration. This includes, for example, multivariate (and dynamic) attributes attached to both nodes and edges [NMSL19], such as gene expression fold changes [LJT21], labels of metabolites and reactions [OSA+21], or the database origin and type of a particular protein-protein interaction [DMF17]. Many possible solutions to this well-known problem have been put forth [KSP23]. Additionally, within the context of multi-omics data integration and analysis (Section 2.9.1), a domain expert may be faced with so-called multilayer networks [MGM+19], in which a network is comprised of multiple, separate inhomogeneous omics networks [SVK+20, HSL17, HdDTMM+18]. Tackling the visualization of dynamic biological networks will require close collaboration between network scientists, bioinformaticians, and domain experts alike to ensure such networks are visualized in a user and task-oriented manner.

2.11 Conclusion

Based on Card et al.'s [Car99] visualization pipeline, outlined and summarized the various components that make up biological network visualization, i.e., i) its underlying *Graph Models*, ii) the concrete *Graph Analyses* that a user might perform, iii) the various *Graph Models* approaches, and iv) the abstract *Graph Analysis Tasks*, which in turn motivate v) the provided methods of *Human-Computer Interaction*, which are motivated by vi) particular *Application Areas*. In order to understand the current state of the field, we collected frequently used graph analysis approaches, visualization techniques, abstract graph analysis tasks, and interaction strategies from *a large body of systematically collected literature Tasks*. Based on this categorization of literature, we identify *six outstanding challenges* in the field of biological network visualization, which outline

1. the opportunity to move beyond schematic and straight-line node-link diagrams in order to embrace powerful alternatives that exist,
2. the challenge of incorporating uncertainty in biological network visualization frameworks and tools, be they measurement, statistical, or inherent types of uncertainty,
3. the potential of incorporating sophisticated graph analysis techniques that go beyond straightforward descriptive metrics,
4. the gap in (visual) network comparison tools, which allow users to effectively analyze the differences and similarities of networks across time points, experimental conditions, or samples,
5. the possibilities of keeping track of provenance and user trust to better communicate how certain findings have been reached, and finally
6. the still relatively unexplored topic of dynamic biological network visualization, which visually describes the temporal topological evolution of biological networks.

This introduction to biological network visualization and our identification of outstanding challenges should, however, not be viewed as exhaustive. Several topics could not be covered or fell outside of the scope of this chapter, such as community detection algorithms and their application to biological networks [CK21]. Additionally, the scalability of the various visualization and analysis approaches would justify a follow-up survey in and of itself.

CHAPTER 3

Untangling the Hairball: Principled Vertex Splitting for Iterative Edge-Crossing Resolution

The number of nodes and edges in modern biological graphs makes their visualization especially difficult. Such visualization efforts often result in poorly readable networks, which makes their visual analysis and exploration challenging. The readability of such networks can be measured in a variety of ways, but one particular approach is through the use of so-called quantitative graph aesthetic metrics, e.g., the orthogonality of nodes and edges, the number of edge bends, or the minimum incident edge angle ratios. One such metric of particular importance, however, is the *number of edge crossings*: the more edge crossings, the more challenging it is to read a network visualization. Conventional (graph theoretical) approaches to edge crossing resolution, such as NP-complete edge-crossing-minimizing graph drawing algorithms or (NP-hard) minimal node/edge deletion approaches, are neither applicable to modern biological problems nor desirable to domain experts, respectively.

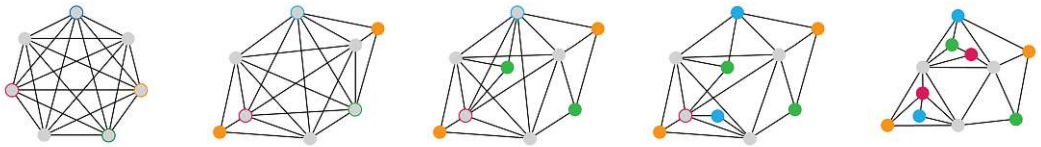
Here, we make the first steps towards investigating *iterative algorithmic vertex splitting* as a novel approach to resolving such edge crossings and improving the overall readability of network visualizations without outright removing nodes or edges from the drawing. However, as an excessive number of split vertices are hypothesized to also negatively contribute to graph readability, we investigate the effect of vertex splitting in small-scale graphs in a crowd-sourced user study. The contents of this chapter are based on the full paper *"Improving Readability of Static, Straight-Line Graph Drawings: Towards a Systematic Approach of Edge Crossing Resolution through Iterative Vertex Splitting"* [EVRW23], published in the journal *Computer & Graphics*, with myself as first author, in collaboration with Anaïs Villedieu, Renata Raidou, and Hsiang-Yun Wu.

3.1 Overview

We present a novel vertex-splitting approach with which to iteratively resolve edge crossings in order to improve the readability of graph drawings. Dense graphs, even when small in size (10 to 15 nodes in size), quickly become difficult to read with increasing numbers of edges, and form so-called "*hairballs*". The readability of a graph drawing is measured using many different quantitative aesthetic metrics. One such metric of particular importance is the number of edge crossings. Classical approaches to improving readability, such as the minimization of the number of edge crossings, focus on providing overviews of the input graph by aggregating or sampling vertices and/or edges. However, this simplification of the graph drawing does not allow for detailed views into the data, as not all vertices or edges are rendered, and also requires sophisticated interaction approaches to perform well. To avoid this, our locally optimal vertex splitting approach aims to minimize the number of remaining edge crossings while also minimizing the number of vertices that need to be split. In each iteration, we identify the vertex contributing the largest number of edge crossings, remove it, locate the embedding locations of said vertex's two split copies, and determine each copy's unique adjacency. We conducted a user study with 52 participants to evaluate whether vertex splitting affects users' abilities to conduct a set of graph analytical tasks on graphs 12 nodes in size. Users were tasked with identifying a vertex's adjacency, determining the shared neighbors of two vertices, and checking the validity of a set of paths. We ultimately conclude that within the context of small, dense graphs, systematic vertex splitting is preferred by participants and even positively impacts user performance, though at the cost of the time taken per task.

3.2 Introduction

From social media to multi-omics interaction analyses, modern network visualization deals with increasingly large and dense (sub-)networks [KKPEP20]. Visual analysis efforts of such networks focus on understanding not only the importance of individual entities, such as genes or persons, but also their in-between relationships, such as biochemical interaction types or social relationship statuses. These networks are most commonly visualized as straight-line node-link diagrams, which illustrate entities as points of potentially different shapes or colors, and edges as straight lines connecting them (Figure 3.1 (a)). If not drawn by hand, such graphs are commonly laid out using force-directed layout algorithms, owing to their implementations' availability and computational tractability, such as (extensions of) Eades [Ead84], Fruchterman-Reingold [FR91], or Kamada-Kawai [KK89]. These techniques optimize the graph's drawing in terms of physically modeled stress or cost functions. While some research effort has been made to incorporate one [BFG+21, SOGB10] or multiple [HEHL13, ADLD+20] graph aesthetic metrics into force-directed algorithms, the majority do not. Subsequently, most readily available force-directed algorithm implementations do not scale well visually to dense and/or larger graphs, producing hard-to-read, or even completely unintelligible,



(a) No Splits (b) 1 Split (c) 2 Splits (d) 3 Splits (e) 4 Splits

Figure 3.1: A $k = 7$ complete graph, initially laid out using Kamada-Kawai (a), is iteratively split by hand, one vertex at a time, to resolve edge crossings (b-d) until all are resolved (e). To do so, four nodes are split. Before being split, these nodes are visualized as grey circles with a colored border around them. After splitting, they are shown as a correspondingly colored circle. It should be noted that, while the initial embedding (a) may be difficult to read, so is the final embedding (e). In (e) specifically, while no edge crossings remain, the number of split vertices complicates the reading of the graph's and individual node's topologies.

drawings [Kob12]; so-called "hairballs" [PSM+11, SHW+20]. Ultimately, such drawings do not allow for the effective and/or efficient visual analysis of the underlying graph data [YDK+18]. To handle "hairball" effects in larger graphs, existing work has focused largely on interactive graph summarization [LSDK18], i.e., presenting graphs at different levels of detail. By introducing such level-of-detail hierarchies, summarization unavoidably manipulates graph topology, thereby altering the perceived relationships within the graph and potentially confusing the mental map created by the user [MELS95, AP13]. For completeness' sake, it should be mentioned that recent efforts have been made to create such level-of-detail hierarchies *without* altering the graph's topology [GLA+23].

3.2.1 The Relevance of Small, Dense Graphs

While graph data have become larger, the analysis and visualization of small, dense graphs remain important challenges, as domain experts are not only interested in a graph's topology on a global but on a local level. Hence, many interactive techniques and tools, in line with Shneiderman's "*Overview first, details on demand*" mantra [Shn96], aim to provide a summarized view of a network's topology from which a subgraph of interest can then be investigated in more detail.

For example, the Kyoto Encyclopedia of Genes and Genomes (*KEGG*), currently has over 40,000,000 genes recorded [KFS+22, Kan19, KG00]. Research has so far not attempted to understand or navigate an organism's complete network of genes, metabolites, and chemicals. Instead, work focuses largely on visualizing and analyzing subgraphs, i.e., pathways, or collections of such subgraphs [MLM+17, AKY+05b]. Outside of the context of *KEGG* pathways, similar *Focus+Context* visualizations of small subgraphs can be found in, for example, general systems biological network [WZC+15, JSWD09] or social network visualization [JGH11], just to name a few. Thus, as visualization and visual analysis of small, dense graphs remain relevant to domain experts, so does the search for aesthetically optimized embeddings of such graphs.

3.2.2 Existing Methods to Edge Crossings Resolution

Within the context of smaller graphs ($10 \leq n \leq 15$), Purchase et al. [PCA02, Pur00, Pur02] extensively studied the importance of edge crossings on user preference and performance. These findings were corroborated more recently by Kouburov et al. [KPS14], who found that larger numbers of edge crossings indeed (statistically significantly) negatively impact the accuracy and efficiency of user performance in smaller graphs ($n = 40$). Unfortunately, finding some optimal embedding that produces the minimum (edge) crossing number k [CMS11] is, even for highly restricted graph classes [GJ83], NP-complete [KPS14, Hli06]. Thus, beyond "simply" determining whether a planar k-crossing graph exists or what a graph's minimal crossing number k may be, extensive work has focused on finding means of eliminating edge-crossings from non-planar graphs [OT81, TJS86]. One such example involves computing the minimal sets of edges [GT94] or vertices [JLS13, JdKW21] that must be deleted to resolve all edge crossings. While this elimination of edge crossing through the removal of some minimal set of edges or vertices may be an interesting solution from a graph-theoretic perspective, it is not a useful paradigm for domain experts. In many biological and biochemical application domains, understanding the relationships between, as well as the attributes of, *all* entities of interest is crucial, making the outright removal of entities or their relationships not practical.

3.2.3 Approach and Contribution

In this chapter, we propose an approach to reduce graph complexity to improve the readability of graphs *without removing relationship or entity information* that is *agnostic of the graph drawing method chosen*. Our approach is centered on *splitting* "problematic" vertices, i.e., vertices whose edges are involved in large numbers of edge crossings, in a graph's drawing. As noted by Henry et al. [HBF08], a set of terms have been used interchangeably to describe such approaches, such as *duplicating*, *cloning*, *aliasing*, or *mirroring*. Here, we keep to the term *vertex splitting*, common to the graph drawing community. Vertex splitting involves identifying a vertex v, removing it from the drawing, and replacing it with two non-adjacent copies of v_1 and v_2, while the original edges of vertex v are distributed across the two split copies [NST+22]. As shown in Figure 3.1,

the orange-circled vertex in (a) is selected, split in two, and each copy visualized as an orange-colored vertex (b). Importantly, the adjacency of the original vertex is now shared across both copies.

Given some initial graph embedding, this splitting could be achieved in a principled manner to optimize graph aesthetic criteria [EMW86]. As discussed previously, while many different aesthetic criteria influence a graph's readability [Pur02], overwhelmingly, the *number of edge crossings* has been shown to negatively affect user performance and preference. This is especially true for smaller graphs, ranging in size from 10 to 20 vertices, [Pur00, PCA02, Pur97]. However, vertex splitting approaches featured in literature are not directly driven by graph aesthetic metrics, but an *a-priori* selection of "problematic" vertices [ERG02, LPP+06a], or the simple *complete splitting of selected vertices* based on groups/clusters featured within the data [HBF08, RD10, WNSV19]. To the best of our knowledge, there exists no principled approach to vertex splitting that aims to improve graph aesthetic criteria.

The *contribution* of our work is the development, assessment, and discussion of a novel, iterative approach to vertex splitting for static, straight-line graph drawings. Specifically, our approach i) minimizes the total number of edge crossings in the embedding, given a selection of embedding faces, which ii) maximizes the number of edge-crossing-free connections, while iii) splitting the minimal number of vertices necessary. We achieve this by discretizing the graph drawing into edge-crossing-equivalent polygons, i.e., polygons within which all points induce the same number of edge-crossings when connected to any vertex in some graph drawing, thereby simplifying the search for a split vertex copy's placement within this embedding. To investigate the effect of our vertex splitting on the readability of small graphs — here, 12 nodes in size — for which the importance of edge crossings has been thoroughly documented [Pur00, PCA02, Pur97]), we conduct a user study with 52 participants. The outcomes indicate that our approach has the potential to improve readability and aesthetic quality in static, straight-line graphs.

3.3 Related Work

Many aesthetic criteria affect a user's ability to effectively navigate and analyze a graph drawing [Pur02]. A criterion of particular importance is the number of edge crossings, which has been found to especially affect readability in small- and intermediate-sized graphs [Pur00]. In addition to drawing criteria, the size, density, and topological structure of a graph will also influence the layout quality [YAD+18]. In this section, we summarize previous works related to our proposed approach.

Graph Drawing and Graph Aesthetics Heuristic-based graph drawing, such as force or energy-based approaches, often produces difficult-to-read embeddings with poor graph aesthetic criteria, as they optimize these criteria only indirectly [KPS14]. However, work has also been done on rendering more readable graphs by drawing them directly as a function of these aesthetic metrics. As these graph aesthetics often conflict with each other

[WYHS21], most such approaches aim at satisfying either a single or a *balance* of multiple such criteria [DLR11]. Indeed, the inclusion of one [DH96, BFG+21, SOGB10, DLR11] or multiple [HEHL13, ADLD+20, KDT14] such aesthetic criteria for classical graph drawing approaches has been discussed for a while. Additionally, multi-objective neural-network-based approaches to graph embedding have also found popularity [TCG24, WYHS21]. However, owing to their complexity and novelty, most approaches are simply not readily available.

Reducing Graph Complexity for Visualization As the graph density increases, theoretical graph drawing techniques can only handle limited graph classes [DLM19], which still cannot cover all types of real-world graphs. To solve the hairball problem for practical analysis purposes, several approaches have been developed. One common strategy called *summarization* aims to abstract a graph through vertex or subgraph aggregation, as summarized in the survey by Liu et al. [LSDK18]. This strategy includes *vertex aggregation*, where clustering that produces a data hierarchy is often performed for interactive data navigation [EF10]. Another technique, *edge aggregation*, such as edge compression [DHRMM13], edge concentration [OKSK16], and edge bundling [EHP+11, BRH+17] manipulate edge topology and geometry for better visual quality. The two ideas can also be adapted simultaneously. Dwyer et al. optimized power graphs to achieve lossless vertex and edge aggregation, which facilitates following paths in the graphs [DMM+14]. However, the problem has been proven to be NP-hard, and no scalable technique is readily available. *Simplification* is another type of summarization, which deletes edges [GT94] or vertices [JLS13, JdKW21]. In addition to deleting vertices, OntoVis [ZKER06] simplifies graph topology by also removing duplicated paths. More recently, *graph sampling* has been proposed to approximate large graphs for visualization purposes [NHEM17, HCH+21]. The aforementioned approaches either remove data or require interaction [YAD+18], which is sometimes not preferred for specific user tasks. Other techniques that use spanning trees and graphs as backbones [VHW08, LHLW19] provided a partial solution, yet the complexity of edge rendering persists. *Toroidal wrappings* allow for vertices, and thereby their edges, to be spread across a (2D) torus topology. This allows for links to wrap across the rectangular view boundary, thereby reducing the number of edge crossings had the network been embedded on a 2D surface [CDMB20]. Such embeddings not only improved the network's aesthetic metrics but also were shown to assist in user understanding [CDBM21]. However, this comes at the cost of both time taken and accuracy, especially for path tracing tasks [CDBM21, CDMB20].

Vertex Splitting in Applications Vertex splitting is not a brand new idea: several overview maps of pathway databases (ReconMap [NDG+17] and KEGG map [KFS+22]) incorporate vertex splitting to reduce the visual complexity of pathway diagrams. However, these diagrams are often designed and curated *manually* by illustrators. Due to this common usage of vertex splitting, the idea has been investigated in several applied papers. For example, to produce a fully planar drawing, the ontology tool OntoRama [ERG02] and the gene ontology tool TreePlus [LPP+06a] allowed for vertices to be cloned. Cross-linked vertices present in multiple branches are simply fully duplicated to produce a

more readable, tree-style layout. Beyond tree graphs, Wu et al. [WNSV19] incorporate different duplication schemes based on vertex properties and extend the idea on multi-level clustered graphs [WNV22]. Lastly, Nielsen et al. [NOM+21] developed a learning-based approach to determine which vertices should be duplicated based on human-curated graph drawings featuring (iteratively) split vertices. However, these approaches do not necessarily tackle the challenge of readability. Complete splitting [LPP+06a, ERG02] may produce too many copies of the same vertex, thereby hindering readability in and of itself; as discussed by Henry et al. [HBF08]. On the other hand, splitting a vertex once per group assignment [WNSV19] does not necessarily ensure the readability of each group individually. Such split vertices may still be responsible for large numbers of edge crossings or be overly connected to other vertices within the group. Splitting these vertices again could further improve selected graph aesthetic criteria. Nonetheless, these approaches are strongly bound with specific applications and cannot be easily extended to more generic cases.

Vertex Splitting in Graph Drawing and Visualization Beyond the examples learned from application domains, previous research also tackled more general solutions. The spring-based layout algorithm of Eades and Mendoça [EdMN96] aims to split vertices based on the "tension" of adjacent edges, to relax edge springs in the embedding. They demonstrate that even their heuristic approach allows for the resolution of edge crossings through vertex splitting. Henry et al. investigated NodeTrix [HBF08], a hybrid visualization integrating node-link diagrams and matrix representations. This work researches the users' abilities on analytical tasks pertaining to relationships, and our project is primarily inspired by the positive findings of these two publications. Additionally, Lambert et al. [LDB11] and Rohrschneider et al. [RHR+10] propose approaches to duplicating vertices within and across metabolic pathways. Additionally, vertex duplication has not only been used for graph visualization but is also applied to improve the legibility of Euler diagrams [RD10]. Lastly, Nöllenburg et al. [NST+22] study the vertex splitting problem from a theoretical perspective; given some input drawing, they find that determining the smallest set of splits to produce a planar drawing is NP-complete.

In summary, to the best of our knowledge, the effectiveness of vertex splitting has not yet been fully investigated in common visualization scenarios, although it has been used in some practical applications. In this chapter, we aim to develop and propose a vertex-splitting approach to iteratively resolve edge crossings with the purpose of improving the readability of graph drawings. Our approach can be applied to network visual analytics by assisting users to navigate and read graphs, while also retaining local algorithmic guarantees.

3.4 Desiderata

Many different quantitative aesthetic metrics can negatively impact the readability of a graph drawing [Pur02]. As we are interested in improving these metrics in a given embedding, any one or multiple metric(s) could form the basis of our vertex-splitting approach. However, as the edge crossing criterion [GJ83] forms arguably the most important metric, especially for smaller graphs, we opt to focus on resolving edge crossings [NST+22] specifically. Moreover, Eades and Mendoça's heuristic approach to tension reduction showed that splitting can indeed reduce the number of edge crossings [MSX+02]. However, as highlighted by Henry et al. [HBF08], splitting a vertex too often or splitting too many vertices can itself have an undesirable impact on readability (Figure 3.1 (e)). Ultimately, we wish to resolve as many edge crossings as possible, while splitting as few vertices as necessary. To describe our goals more concretely, we define three desiderata which our approach aims to locally guarantee, namely:

(D1) Minimize the Number of Vertices Split As discussed by Henry et al. [HBF08, HFM07], too many split vertices, or a vertex split too often, can negatively impact a user's ability to parse a graph's drawing. Thus, we aim to resolve as many edge crossings as possible, while locally minimizing the number of vertices to split. That is to say, we only ever split a vertex in two, i.e., the smallest split possible. Looking at Figure 3.2 (D1), the orange embedding shows how edge crossings can be resolved by simply splitting a target vertex completely, i.e., splitting it once for every one of its edges. Such approaches result in a set of split copies of degree $d(v) = 1$ and add unnecessary visual clutter. Instead, we thus opt to only ever split a vertex in two, as seen in red, to ensure that splits minimally affect readability.

(D2) Maximize the number of edge-crossing-free connections Embedding a vertex in a graph drawing that minimizes the overall number of edge crossings can be done in polynomial time. However, currently available methods are not practical for real-world applications owing to the amount of time needed [SAHB11]. Thus, we first employ a straightforward heuristic for the subsequent, and more general, desideratum **D3**: identifying an embedding location that minimizes the vertex's number of edge-crossing-free edges to its neighbors. In essence, we must first identify the *face* which is incident to the maximal number of neighbors of our vertex. For example, in Figure 3.2 (D2), the red embedding is chosen over any of the orange ones, as the former allows for 3 edge-crossing-free connections, compared to any of the latter's 2.

(D3) Minimize the Drawing's Total Edge Crossings While **D2** allows us to identify a face f within which to embed a vertex v, the vertices incident to f will most likely not encompass all of v's neighbors. Connecting to the remaining neighbors of v not incident to f will thus induce edge crossings, which will negatively impact readability [KPS14, Pur00, PCA02, Pur97]. Thus, within f, we must identify where to place this vertex, such that the number of induced edge crossings is minimized. In Figure 3.2 (D3), the red embedding is selected, as it only indicates 1 edge crossings, compared to the 2 induced by the orange drawing.

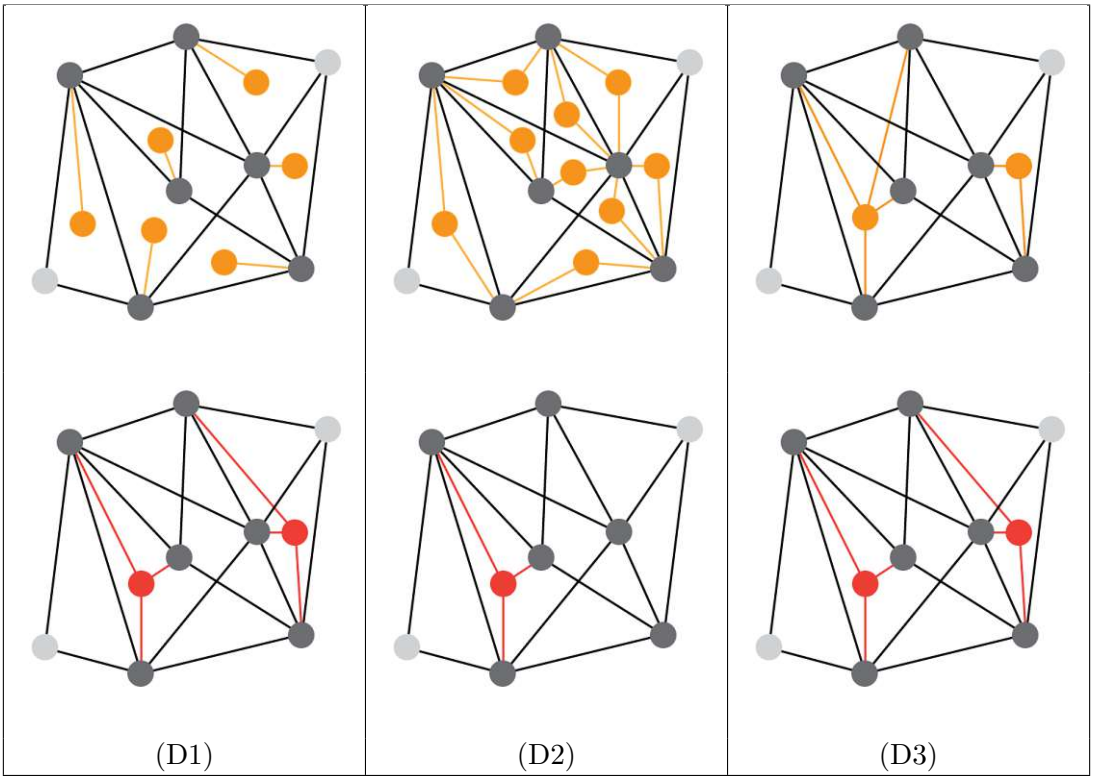
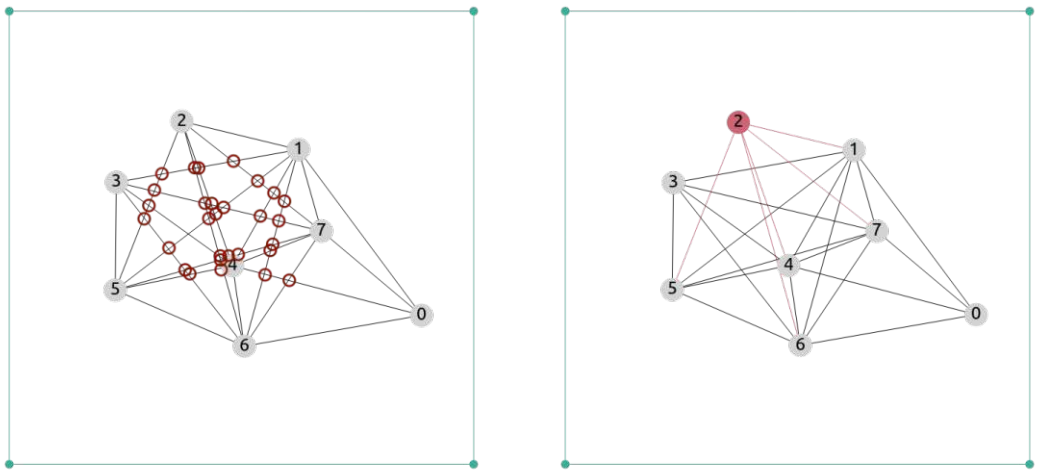


Figure 3.2: A visual example of the three desiderata. **(D1)** visualized the *minimization of split vertices* desideratum. Instead of resolving all edge crossings possible by simply splitting a vertex completely (as shown in orange), we instead only ever split a vertex in two, as this is the smallest split possible. **(D2)** showcases the *maximization of edge crossing free connections* desideratum. Assume we aim to insert a single vertex into an existing embedding, adjacent to a set of given set of neighbors (shown in dark grey). Instead of embedding said vertex in any of the orange positions, which all only allow for 2 edge-crossing-free connections, we embed this vertex as shown in red, where 3 adjacent vertices can be reached edge-crossing-free. Lastly, **(D3)** illustrates the *minimization of edge crossings* desideratum. Given two split copies of a vertex embedded within two faces that fulfill D1, we now must find the adjacencies of each copy that minimize the total number of induced edge crossings. Here, the orange set of vertices and edges is discarded, in favor of the red one, as the former induces 2 edge crossings, compared to the latter's 1.

3.5 Vertex Splitting Algorithm

Here, we first provide a high-level description of our algorithm, visualized in Figures 3.3-3.6 (a). Let $G = (V, E)$ be our input graph where $V = \{v_1, v_2, ..., v_n\}$ and $E = \{e_1, e_2, ..., e_m\}$. We additionally consider a two-dimensional, straight-line drawing D of G as input. Our algorithm functions in three steps:



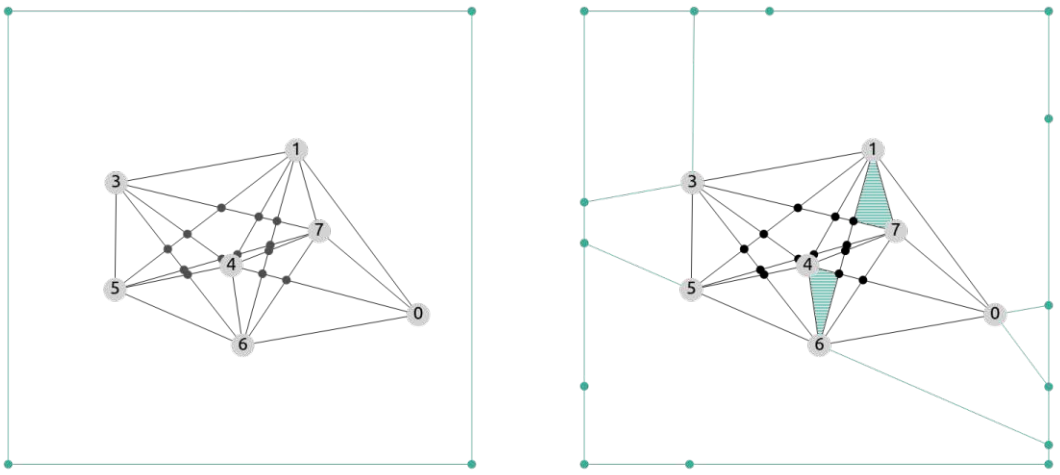
(a) Count Edge Crossings (b) Select Target Vertex

Figure 3.3: *Step 1: Target Vertex Selection.* In order to ultimately detect the vertex to split, we first count all edge crossings present in the provided graph embedding (a), where edge crossings are circled in red. For each edge crossing, the two involved edges' incident sets of vertices have their crossing numbers incremented. In order to resolve as many edge crossings as possible, the vertex with the highest crossing number is selected as the target to be split (b), here vertex 2. The canvas border is shown in blue ( ).

Step 1: Target Vertex Selection First, the target vertex to be split must be identified. To do so, all edge crossings in the given embedding are counted, each mapped to the edges' incident vertices (Figure 3.3 (a)). The vertex whose incident edges are involved in the largest number of edge crossings is selected for splitting (Figure 3.3 (b)).

Step 2: Face Decomposition and Selection After removing the selected target vertex from the embedding, in line with desideratum **D1**, we select only a pair of faces to embed each of the two copies of the removed vertex. Moreover, we must also select a subset of the copies' neighborhoods, namely the neighbors who will not induce any crossings in the input drawing Figure 3.4 (b), as discussed in desideratum **D2**. Note that we only split a vertex into two at each iterative step. To do so, we start by computing a planarization of D to identify a set of faces $F = \{f_1, f_2, \ldots, f_k\}$ (Figure 3.4 (a)) that partition the canvas disjointly. When placing a vertex v in a non-convex face f, there is no guarantee that we will be able to draw crossing-free edges between v and the vertices incident to f. Thus, we compute a set of *sight cells* $S = \{s_1, s_2, ..., s_r\}$, which are convex polygons that partition non-convex faces (Figure 3.4 (b)). Note that an already convex face is its own sight cell.

Step 3: Subface Decomposition and Selection Given two selected sight cells, we must now decide where within them to place each split copy of the selected vertex, such

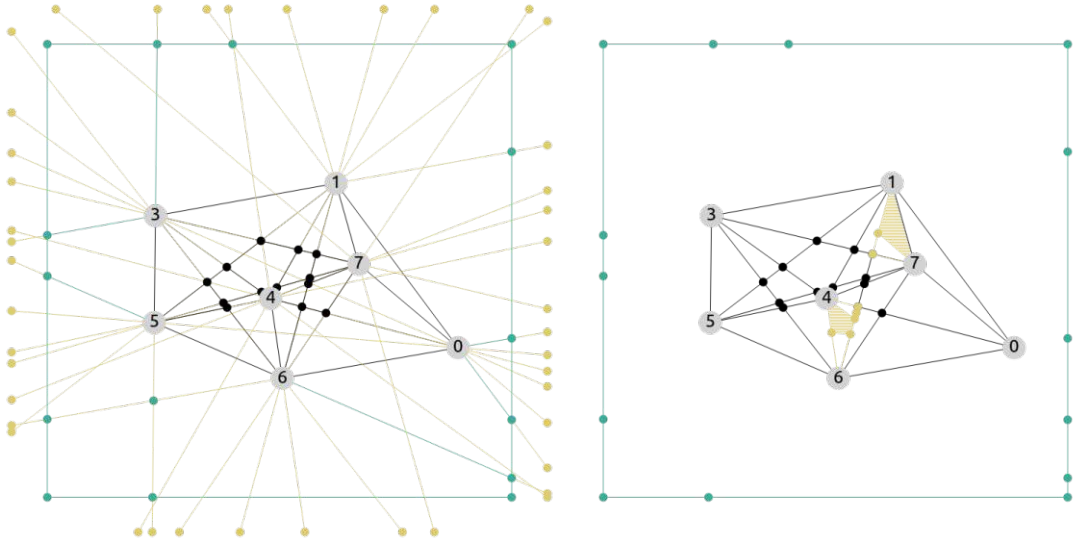


(a) Target Vertex Removal and Face Detection (b) Sight Cell Decomposition and Selection

Figure 3.4: *Step 2: Face Decomposition and Selection.* The selected target vertex, as well as its incident edges, are removed from the embedding, and the remaining graph is planarized (a) using Planarization vertices, illustrated as smaller, unlabeled, dark grey circles (•). All faces in the embedding can now be identified. Given desideratum *D2*, we aim to maximize the number of edge-crossing-free connections when selecting the embedding faces. However, as not all inner faces (as well as the outer face) are convex, they must be decomposed into convex sight-cells polygons (b). This is achieved through line projection and the placement of tiling vertices and tiling edges within and until the canvas border, both shown in blue (•). The selected sight cells are indicated as turquoise-color surfaces. As we are trying to achieve this with minimal numbers of split vertices (D1), only two faces, i.e., splits, are selected.

that the number of induced edge crossings is minimized. To achieve this, sight cells are further decomposed into *subfaces*; convex polygons that each form an equivalence class with regard to crossings (Figure 3.5 (a)). More precisely, for a region $b \in B$, any embedding of a vertex v within b will induce the same amount of crossings. Instead of infinitely many possible embedding locations with a selected sight cell, we can consider only a discrete set of subfaces within it. Now, in line with desideratum **D3**, we can select a region for each of the selected sight cells such that the smallest number of additional edge crossings are induced when connecting them to vertices non-incident to the selected (sub-)face (Figure 3.5 (b)).

Step 4: Embedding of the Split Vertex Copies With two subfaces selected and the edge-crossing-minimizing adjacency of each subface determined, the final step is the simple embedding of the two split vertices and the drawing of the edges between their neighbors. Again, only two split copies are created to adhere to desideratum **D1**



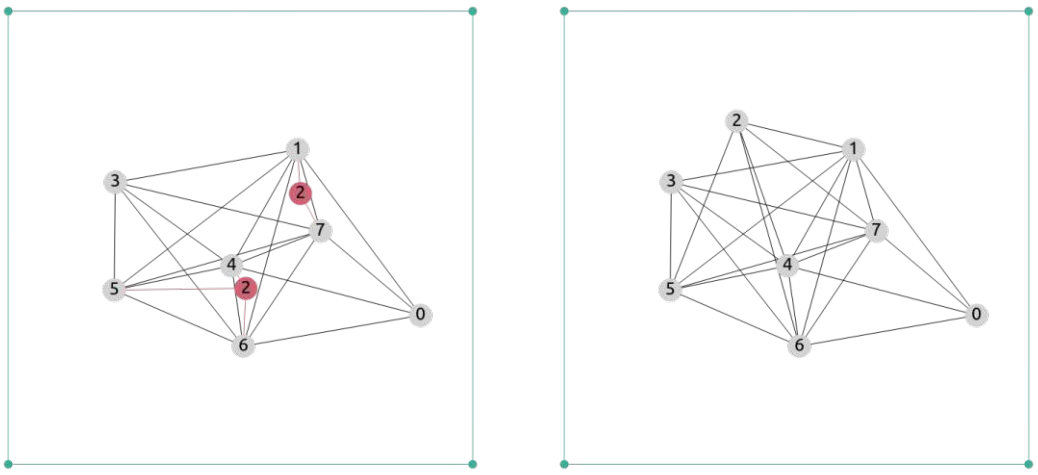
(a) Line Segmentation of Embedding (b) Subface Decomposition and Selection

Figure 3.5: *Step 3: Subface Decomposition and Selection.* In line with desideratum D2, given a minimization of the number of edge-crossing-free connections, we aim to minimize the total number of edge crossings. To identify regions within the embedding that are equivalent in terms of their number of induced edge crossings, a complete line arrangement drawing of all pairs of real vertices is drawn, bounded by user-defined embedding limits (a). As we are interested only in placing vertices within the two selected target faces (Figure 3.4), we use these line segments to tile only them into sight cells in order to identify the pair of sight cells which induce the fewest number of edge crossings possible when connected to the original target vertex's neighbors (b). The newly placed tiling vertices and tiling edges are drawn in brown (•), as are the highlighted selected subfaces. The canvas border is shown in blue (•).

(Figure 3.6 (a)). Similar to Riche and Dwyer [RD10], in the interest of preserving the user's *Gestalt-theoretic* [Kof35] mental map of the original graph embedding (Figure 3.6 (b)) as much as possible, the graph is not redrawn using spring embedding.

3.5.1 Step 1: Target Vertex Selection

The first step of the algorithm involves the identification of the vertex to be split, based on the number of edge crossings in the embedding (Figure 3.3 (a)) The vertex involved in the largest number of edge crossings must be identified (Figure 3.3 (b)). By selecting the vertex with the largest number of edge crossings, we aim to thereby also resolve the largest number of edge crossings associated with other vertices, in line with desideratum **D3**. Specifically, for each $e \in E$, let $cr(e)$ denote the number of edges that cross e. For $v \in V$, let $cr_n(v)$ denote the sum of $cr(e_v)$ where $e_v \subseteq E$ are edges incident to v, i.e., $cr_n(v) = \sum cr(e_v)$. All vertices are then ranked by their number of edge crossings, i.e.,



(a) Embed Split Vertices (b) Original Unsplit Graph

Figure 3.6: *Step 4: Embedding of the Split Vertex Copies.* We now place the two split copies at the centroids of the selected sight cells and connect them to all neighbors vertices incident to their selected face (Figure 3.3), as well as all other neighbors in accordance with their selected sight cell (Figure 3.4 (a). Note the "suboptimal" placement caused by separating out desiderata **D2** and **D3**; placing it to the left of the edge connecting vertices 4 and 6 would have resolved an additional edge crossing. The original, unsplit graph is presented for the sake of comparison (b). The canvas border is shown in blue (•).

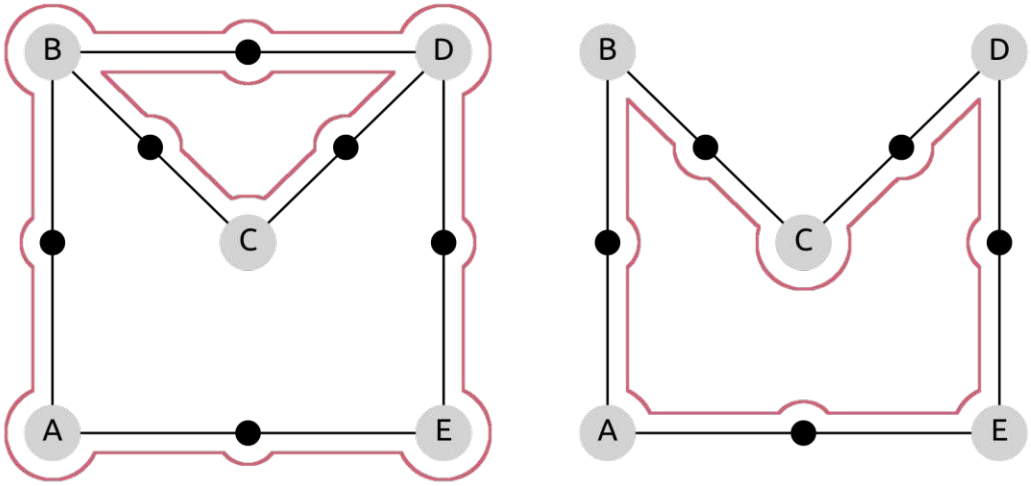
$cr_n(v)$. The vertex with the largest crossing number is selected as the target vertex to be split, its adjacency is recorded, and subsequently removed from the drawing D. Ties in the vertices' edge crossing numbers are resolved based on their degree. To illustrate, consider the example graph $G(V, E)$ of size $|V| = 8$ shown in Figure 3.3; (a) highlights all edge crossings which ultimately leads to the selection of vertex 2, as its crossing number, $cr(v_2) = 18$, is the largest.

3.5.2 Step 2: Face Decomposition and Selection

Desideratum **D2** states that we aim to not only minimize the number of edge crossings in the embedding but also maximize the number of edge crossing-free connections. Thus, when selecting a potential embedding location for a copy, we first identify a face whose incident vertices are part of the split vertex's neighborhood. The aim of the second step of our algorithm is to select two convex regions that jointly maximize the number of edge-crossing-free connections to the split vertex's neighborhood. First, we consider D_P, a planarization of D, where each line crossing induces a vertex in the graph. We obtain it by placing a vertex called a *planarization vertex* on each crossing and replacing the crossed

edges by paths through that new vertex (Figure 3.4 (a)). Since D_P is a planar graph we can find its faces: $F = \{f_1, f_2, \ldots, f_k\}$. However, as there is no guarantee that the faces obtained after the planarization are all convex, we must further decompose the faces into convex polygons to ensure incident vertices can indeed be reached edge-crossing-free (Figure 3.4(b)). Let $F_C \subseteq F$ denote the set of faces which are convex, and $F_{NC} \subseteq F$ the set of faces which are not convex, such that $F_C \cap F_{NC} = \emptyset$ and $F_C \cup F_{NC} = F$. Returning to the previous example, Figure 3.4 (a) showcases the planarization of the original graph with the selected target vertex removed, and (b) illustrates both the decomposition of the outer face into sight cells and the selection of two subfaces which jointly maximize the number of edge-crossing-free connections.

Face Detection and Selection To compute F, we adapt the graph minimum cycle basis algorithm by Kavitha et al. [KMMP08]. For a Graph G with non-negative edge weights, they define a *cycle basis* as a maximal set of linearly independent cycles. Correspondingly, the *minimum cycle basis* is the cycle basis whose sum of edge weights is minimal. However, the minimum cycle basis only extracts cycles topologically and does



(a) (b)

Figure 3.7: Given in (a) is a graph with vertices $\{A, B, C, D, E\}$. *Cycle vertices*, shown as smaller, unlabeled circles, are placed at the center of all edges. The minimum cycle basis is identified; here (A, E, D, B, A) and (C, D, B). Cycle (C, D, B) is a valid face. Cycle (A, E, D, B, A) is not, as it contains vertex C. Thus, the cycles' subgraph is extracted. Edge E_{BD} is mapped to one valid face (C, D, B) and is also part of the graph's outer face, so it is removed from the subgraph. The minimum cycle basis is again identified for the pruned subgraph (b). The only identified cycle, (A, E, D, C, B, A), is now also a valid face.

not necessarily correspond to a graph's set of faces geometrically. Thus, two additional steps have been incorporated to identify drawing D_P's faces (Figure 3.7).

First, we must ensure that not just all vertices, but *all edges are traversed* as well. To do so, we place a *cycle vertex* (shown as • in Figure 3.7 (a)) at the center of each edge in D_P, and replace this edge with a path through the newly placed vertex. Second, we must ensure that *all identified cycles are indeed faces*, i.e., that each cycle does not contain any vertices or edges within it. If an identified cycle does not contain any vertices within itself, it is deemed a valid face. Otherwise, its subgraph, defined by the cycle in question as well as all vertices and edges within it, needs to be extracted as shown in Figure 3.7 (b). From this subgraph, edges are removed that can be mapped to two valid faces, or if it is part of the cycle in question and already mapped to one valid face. The minimum cycle basis is now again identified within this subgraph, and the face-check is recursively repeated.

Lastly, to fulfill desiderata **D1** and **D2**, we need to select the *two* faces whose set of incident vertices, which are part of the target vertex's adjacency, is the largest. Ties are resolved using each edge's absolute Euclidean difference from its Kamada-Kawai's graph-theoretic optimal length. Figure 3.4 (b) showcases the selection of target embedding faces. The selected target vertex 2 's neighbors were vertices v_1, v_4, v_5, v_6, v_7. By selecting the blue highlighted faces, namely { 1 , 7 , •} and { 4 , 6 , •}, 4 or the 5 target neighbors can be reached edge-crossing free.

Decomposing Non-Convex Faces In accordance with desideratum **D2**, we aim to connect our copies to a face's incident vertices without inducing edge crossings. For convex faces, this can be straightforwardly achieved. For non-convex-shaped faces, however, this is not necessarily possible. Here, to ensure that a non-convex face's incidence can be realized as edge-crossing-free connections, the connections must be decomposed into convex polygons, which we call *sight cells*. We denote our sight cells as $S = \{s_1, s_2, ..., s_r\}$. For a non-convex face $f_k \in F_{NC}$, let its sight cells be denoted as $S_k \subseteq S$. These sight cells are created by placing additional *tiling vertices* V_t and *tiling edges* E_t to D_P (Figure 3.8). Note that a convex face is its own sight cell.

First, given a non-convex face $f \in F_{NC}$, we identify at least one vertex v incident on f whose incident edges form an inner angle greater than 180°. For each edge pair e_{uv} and e_{vw}, two rays are projected from u and w respectively, through v (thus, inside f). We compute the intersection of each ray and the boundary of f. At the location of the intersection, a new tiling vertex $t \in V_T$ is placed, and the edge intersected is replaced by a path through t. An additional tiling edge connecting vertex v and t is added. For example, in Figure 3.8, a hypothetical, non-convex face $\{A, B, C, D\}$ is decomposed along the rays projected from edges e_{AB} and e_{CB} as they form an angle greater than 180° around vertex B , resulting in sight cells (B, C, X, B), (A, B, Y, A), and (B, X, D, Y, B). These newly placed edges define the bounds of a vertex's "line of sight", i.e., a vertex can be connected edge-crossing-free from one side of this edge, but not the other. This is repeated for the second edge incident to v, as well as all other vertices incident to

f whose incident edges form angles greater than 180°. The subgraph defined by the vertices incident to f and its tiling vertices and tiling edges undergoes planarization, and all of its faces are identified as in Section 3.5.2. Finally, the incidence of a sight cell s is defined as the set of neighbors of the split vertex which can be reached from the centroid of s without crossing any non-tiling edges. Convex faces may be considered their own sight cells.

At this stage, the outer face e.g., (A, Y, D, X, C, B, A) in Figure 3.8, can be considered a special case of a non-convex face, as unlike inner faces, the outer face is not bounded. Thus, a rectangular bound, the size of the drawing bounds, is added within which we identify the outer face's sight cells (Figure 3.4 (b)). The decomposition process is, however, largely the same, with the exception that intersections with the bounding rectangles must also be considered. Thus, we obtain the drawing D_t. For an example of non-convex face decomposition, consider the outer face's decomposition in Figure 3.4 (b) indicated in blue (•).

Selecting Sight Cells With all the incidences of the sight cells computed, we must select a pair of sight cells in which we will embed each copy. In line with desideratum **D2**, we aim to select two faces that jointly maximize the number of unique incidence vertices. Moreover, in line with desideratum **D1**, we only select two faces as we only split the selected target vertex into two. Ties are resolved by considering the absolute Euclidean difference between each edge and its Kamada-Kawai optimal graph theoretic length; *the smaller the difference, the better.*

We model each problem using an Integer Linear Programming (ILP) formulation, where we have only integers as variables in the linear system. We are given a set S of cells and the set N of neighbors of the split vertex, and look for the pair of sight cells that maximizes the incidence of crossing free edges to the copies of the split vertex. We denote by s_i the i-th sight cell, v_j the j-th neighbor of the split vertex, and $e_{i,j}$ the edge between a copy embedded in cell s_i connected to vertex v_j. We set $s_i = 1$ if a copy is embedded into cell s_i, and set the value to 0 otherwise. Similarly, if vertex v_j has its incident edge to the split vertex assigned to a copy embedded in cell s_i, then $e_{i,j} = 1$. We maximize the realized edges as:

$$\sum_{i \text{ s.t. } s_i \in S} \sum_{j \text{ s.t. } v_j \in N} e_{i,j} \tag{3.1}$$

under the following constraints: At most two cells can be selected: if $\sum s_i \leq 2$, a neighbor can only be assigned to one cell (meaning the copy to be embedded in that cell): if $\sum_i e_{i,j} \leq 1$ for each neighbor vertex. Additionally, a neighbor cannot be assigned to a cell that has not been selected: $s_i \geq e_{i,j}$. Lastly, if there is no direct visibility between the neighbor and the sight cell, no crossing free edge can be drawn to a copy embedded in that sight cell. We consider $a_{i,j}$ the binary variable that denotes with $a_{i,j} = 1$ the direct visibility of neighbor j to cell s_i. Thus, our last constraint is $e_{i,j} \leq a_{i,j}$.

Figure 3.5 (b) shows the selected sight cells as blue-shaded areas. These particular sight cells happen to correspond to the previously selected faces, as discussed in Section 3.5.2.

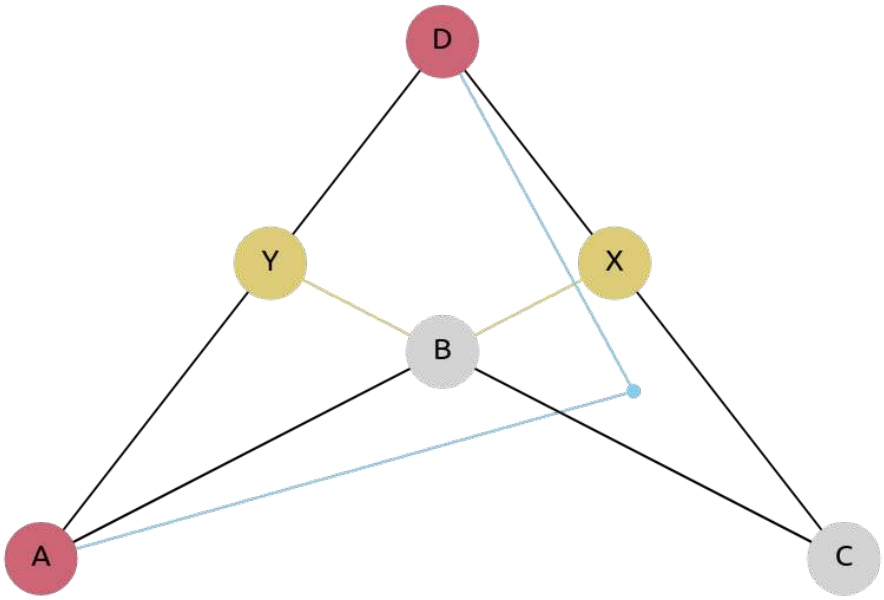


Figure 3.8: Illustration of the decomposition of some non-convex inner face (A, B, C, D, A). Incident vertices A and D are adjacent to the target vertex to be split. Here, the angle formed by the edges incident to vertex B, (E_{AB}) and (E_{BC}) forms an angle greater than 180°. Ray (AB) is projected until it intersects the closest edge, E_{CD}, where a new tiling vertex X is placed. The intersected edge, E_{CD}, is replaced by paths through that new vertex, i.e., E_{CX} and E_{XD}. A tiling edge is created between the newly placed tiling vertex X and vertex B. This process is repeated for ray (CB), creating *tiling vertex* Y. The identified sight cells are, (A, B, Y, A), (Y, B, X, D, Y), and (B, C, X, B). To illustrate the calculation of sight cell incidence, consider sight cells (B, C, X, B). Lines, in blue, are drawn between the sight cells' centroid (•) and the two vertices adjacent to the split target. Vertex D can be reached without crossing any edges. Vertex A, however, cannot, as the project line crosses the edge E_{BC}. Thus, its incidence is only $\{A\}$. The incidence of the two remaining sight cells is $\{A, B\}$.

3.5.3 Step 3: Subface Decomposition and Selection

With two sight cells selected, we must now determine where within each face to embed the split copy of the target vertex. More specifically, in line with desideratum **D3**, we must find the locations which jointly minimize the number of remaining edge crossings, when connecting to the split vertex's two copies to neighbors not incident to either selected face.

Line Segmentation Given D_T, the drawing in which each face is a sight cell, we create a drawing D_R by drawing lines between vertex pairs (Figure 3.5 (a)). Note that, unlike the version presented by Sanatnama et al. [SAHB11], we are not considering an arrangement of lines, but rather of segments as we end the line segments on their intersection with the outer bounding rectangle. This allows us to skip area candidates located outside of the drawing canvas. Similar to the previous planarization steps, we replace crossing lines with non-crossing paths of segments, and we identify a set of cells in the resulting drawing. These cells are called *subfaces* $B = \{b_1, b2, ..., b_g\}$ (Figure 3.5 (b)). Note that there is a unique mapping of each subface to the face in F it belongs to.

Sanatnama et al. [SAHB11] proved that for such a decomposition of the space, all points within a particular subface are equivalent to one another in terms of the number of edge crossings induced when connecting to other vertices in the graph drawing. However, in line with desideratum **D2**, we do not consider all subfaces, but only those within the two selected sight cells (see Figure 3.5 (b)). Lastly, we compute the centroids of all the subfaces within the two selected faces. For each centroid c and each neighbor u, we store the number of crossings induced by the edge $e_{c,u}$ in an array. Using this array, we solve the problem by selecting one subface in each of the sight cells previously selected. Figure 3.5 (a) showcases the line decomposition of the previous example graph. The project lines are drawn in yellow ( ), terminating just beyond the user-defined canvas bounds shown in blue ( ). Once planarized, like the graph of Figure 3.4 (a), the line-segmented graph's faces can be identified, highlighted as yellow-shaded areas in Figure 3.5 (b).

Subface Selection We use a similar process as for the sight cell selection step, but rather than having as input the visibility between the neighbors of the split vertex and the candidate sight cells, we have as input the number of crossings $x_{i,j}$ that would be induced by an edge between a vertex v_j and a copy of the split vertex embedded in subface b_i. We minimize the number of crossings using the following objective function:

$$\sum_{i \text{s.t.} b_i \in B} \sum_{j \text{s.t.} v_j \in N} x_{i,j} \tag{3.2}$$

For each sight cell we only want to select one sight cell, $\sum b_A(i) \leq 1$ where b_A corresponds to the subfaces of one of the previously selected sight cells, and b_B the others. We use the same constraint for the b_B. Similarly to previously, we define a constraint that limits a neighbor to be assigned to only one subcell, and additionally to a sight cell that has been chosen. This ILP returns the two subfaces that induce the least amount of crossings, as well as the assignments of the neighborhoods to the subfaces. Figure 3.5 (b) highlights the selected subfaces as yellow-shaded areas. Here, the subface incident to vertex 4 allows for the connection to target neighbor 5 by only inducing two additional edge-crossings. Admittedly, in this case, that would have held for all subfaces within the selection sight cell in question.

3.5.4 Step 4: Embedding of the Split Vertex Copies

With two subfaces selected based on the induced number of edge crossings to non-sight cell-incident vertices, the process of embedding the two split copies is straightforward. While more complex embedding rules based on other graph aesthetics could have been chosen, we simply put each copy on the coordinates of the centroids of the selected subfaces. Alternatives include basing the exact embedding location on the best edge-angle ratio or edge-angle ratio. Edges are subsequently drawn according to the assignment we just obtained (see Figure 3.6 (a)). Comparing this new embedding to the original graph drawing of Figure 3.6 (b), it has indeed successfully resolved 16 of the 18 edge crossings that vertex 2 's incident edges were involved in.

3.5.5 Implementation

The algorithm was implemented in *Python* around the *NetworkX* [HSS08] package. Specifically, we used its implementation of the Kamada-Kawai [KK89] and Kavitha et al.'s [KMMP08] algorithms. The implementation will be made publicly available on GitHub.

3.6 User Study Design

Lastly, we are interested in evaluating how this approach to vertex duplication affects a user's ability to perform a set of commonly encountered graph analysis tasks. These tasks and the data analyzed are representative of actual types of analyses a user could expect to face when visually analyzing or reading graph drawings. For the purpose of aesthetic simplicity, split vertex embeddings were limited to inner faces only.

3.6.1 Tasks

Within the context of the work of Lam et al. [LBI+11] on proposing a high-level taxonomy of information visualization tasks, we are interested in obtaining *subjective user feedback* on perceived effectiveness and preferences, i.e., *user experience*, and an *objective measure of how well and quickly* users performed select tasks, i.e., *user performance*. For the latter, we can further couch our objective evaluation goals within the mid-level typology of task types by Brehmer et al. [BM13]. Specifically, we are primarily interested in evaluating how effectively a user can *locate* or *extract* information in a split graph, as compared to a non-split graph. Depending on whether the target entity is clear or its location known, the tasks performed fall within the *Search* category and its four subcategories of *Lookup*, *Browse*, *Locate*, or *Explore* [BM13]. Lastly, we define the three actual (or low-level) tasks for users to perform based on Lee et al. [LPP+06b] taxonomy of graph analysis tasks. Here, we focus primarily on *topological* and *browsing* tasks, as we suspect these to be most affected by splitting operations [HBF08], but also lend themselves more readily to objective measurement. Within the category of *topological tasks*, we define two task

types of interest, **T1** and **T2**, and within the *browsing tasks* category, we define a final task **T3**. Thus, our study comprises the following three tasks:

T1 **Vertex adjacency**, i.e., identifying which vertices are adjacent to some given vertex;

T2 **Common connections**, i.e., identifying the set of vertices adjacent to two given vertices; and

T3 **Path tracing**, i.e., identifying which of a set of paths is indeed possible in the given graph drawing.

Each of these three tasks was presented to each participant twice; once for the normal Kamada-Kawai [KK89] embedded drawing, and once for that graph's split drawing, with randomized vertex labels.

3.6.2 Data

Normal Drawings To produce graphs with small-world properties, we simulate three networks using the Watts-Strogatz graph model and embed them using Python's *NetworkX*'s [HSS08] implementation of the Kamada-Kawai algorithm [KK89]. To ensure that the drawings had sufficient crossings for splitting while remaining readable for non-graph experts, we specifically simulated these graphs with a number of vertices $n = 10$, a mean degree $m = 6$, and a rewiring probability $p = 0.5$. The non-split graph cases, as selected for the study, are depicted in the upper row of Figure 3.9, i.e., graphs A_{normal}, B_{normal}, and C_{normal}.

The small number of vertices $n = 10$ was selected for three key reasons. First, the (statistical) importance of edge crossings for graphs of such small sizes has been thoroughly documented [PCA02, KPS14], thus presenting a compelling argument for vertex splitting to be applied to graphs of this size. Second, as the participant list includes non-graph-experts and to ensure enough participants could be gathered, a smaller number of vertices was selected to ensure the study would not take longer than 20-30 minutes in total to complete. And lastly, our approach was able to consistently produce aesthetically sensible and pleasing results for graphs of such sizes. The combination of mean degree parameter $m = 6$ and rewiring probability $p = 0.5$ was selected to ensure that at least 2 split vertices were needed to resolve 50% of the edge crossings in the embedding.

Split Drawings Comparable to the study performed by Kobourov et al. [KPS14], each of these three graphs was iteratively split until 50% of the original embedding's edge crossings were resolved. As identifying the aesthetically optimal number of splits is beyond the scope of this chapter, we opted to resolve 50% of edge crossings as, in informal previous testing, we found it provided enough challenge for users without being completely overwhelming. Specifically, to ensure participants would actually complete the study, we aimed for a completion time no longer than 30 minutes. In order to ensure split vertices were recognizable in non-interactive visualization, split vertices were color-coded

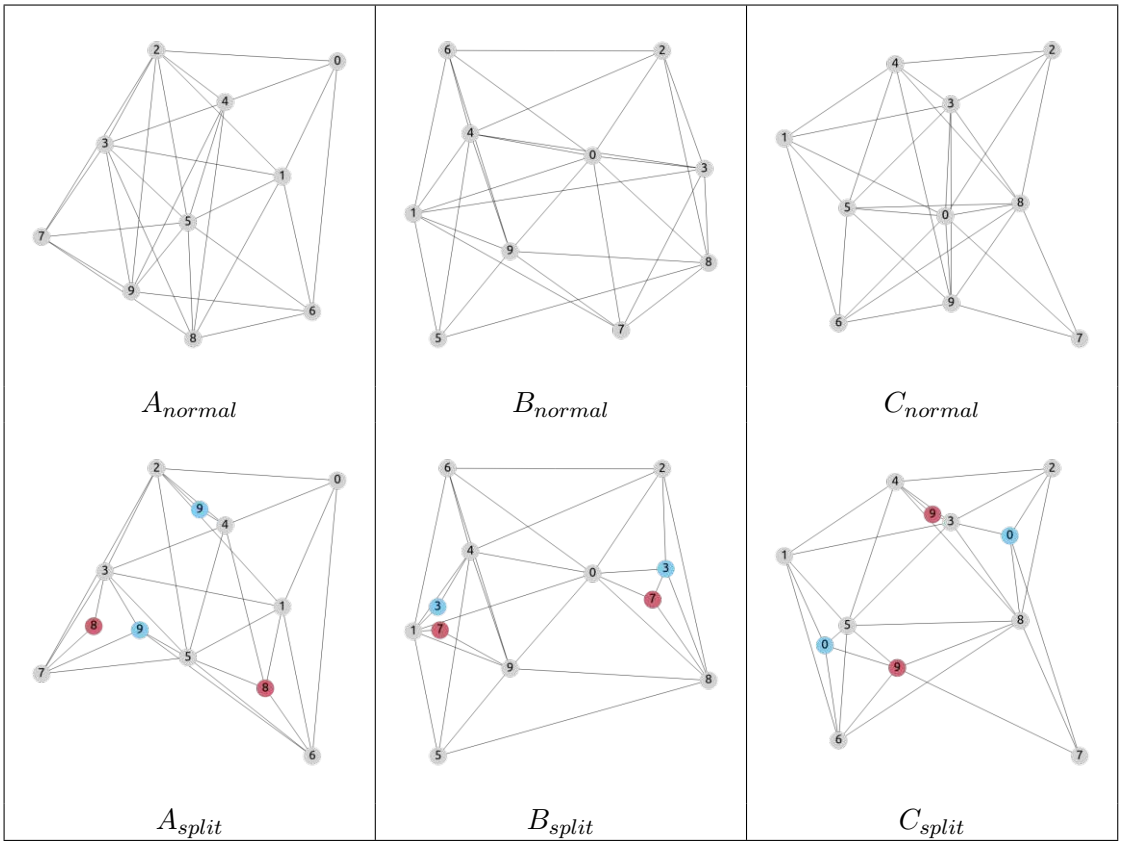


Figure 3.9: Three Watts-Strogratz-simulated graphs, (A), (B), and (C), presented to participants during the conducted user study. All graphs were simulated using the same parameters of $n = 10$, $m = 6$, and $p = 0.5$. Each graph was laid out using Python's *NetworkX* library's Kamada-Kawai spring embedding (*normal*) and then split until 50% of the original drawing's edge crossings were resolved (*split*). Please note that, for illustration purposes, the graphs presented here neither have their labels randomized nor their positions inverted. Split vertices share the same label and are color coded (e.g., 8 or 9) to communicate that they have been split. These graphs' vertices and labels have been enlarged threefold for the sake of readability in print.

using a color-blind-friendly palette [HB03]. Lastly, to avoid any learning effect, each graph has its labels randomized, and the original drawing had each vertex's positions inverted. The split graph cases, as selected for the study, are depicted in the lower row of Figure 3.9, i.e A_{split}, B_{split}, and C_{split}. These are the split counterparts of the upper row of the figure.

3.6.3 Hypotheses

User Performance Before beginning the study, participants were instructed on what graph drawings are, what split vertices are, and how the three tasks were to be performed. Users were made aware of the fact that the accuracy and time taken were being evaluated. Over the course of the anonymous evaluation, each participant was presented with each of the three simulated graphs. Such a within-subject experimental design was chosen to ensure we had each user's performance across all tasks and preference for both unsplit and split graphs. For each graph, once for the split and once for the non-split drawing, participants were tasked with answering one of the three aforementioned tasks (**T1**–**3**), as accurately and quickly as possible. Each task had a multiple-choice list from which to select the set of correct answers, which were known to us. Each participant was randomly assigned to one of the six unique assignments of tasks to graphs. The randomization ensured that no participant saw the same graph twice, be it split or unsplit. Additionally, the order in which graphs, tasks, and multiple-choice answers were presented was also randomized to ensure no potential learning effect. Thus, each participant was presented with six questions in total, i.e., vertex-adjacency for a normal and split graph, common-neighborhood for a normal and split graph, and path validity for a normal and split graph. For each question, we recorded the time taken by the participant, as well as the accuracy of the answer, measured as the percentage of choices correctly selected. For this performance evaluation, we posit the following hypotheses:

H1 The accuracy of determining a vertex's adjacency is positively affected by splitting operations, because each individual vertex drawing's degree is lower and, thus, its neighbors are easier to identify.

H2 The accuracy of determining the common neighbors of two given (split) vertices will be positively affected, as the connectivity of individual vertices is clearer.

H3 The accuracy of determining whether a path exists is negatively affected by splitting operations, as a path passing through a split vertex necessitates scanning all copies of the vertex to determine whether the next connection exists.

H4 The time necessary to answer all of these questions will be negatively affected by splitting operations, as more vertices are present in a drawing.

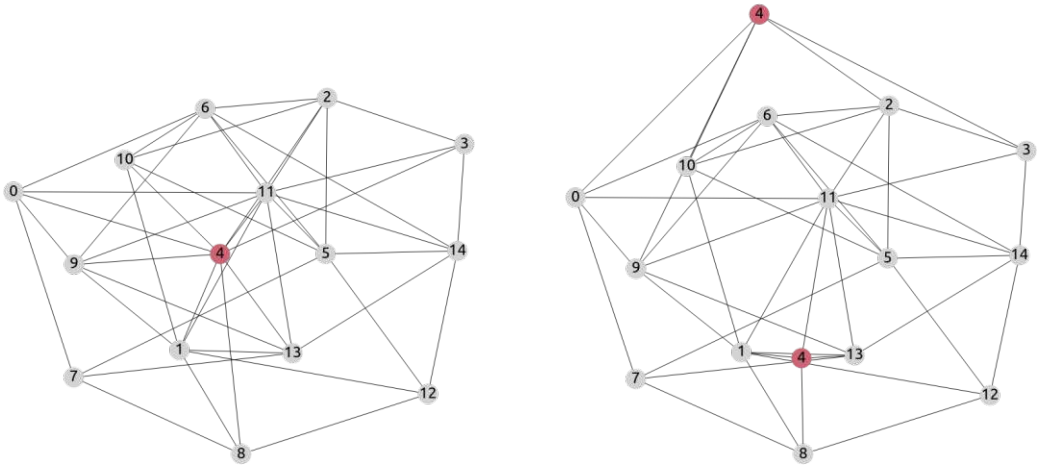
User Preference Additionally, we are interested in evaluating user preferences. As outlined by Lam et al. [LBI+11], we aim to evaluate each participant's perceived effectiveness and preference, i.e., *user experience*. Specifically, at the end of the study, for each task, each participant was asked i) whether they believed split embeddings allowed them to answer questions more accurately, and ii) whether split embeddings allowed them to answer questions more quickly. Additionally, to evaluate *usefulness* and preference, we probe whether split graphs are easy to learn and easy to use. We also probe whether they are aesthetically preferable over non-split graphs [GKO18]. All answers were given in the form of a 5-level Likert scale: (1) strongly disagree, (2) disagree, (3) neutral, (4) agree, and (5) strongly agree. For this preference evaluation, we posit the following hypotheses:

H5 Users will aesthetically prefer split over non-split drawings of graphs, as these feature fewer edge crossings.

H6 Users will find split graphs difficult to learn, as certain tasks, such as path identification and common neighborhoods, involve some initially counter-intuitive thinking.

H7 Users will find split graphs more difficult to use, for reasons similar to those in **H6**.

3.7 User Study Results



(Original Embedding) (Split Embedding)

Figure 3.10: Vertex Splitting applied to a Watts-Strogratz simulated graphs with numbers of vertices $n = 15$, mean degree of $m = 6$, and rewiring probability of $p = 0.5$: (Original Embedding) shows the initial Kamada-Kawai spring embedded drawing, and (Split Embedding) the drawing produced by splitting vertex 4.

52 participants took part in this anonymized, crowd-sourced, online user study. Users were encouraged to use large monitors when performing the study. In this setting, cheating was technically not preventable or controlled. The study was not pre-registered. Participants were invited by snowballing, i.e., reaching out to university and academic contacts who were also encouraged to forward the study invitation to their contacts. As all submissions were anonymous, we do not know exactly who participated in this study. Most participants fell within the 21–25, 26–30 and 31–35 year old age brackets with 9, 21, and 15 people respectively. Moreover, of the 52 participants, 18 self-reported being *experienced* with graphs, 18 as *very familiar*, and 6 as *expert*. Only 1 self-reported as being *completely unfamiliar*, and 9 as *not very familiar*. Most participants had completed

either a Master's degree (28) or a Ph.D. (13), though 8 reported having completed a Bachelor's degree. Lastly, 16 participants identified as female, 25 as male, and 1 as non-binary. All participants had normal (potentially corrected) vision. As assumptions of normality could not be made or validated for the data at hand, answer accuracy and time taken per task were analyzed using a two-sided, paired Wilcoxon Signed Rank test (Figure 3.11 and Figure 3.12). Preferences were analyzed using a two-sided, one-way Wilcoxon Signed Rank test (Figure 3.13) [Wil92]. When probed using Wobbrock et al.'s [WFGH11] *Aligned Rank Transformed ANOVA*, no statistically significant or substantively notable association between the expertise of users and their performance or preference was found.

3.7.1 User Performance

Task accuracy *P*-values for a study such as this should be viewed with skepticism. However, it is interesting to note that the differences in answer accuracy were statistically significant for both the *Vertex Adjacency* and *Path Tracing* tasks, in favor of split graphs (Figure 3.11). However, likely a product of the small graphs used in this study, the actual differences in answer accuracy observed are fairly small, i.e., $\Delta_{VertexAdjacency} = 0.091$, $\Delta_{CommonNeighbors} = 0.043$, and $\Delta_{PathTracing} = 0.101$ These findings are in agreement with hypotheses **H1**, i.e., that splitting would positively affect the user's ability to determine the adjacency of given vertices, and **H2**, i.e., that splitting would positively affect the user's ability to determine the common neighbors of two given vertices. Interestingly, while we hypothesized that vertex splitting would negatively impact the accuracy of path tracing in **H3**, the opposite appears to be found. The results of this analysis are shown in Figure 3.11.

Task completion time In agreement with **H4**, the time needed to complete tasks was negatively impacted by vertex splitting, though, again, the mean differences were very small, $\Delta_{VertexAdjacency} = 1$ second, $\Delta_{CommonNeighbors} = 5$ seconds, and $\Delta_{PathTracing} = 10$ seconds. Differences were statistically significant for both the *Common Neighbors* and *Path Tracing* tasks, where more time was required to complete the tasks with the split graphs. The results of this analysis are shown in Figure 3.12.

3.7.2 User Preference

Interestingly, participants showed a fairly strong preference for split over non-split graphs (Figure 3.13), with an average Likert scale agreement of 3.89 with the statement "*I found split graphs more aesthetically pleasing*", agreeing with our hypothesis **H5**. Participants also reported split graphs as being both easy to learn and understand, disagreeing with our posited hypotheses **H6** and **H7**. Lastly, on average, users perceived correctly that split graphs allow them to answer all questions more accurately, though no definitive conclusions regarding perceived completion time could be made. The results of this analysis are shown in Figure 3.13.

3.7.3 Results Summary

In summary, we find that vertex splitting does not meaningfully affect the users' ability to identify shared neighbors given two vertices. Vertex splitting even *improves the accuracy* in identifying the adjacency of a given vertex or checking the validity of a set of given paths. However, it also *negatively affected the time* taken to perform the three tasks — in particular, the identification of valid paths. Furthermore, participants reported *notable aesthetic preference* for split over non-split drawings of the graphs shown. Participants also found split graphs *easy to learn and understand*. All in all, these results showcase that vertex splitting, and our approach to vertex splitting, could be useful in improving the readability and aesthetic quality of small graph drawings.

3.8 Discussion and Future Outlook

While we hope to have demonstrated that vertex splitting, and our approach in particular, can be useful in assisting users for small graph drawings, a number of issues, both conceptual and computational, remain to be addressed in future work.

3.8.1 Algorithm and Implementation

Computational Scalability To ensure such an approach could find application to larger and denser graph drawings, a more computationally scalable approach to the face identification problem is needed, as our current approach is centered around Kavitha et al.'s [KMMP08] $O(m^2n)$ minimum cycle basis detection algorithm as implemented in *Networkx* for m edges and n vertices. This implementation forms the key bottleneck at three steps, namely the detection of sight cells, the detection of subfaces, and, most severely, the detection and (recursive) checking of faces (Figure 3.14). Here, moving the implementation out of *Networkx* would allow for the planar graphs to be stored as a rotation system, i.e., circular, doubly-connected edge lists, with pointers to faces. Additionally, moving the entire implementation to a more efficient, low-level programming language, or potentially parallelizing the face identification on the GPU, could be useful in making this approach applicable to larger systems.

Beyond Edge Crossings While this approach produces more aesthetically pleasing results for smaller graphs, it fails to do so consistently for increasing numbers of vertices. For example, consider in Figure 3.10 the edges placed between the top-most split copy of vertex 4 and vertices 9 and 10, which are difficult to distinguish from one another. Here, a more sophisticated cost function incorporating not only the number of edge crossings but also other quantitative graph aesthetic criteria would allow for a better selection of target vertices, sight cells, and subfaces. Additionally, such criteria could assist in choosing better embedding locations within selected subfaces. Here, some aesthetic criteria of interest could include, for example, edge crossing angles, edge length ratios, edge angle ratios, vertex density, or edge/vertex occlusion [Pur02]. For the example given above, the edges $e_{(9,4)}$ and $e_{(10,4)}$ would produce a poor edge angle ratio,

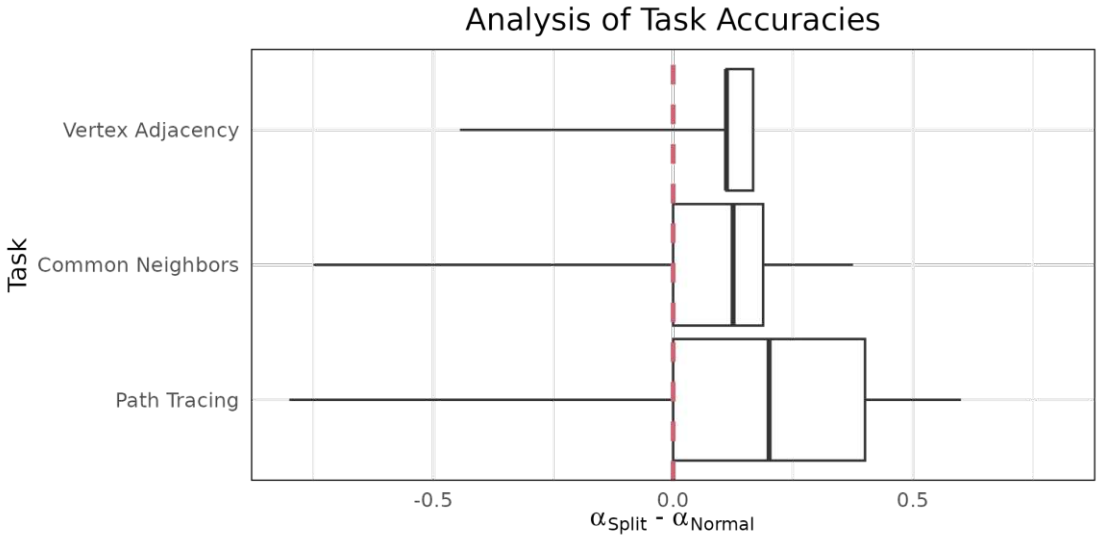


Figure 3.11: Box plot of performed paired Wilcoxon signed rank test [Bau72] for answer accuracy across the three task types (**T1**–**3**). *P*-values are calculated using a null hypothesized difference of $\mu_0 = 0$. At an *a-priori* specified significance level of $\alpha = 0.05$, accuracies are statistically significantly different. More specifically, for **T1** and **T3**, users were more accurate when using split graphs, with a median difference of 11.1% and 20.0%, and *p*-values of $p_{T1} = 0.000003$ and $p_{T2} = 0.0144$, respectively. Task **T2** was not statistically significant between groups with a median difference of 12.5% and $p_{T1} = 0.1104$.

guiding the algorithm away from that particular embedding location. Additionally, this could allow for a more sophisticated approach to tie resolution. Currently, selected target vertex (Figure 3.3), sight cell (Figure 3.4), and subface (Figure 3.5) ties are resolved quite naively, such as vertex degree to select an ultimate target vertex to split from a set of vertices with identical crossings numbers. In general, the application of vertex splitting to improve aesthetic criteria beyond edge crossings would be interesting. Of course, the inclusion of such additional criteria would also bring with it additional computational cost as well.

Beyond Local Optimality Given a more computationally tractable implementation, one could consider relaxing desideratum **D2**, the minimization of edge-crossing-free connections, and focus instead only on desideratum **D3**, the minimization of the total edge-crossing number. The unfortunate side effect of first satisfying desideratum D2 and then D3 can be seen in Figure 3.6 (a), where the lower of the two split vertices is suboptimally placed. Had it been placed to the left of the edge connecting vertices 4 and 6, instead of the right, one additional edge crossing could have been resolved. However, as the embedding face was selected before the subface within it, these two faces were tied in the number of incident adjacent vertices during the face-selection step.

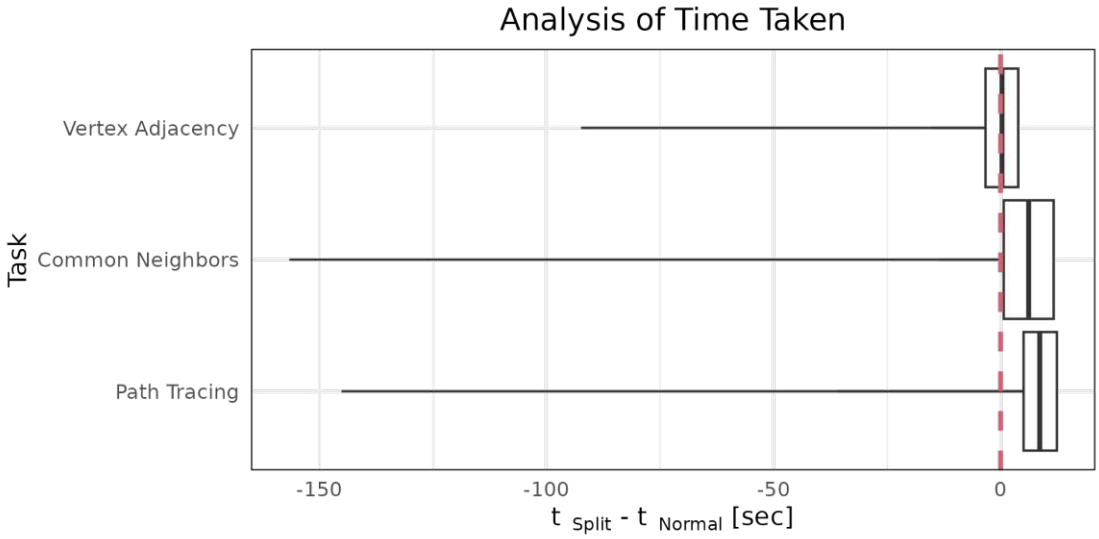


Figure 3.12: Box plot of performed paired Wilcoxon signed rank test [Bau72] for the time taken to answer the three task types (**T1**–**3**). *P*-values are calculated using a null hypothesized difference of $\mu_0 = 0$. At an *a-priori* specified significance level of $\alpha = 0.05$, the differences in time taken are statistically significantly different for **T2** and **T3**. More specifically, users were slower using split graphs, with a median difference of 6.17 seconds ($\sim$ 13.9% slower) and 8.61 seconds ($\sim$ 14% slower), and *p*-values of $p_{T2} = 0.036$ and $p_{T3} = 4.967E^{-05}$, respectively. Task **T1** was not statistically significant between groups with a median difference of 0.306 seconds and $p_{T1} = 0.866$.

Here, instead of selecting a face and then a subface within it, this would allow for the straightforward selection of 2 (or k) subfaces from the line-segment-tiled embedding (Figure 3.5 (a)). While the tiling of the entire embedding would be more computationally expensive, it would also allow for the identification of a vertex split configuration that minimizes the number of crossings for this graph embedding.

Beyond Single Splits In line with desideratum **D3**, i.e., minimizing the number of split vertices, we only split a selected vertex in two. However, beyond identifying *where* to embed split copies, it would be interesting to also identify *how many* splits are necessary to minimize the selected graph aesthetic cost function. On the one hand, extending the ILP selection of sight cells and subfaces to allow for a selection of an arbitrary number of splits would be conceptually straightforward. On the other hand, it would also necessitate tiling more sight cells into subfaces, thereby introducing additional computational overhead.

Combining *a priori*, Interactive, and Algorithmic Splitting: It would be interesting to combine this technique with existing approaches to vertex splitting, such as *a priori* specification of vertices to split completely. Such *a priori* split vertices would probably resolve many aesthetic issues from the beginning, making the subsequent algorithmic splitting operations easier and faster. Alternatively, an interactive, human-in-the-loop

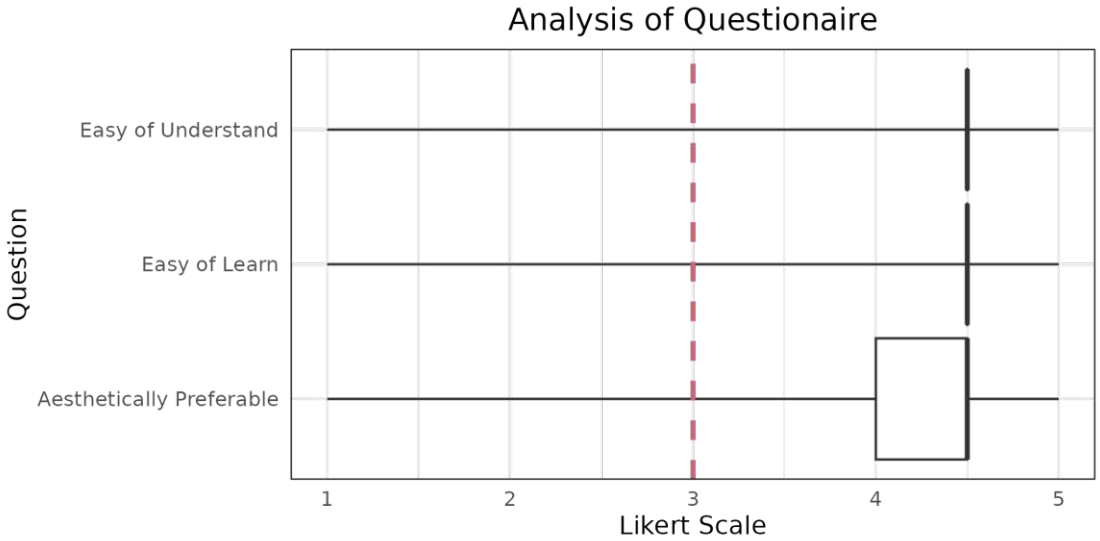


Figure 3.13: Box plot of performed paired Wilcoxon signed rank tests [Bau72] on Likert-scale user responses. Answers range from 1 (Strongly Disagree) to 5 (Strongly Agree). P-values calculated with a null hypothesized value of $\mu_0 = 3$. At an *a priori* specified significance level of $\alpha = 0.05$, **T1**, **T2**, and **T3** were all statistically significant with median differences of ~ 4.5, and p-values of $4.436E-08$, $2.656E-07$, and $6.665E-05$, respectively.

approach to graph drawing could allow users to select vertices to split and/or embedding locations, which could side-step some of the computational and conceptual complexity discussed. This would, of course, also require additional features, such as interactive highlighting of split copies [HBF08].

Compromising the Mental Map Motivated by the preservation of a user's mental map, graph drawings were not re-embedded after each vertex split. For graphs of this size ($10 \leq n \leq 15$ and density, this approach still functions. However, even for graphs of size $|V| = 20$, we encounter several limitations. First, as the number of vertices increases, so does the complexity of the line segmentation (Figure 3.5 (a)), resulting in smaller and smaller subfaces. Embedding a split vertex copy within such a subface can cause considerable occlusion of both incident and nearby edges, as well as nearby vertices. Moreover, for graphs of high edge density, a single split will often not resolve many edge crossings, and necessitate either splitting said vertex not twice, but k times, or splitting additional vertices. Here, allowing for a split graph drawing to be re-embedded using, for example, Kamada-Kawai, would allow for better graph aesthetic criteria in the final drawing, as well as additional edge crossings to be resolved "passively".

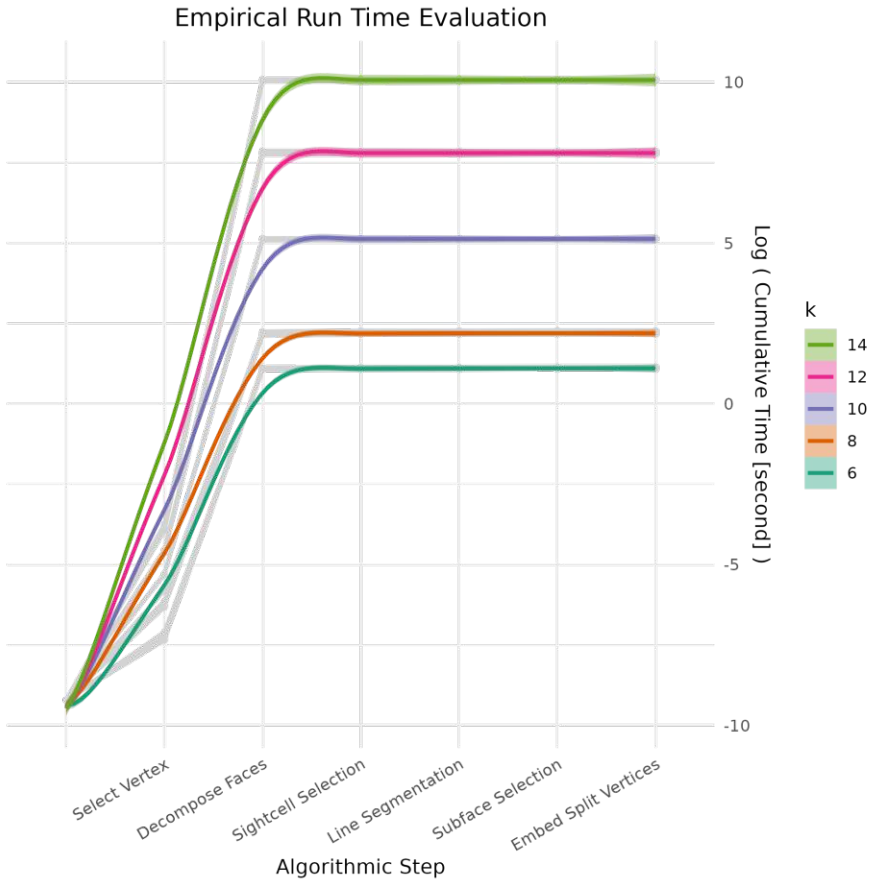


Figure 3.14: Cumulative empirical (logarithmic) runtimes of the vertex splitting method evaluated on k-complete graphs of increasing size, from $k = 6, 8, 10, 12$ and 14. Time was collected for the 6 steps of the algorithm, namely *Select Vertex* (Figure 3.3 (a) and (b)), *Decompose Faces* (Figure 3.4 (a)), *Subface Selection* (Figure 3.4 (b)), *Line Segmentation* (Figure 3.5 (a)), *Sight Cell Selection* (Figure 3.5 (b)), and *Embed Split Vertex* (Figure 3.6 (a)). For each scenario, 100 graphs were simulated and the time taken to perform these steps was recorded, shown as light grey lines and points. For each k graph, the smoothed average run time and 95% confidence interval of its 100 runs is calculated and overlaid in color. As expected, the face decomposition step, reliant on Kavitha et al.'s [KMMP08] $O(m^2n)$ algorithm, where m and n denote the number of edges and vertices, respectively, forms the computational bottleneck of our implementation.

3.8.2 User Performance and Graph Aesthetics

Extending the User Study While the results of this user study are promising and corroborate the findings of Henry et al. [HBF08], they are, of course, limited by the size and complexity of the graphs chosen. Here, a larger user study featuring larger and denser graphs [KPS14], more varied graph analysis tasks [LPP+06b], and graphs

simulated using various graphs models [WS98, BB03, Alb05, BA99, Bol01], would provide the opportunity to more completely investigate the effect of vertex splitting on both performance and preference. Of course, this would also necessitate a much larger number of participants and then require the use of a crowd-sourced or Amazon Mechanical Turk approach to data acquisition. However, such a study would be instrumental in uncovering both *how* and *when* to split, in order to most effectively assist users.

A Quantitative Evaluation Beyond studying user performance and preference, it would be valuable to study the effectiveness of such an approach to vertex splitting quantitatively. More specifically, it would be interesting to calculate graph aesthetic metrics, beyond simply the number of edge crossings, for multiple graphs simulated using different random graph models, vertex sizes, and fixed split numbers, to observe how these metrics change across iterations, i.e., splits. Combining such empirical results with an additional user study could help us understand when vertex splitting is useful and most appropriate. This could be as simple as understanding what graph sizes and edge densities users prefer, split over non-split graphs, or to what degree they do so. This could subsequently lay the groundwork for developing a graph-aesthetic-based stopping criterion for vertex splitting. Beyond the number of edge crossings resolved, such a stopping criterion could be based on, for example, edge angle ratios, edge length ratios, or (remaining) edge crossing angles.

Application to Real-World Networks Our results showcase the potential for vertex splitting as a means of improving readability. However, owing to the computational complexity of our approach, we are currently limited to smaller, but also *in-silico* results. Subsequently, beyond extending our method along the lines discussed here, studying their utility to actual biological pathways, social networks, or transportation networks would be interesting. Adapting our approach for such real-world networks would present additional domain-specific challenges, such as canonical and potentially non-straight line representations, such as transportation maps or hand-drawn embeddings, and analytical tasks that may be negatively affected by vertex splitting.

3.9 Conclusion

We presented a novel, systematic approach to vertex splitting for static, straight-line graph drawings, agnostic of how their embedding was produced. We showed its applicability to small-scale graphs. A conducted user study demonstrated that (this particular approach to) vertex splitting allowed the users to answer three graph analysis tasks more accurately, at the cost of requiring additional time to do so. Additionally, these split graphs were aesthetically preferred over non-split graphs, while still being considered easy to understand and learn. These results, as well as the pragmatic use of vertex splitting in the field, indicate to us that vertex splitting could be a useful approach to making dense graphs more readable. We note a sizeable gap between graph theory and visualization applications when it comes to the study of vertex splitting; a gap hopefully closed with future research. Beyond making this approach more computationally scalable, its immediate application to small-scale domain networks, such as metabolic pathways, would be an interesting next step.

CHAPTER 4

Think Globally, Act Locally: A Comparison of Ego Network Visualization Approaches

A key challenge underlying biological network visualization endeavors is the massive size of modern graph datasets, with an exploding number of nodes and edges depending on the particular biological subdomain. Straightforward visualization of such networks is often no longer possible or desirable, producing so-called "*hairballs*". While improving the graph aesthetic criteria of such hairballs, e.g., through vertex splitting, is one possible approach to improving such graphs' readability, another is to side-step visualizing the entirety of the total network altogether, focusing, instead, on local substructures only. One possible approach is to only visualize the immediate neighborhood of some selected node of interest, i.e., visualizing its so-called *ego network*. However, while ego networks are a popular approach in (dynamic) social network visualization, little systematic work has been published on the topic, at least at the time of writing. Moreover, outside of the social sciences, they remain a highly underexplored technique. Thus, we set out to:

1. *characterize* the current state-of-the-art of ego network visualization, i.e., understand what kind of visualization strategies are used across various domains
2. *study* the effect of different ego network representations on user performance and preference, and
3. *develop* a first set of guidelines on how and when to utilize certain ego network representations over others.

The contents of this chapter are based on the full paper *"Me! Me! Me! Me! A Study and Comparison of Ego Network Representations"* [EPF+24], published in the journal *Computer & Graphics* with myself as first author, in collaboration with Daniel Pahr, Velitchko Filipov, Hsiang-Yun Wu, and Renata Raidou.

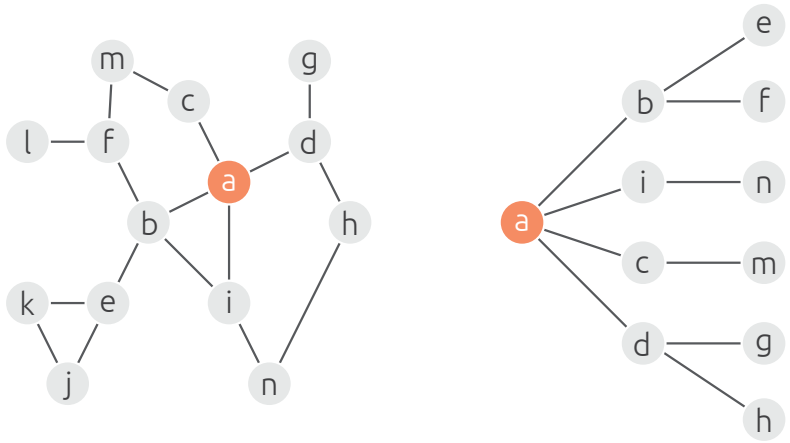
4.1 Overview

From social networks to brain connectivity, ego networks are a simple yet powerful approach to visualizing parts of a larger graph, i.e., those related to a selected focal node — the so-called "*ego*". While surveys and comparisons of general graph visualization approaches exist in the literature, we note i) the many conflicting results of comparisons of adjacency matrices and node-link diagrams, thus motivating further study, as well as ii) the absence of such systematic comparisons for ego networks specifically. In this chapter, we propose the development of empirical recommendations for ego network visualization strategies. First, we survey the literature across application domains and collect examples of network visualizations to identify the most common visual encodings, namely straight-line, radial, and layered node-link diagrams, as well as adjacency matrices. These representations are then applied to a representative, intermediate-sized network and subsequently compared in a large-scale, crowd-sourced user study in a mixed-methods analysis setup to investigate their impact on both user experience and performance. Within the limits of this study, and contrary to previous comparative investigations of adjacency matrices and node-link diagrams (outside of ego networks specifically), *participants performed systematically worse when using adjacency matrices than those using node-link diagrammatic representations.* Similar to previous comparisons of different node-link diagrams, *we do not detect any notable differences in participant performance between the three node-link diagrams.* Lastly, our quantitative and qualitative results indicate that *participants found adjacency matrices harder to learn, use, and understand than node-link diagrams.* We conclude that in terms of both participant experience and performance, a layered node-link diagrammatic representation appears to be the most preferable for ego network visualization purposes.

4.2 Introduction

Ego networks, sometimes also called egocentric or personal networks [WPZ+16], are node-relative subgraph depictions of a larger graph's topology (Figure 4.1). That is to say, instead of drawing a graph in its entirety, one only draws those nodes and edges relevant to some selected focal node. This focal node is commonly referred to as the "*ego*", and its neighbors as "*alters*". More specifically, the ego's immediate neighbors, i.e., nodes of a 1-hop distance from the ego, are called 1-alters. The ego's neighbors' neighbors, i.e., nodes a 2-hop distance from the ego, are called 2-alters, and so on. This subgraph of a selected ego node and its k-alters is called an ego network.

Ego network visualizations, especially widespread in the social sciences [EHAE16, LHG+20], are a conceptually simple yet powerful approach for interactively reducing the visual complexity of larger networks. Consider here a number of such examples: Pu et al. [PZS+20] go beyond traditional supervised and unsupervised methods and employ ego network visualizations to enhance anomaly detection for fraud identification. Liu et al. [LGD+17] used split ego network representations to enable social scientists to tackle the challenging task of comparing two ego networks simultaneously. However, ego network



(a) Global Network Topology (b) Ego Network Topology

Figure 4.1: Construction of a 2-alter ego network representation (b) from some given input graph (a). The selected ego, i.e., node **a**, is highlighted in orange in both the network's global topological representation as well as its own ego network. The alters of the ego are shown in a tree-like structure, i.e., all its 1-alters are depicted on the tree's first layer, and all its 2-alters on the second.

representations of graphs have found (admittedly sparse) application outside of social network analysis as well. For one, Al-Awami et al. [AABS+14]'s *NeuroLines* framework visualizes nanoscale neuronal connectivity as node-relative networks. Alternatively, Sayers et al. [Say04] proposed a node-centric visualization of Resource Description Framework (RDF) graph topology.

In many applications, relationships within an ego network carry nuanced meaning that depends heavily on their importance. For instance, in social networks, the importance of relationships is represented as a weight (i.e., strength or frequency of interactions), allowing for analysis of close versus casual associations [EHAE16]. Similarly, in neuroscientific studies, weighted connections capture the intensity or frequency of neural interactions for identifying dominant pathways [AABS+14]. Weighted ego networks support tasks to reveal specific association patterns, i.e., identifying the most influential or strongest connections.

Depending on the particular application domain and graph data at hand, different types of network embeddings are employed—from conventional straight-line node-link diagrams [EHAE16] to dynamic adjacency matrices [ZGC+16]. However, *no systematic work has yet been conducted to study ego networks, their representations, or their applications.* Looking at general network visualization, there are several, still very much conflicting, user performance and preference studies comparing different types of network representations. For instance, node-link diagrams have been compared to adjacency matrices [GFC04, NWHL20] or different types of node-link diagrams to each other [HHE06, MS15], in order to determine what types of tasks and data are best served by which representations.

Ego networks are graphs centered around some particular node of interest, the ego, and are typically smaller-scale networks compared to conventional non-ego networks. Moreover, ego networks, typically, are employed for very different reasons than non-ego networks. Thus, while certain findings of such general network visualization comparisons transfer partially to ego networks and their visualization, their particular analytical goals, analytical tasks, and challenges warrant isolated and focused investigation. Here, we anticipate conventional straight-line node-link diagrams to be outperformed by more complex and structured visualizations that can better represent the layered nature of ego networks, even if such visualizations prove more difficult to learn and understand. To the best of our knowledge, these issues have caused visualization researchers to lack guidelines and best practices on when and how to use ego network visualizations effectively; a gap we aim to start filling with this chapter.

More specifically, we aim to highlight ego networks, commonly pigeonholed as a social network analysis-specific approach, as a potentially effective network visualization technique across domains that deserves greater attention. We make a first step in this direction by identifying which representations are common in the field and understanding how effective they are for particular graph analysis tasks. To do so, we provide an overview of various (weighted and) undirected ego-network applications, tools, and examples across different domains through an extensive survey of the literature of 50 papers. With the survey's results in hand, we identify four common approaches to visualizing ego networks, namely *(i) straight-line node-link diagrams* (✕), *(ii) layered (tree-like) node-link diagrams* (▲), *(iii) radial node-link diagrams* (●), and *(iv) adjacency matrices* (■). To identify when a specific representation is preferred over another, we conduct a large-scale crowd-sourced user study of 120 participants to empirically investigate the effect the layouts have on user performance and experience across six ego network-specific low-level graph analysis tasks. Additionally, we probe perception and preference in order to gauge which representations are easiest to learn and use. Finally, to go beyond a purely quantitative evaluation of user performance or preference, we collect written user feedback throughout the conducted study, whose qualitative analysis is used to contextualize and understand more deeply our statistical results.

In summary, the contributions of this chapter are:

1. an *overview* of ego network representations across application domains (Section 4.4),
2. a large-scale *user study* to investigate, both quantitatively and qualitatively, the effect of these representations on user performance and preference (Section 4.5), and
3. a *discussion* of the (conceptual) (dis)advantages of the four selected ego network representations to provide recommendations on their usage (Section 4.7).

4.3 Related Work

As we are interested in studying and comparing different layout approaches to ego network visualization, we deem it important to understand what systematic evaluations have been conducted both outside of and within the context of ego networks. Specifically, in this section, we discuss comparison studies between adjacency matrices and node-link diagrams, user performance and graph aesthetics of different node-link diagrammatic layouts, and various evaluations conducted within the context of ego network visualization.

4.3.1 Comparisons of Matrices and Node-Link Diagrams

In their seminal study, Ghoniem et al. [GFC04, GFC05] compare the "*readability*" of node-link diagrams and adjacency matrix representations: the more readable the graph, the faster and more accurately a participant can complete a series of low-level graph tasks. With the results of a completed user study, the authors conclude that node-link diagrammatic representations are more suited for smaller and sparser graphs, whereas matrix representations should be preferred for larger and denser ones, with the possible exception of path-tracing tasks, which proved difficult in both representations. Several follow-up works have further studied the differences between node-link diagrams and adjacency matrices across graph analysis tasks [LPP+06b, BHW+21], as references against some third novel hybrid representation [HF07, DGDMT21, GDL+24], and across application domains, from brain connectivity networks [ABHR+13] to terror networks [BSC13]. Here, we discuss the results of those follow-up studies for different graph tasks.

Adjacency tasks deal with the identification of a given node's immediate neighborhood, i.e., listing or counting its neighbors. Results differ depending on the size of the network under study. For larger networks, node-link diagrams appear to be favorable [OJK18, OJK19], whereas for smaller and medium-sized graphs results are less conclusive, with some finding adjacency matrices to be favorable [HF07, CBDM17], other finding node-link diagrams to be preferable [NWHL20, DKMT18], while others find no difference between the two representations at all [RMO+19, KEC06]. **Accessibility** tasks, closely related to adjacency tasks, concern themselves with the identification of incident edges between given nodes, e.g., checking whether two given nodes are indeed adjacent to each other. Contrary to Ghoniem et al.'s [GFC04, GFC05] (non-statistically significant) initial findings, subsequent work, across graph sizes, appears to point towards the superiority of node-link diagrams over matrices [OJ15a, OJK18, RMO+19, HBW14, DKMT18]. Indeed, only one paper, a replication of Ghoniem et al.'s original study, showcases the statistical superiority of adjacency matrices [OJ15b]. **Common connection** tasks deal with the identification of nodes adjacent to not just one, but two or more given nodes, e.g., locating the common neighbors of a set of given nodes. In their original work, Ghoniem et al. [GFC04, GFC05] were unable to showcase any statistical differences between the two representations. Here, results tentatively favor node-link diagrams: several works have shown the superiority of node-link diagrams for the identification of common connections [HF07, KEC06, OJ15a, RMO+19, DKMT18], while others have shown the

opposite [ABHR+13, OJK18]. Many were unable to find any statistically significant results [MOB+12, NWHL20, OJK19]. **(Shortest) path-finding** tasks challenge the user with tracing multiple possible paths between two given nodes and reporting the shortest one. Here, Ghoniem et al.'s [GFC04, GFC05] initial findings, i.e., the superiority of node-link diagrams over adjacency matrices, have been confirmed by effectively all follow-up studies [NWHL20, OJK18, OJK19, DKMT18]. Relatedly, **path-following tasks** have also been systematically shown to be more accurate using node-link diagrams [NWHL20, OJK18, OJK19]. Lastly, **overview** tasks deal, very generally, with gaining a big-picture understanding of a graph, such as identifying a graph's class [YDK+18] or estimating the graph's density [ASA+23]. In this case, while Ghoniem et al. [GFC04, GFC05] found adjacency matrices to be superior across all overview tasks, subsequent results have been mixed, with some finding node-link diagrams to be superior [ASA+23], others adjacency matrices [YDK+18].

In general, looking at the last twenty years of comparisons, not all of Ghoniem et al.'s [GFC04, GFC05] initial findings can be taken as fact. The collected 21 comparisons certainly highlight the superiority of node-link diagrams over matrix representations for path-finding, path-tracing, and accessibility tasks, both in terms of task accuracy and completion time. For topology or attribute-related tasks, it is difficult to draw any similarly sweeping conclusion. Still, for increasingly large and dense graphs, performance in node-link diagrams does appear to be more heavily affected than in matrices [OJK18, ASA+23].

4.3.2 Comparisons of Node-Link Diagram Layouts

Node-link diagrams can take many different shapes and forms depending on the particular layout approach. Classical node-link diagrams can be drawn using a wide variety of force-directed algorithms and spring embedders. Groups within these graphs can be highlighted by embedding them using group-based layout techniques or laid out radially or hierarchically to highlight other structures within the data. However, it is unclear for which situations certain graph layouts are to be preferred.

Early evaluations, such as the work of Blythe et al. [BMK96] and Purchase et al. [Pur98] went beyond **comparing classical layout algorithms** in terms of their produced graph aesthetic criteria and focused their efforts on studying their effect on human performance. More recently, Meulemans and Schulz [MS15] studied the effect of three different graph layout algorithms within the context of social networks on human perception. In general, while (statistically significant) differences between layouts can be detected, no meaningful differences between individual pairwise layouts can be identified (despite these layouts differing substantially in terms of their graph aesthetic metrics). **Beyond comparisons of classical** node-link diagram layout algorithms, research has focused on comparing, for example, different approaches to "sociogram" layouts, i.e., radial, hierarchical, group, and free layouts [HHE06], or tree representations to classical node-link diagrams [LPP+06a]. Once again, no meaningful conclusion regarding the impact of graph layout on performance can be drawn. Lastly, Didimo et al. [DKMT18] compared four different approaches to the

directed graph layout problem, namely, hierarchical, orthogonal, overloaded orthogonal layouts, and matrix representations. The results, however, indicated that, in terms of error rates (for the particular data studied), overloaded orthogonal drawings outperformed all other layout approaches.

In general, depending on the particular dataset presented, node-link diagram layouts indeed seem to have an impact on human perception, i.e., different layouts result in different task accuracies and completion times. However, it is unclear whether a single "best" layout can be identified, as pairwise statistical differences between layouts are often inconclusive. It is safe to conclude that any differences between such layouts are either fairly minimal or highly task/data-dependent.

4.3.3 Ego-Network Evaluations

Most commonly, ego network visualization evaluations have been conducted either as usage scenarios [LWB18, HMPI+16], or as case studies [FMW+21, AABS+14, TCG+22]. In some instances, additional early [EHAE16] or final user feedback [TCG+22, LSM+17, BHP+24] was also collected. However, task-based evaluations are featured in several papers. For example, within the context of **dynamic ego network visualization**, e.g., visualizing the evolution of such networks over time, several task-based evaluations have been conducted [SWW+15, LHS+15]. These evaluations naturally deal predominantly with time-dependent tasks, such as summarizing the evolutionary trend of clusters for a particular ego [LHG+20], identifying topological changes in the ego's connectivity [HZL+16], or determining whether the 1-alter subgraph increased or decreased in size over time [ZGC+16]. Alternatively, within the context of (dynamic) **comparative ego network visualization approaches**, such as Liu et al.'s *EgoComp* [LGD+17] or Wu et al.'s *EgoSlider* [WPZ+16], similar task-based evaluations can be found. In such cases, tasks focus on evaluating differences between egos, such as identifying which ego has the largest number of 1-alters with certain properties, characterizing the overall similarity between two selected egos, or inspecting whether a particular alter of one ego exists in the network of another. Most importantly, **non-dynamic task-based evaluations focusing on a single ego** can also be found in Shikora et al.'s *InfluViz* [SI17] and Sorger et al. [SAK+21]'s virtual reality ego network visualization. In such cases, participants completed multiple topological tasks derived from Lee et al.'s [LPP+06b] graph analysis taxonomy, such as finding common neighbors of two alter nodes, estimating the degree of a particular given node, or finding a (shortest) path between two given nodes. Unlike previous studies that have primarily focused on either quantitative or qualitative evaluations of ego networks, our study introduces a novel mixed-methods approach that integrates both methodologies. This allows us to provide a more comprehensive examination of differences between various ego network visualization techniques and a more nuanced understanding of how different representations impact the interpretation of ego network structures, compared to prior research.

4.4 Ego Network Representations

To better understand the current state of ego network visualization, we perform a literature survey, collecting 50 papers, comprising visualization techniques, systems, and application papers that feature ego network visualization approaches. A complication to this search is the lack of consensus on terminology to describe such networks. Examples include ego networks, egocentric networks, node-relative networks, or subject-relative networks. In other cases, such networks are not explicitly identified as ego networks at all [AABS+14]. Owing to these difficulties, the literature survey was conducted manually, driven by keyword searches across multiple academic search engines, and, in a snowballing method, combing through their bibliographies exhaustively. These collected works were then manually read and filtered to ensure relevance to the project. Each of the thus collected 50 references is categorized based on the visual representation featured as well as the domain's application area (see Figure 4.2). Ultimately, five common graph representations are identified, namely i) straight-line node-link diagrams (✕), ii) radial node-link diagrams (●), iii) trees / layered node-link diagrams (▲), iv) adjacency matrix representations (■), and v) latent variable space embeddings. However, as latent variable embeddings commonly omit drawing edges altogether, they are unsuitable for topology-based tasks that we investigate (Section 4.5.2). Therefore, we omit them from both this section as well as our user study. Given the large prevalence of weighted ego networks in literature, particularly in applications requiring precise association analysis, we focus on weighted representations for this study. Weighted networks offer critical insights into the strength and frequency of connections, which are pivotal in real-world tasks across diverse domains (see Section 4.3).

In this section, we briefly discuss the four remaining archetypal ego network representations that are commonly used as well as how they were designed and implemented. The implementation, as well as our classification of the collected papers, has been made available on the Open Science Framework[1].

4.4.1 The Straight-Line Node-Link Diagram

Straight-line node-link diagrams (✕) are arguably the most well-known graph representation for networks. For some given graph $G = (V, E)$, its nodes V are represented as points, circles, or rectangles placed freely in 2D space connected by straight line segments, representing its edges E [BETT98] (see Figure 4.3a). Such diagrams also proved to be the most popular form of ego network visualization, featured in 28 unique papers. In its simplest form, such ego-centric node-link diagrammatic representations lay out the graph's global topology and then (interactively) highlight egos and their 1-alters, as presented in Fisher et al.'s [Fis05] visualization of communication networks. In other cases, the ego and its incident edges are not explicitly visualized at all, and, instead, only the ego's 1-alters and their intra-1-alter connectivity are visualized, as discussed in the work of Ezaiza et al. [EHAE16], which focuses on various friendship clusters in

[1] https://osf.io/qzd9x/?view_only=20e4ffa7fedb4f9d897e3144b29d9f97

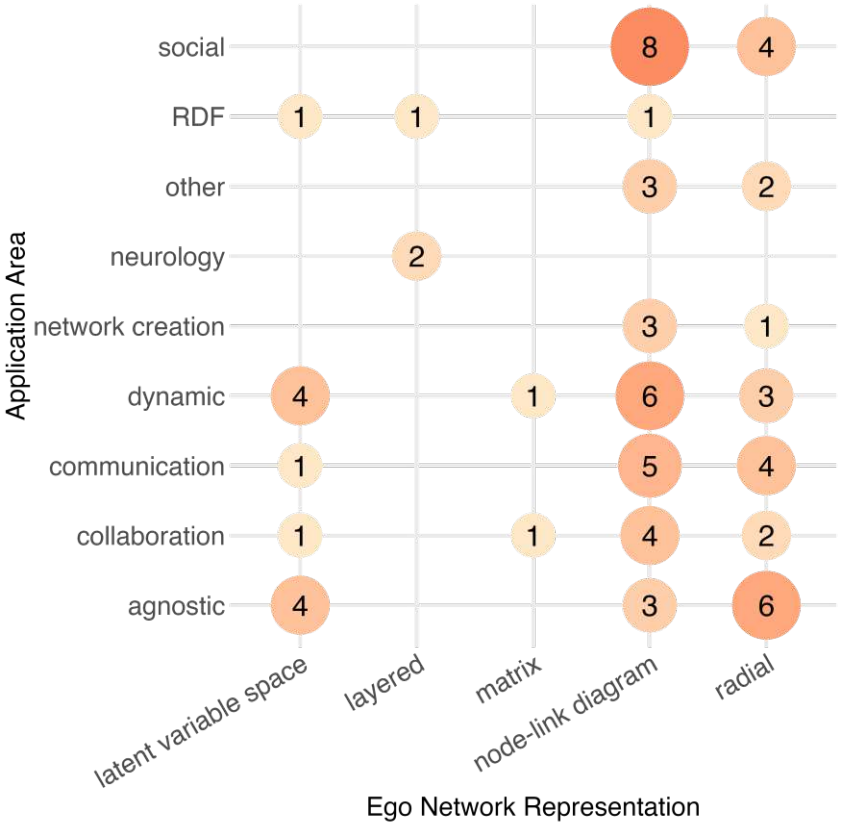
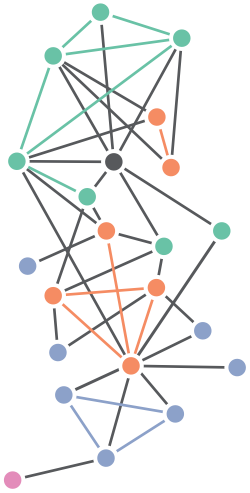


Figure 4.2: Mapping of the 50 collected ego network papers to both the application area and ego network representation. The number of papers that map to a combination of categories is encoded in its circle's size, color intensity, and numerical label. Papers could map to multiple representations and application areas.

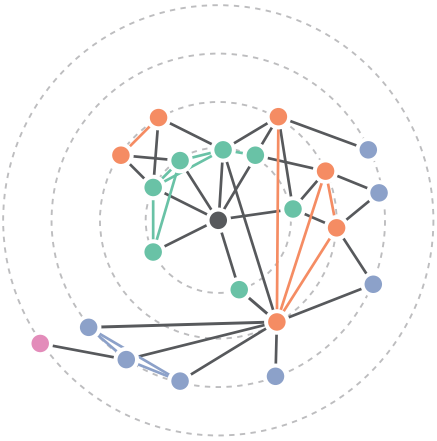
personal social networks. However, we note that such representations naturally do not allow users to investigate an ego-alter's edge weight. Here, edge weights were encoded in each edge's line segment's opacity, i.e., the greater the edge weight, the more opaque the edge. This ensures that edges, much like their node counterparts, remain the same size/thickness both within and across node-link diagrammatic representations. Lastly, opacity is also chosen to ensure comparability to the to-be-discussed layered adjacency matrix representation.

4.4.2 The Radial Node-Link Diagram

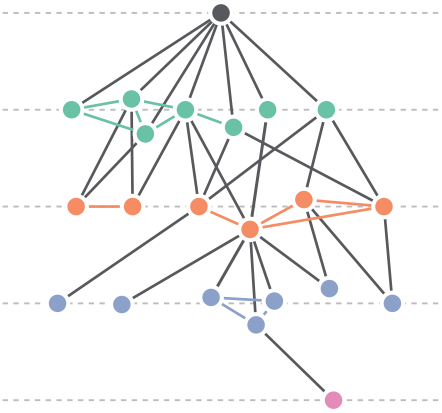
A radial node-link diagrammatic representation (●) is a more constrained form of the previously discussed straight-line node-link diagrams. Instead of allowing nodes to be drawn freely in 2D space, their placement is restricted to a given circle. Edges are then drawn either as straight line segments or smooth arcs, mostly within the confines of the



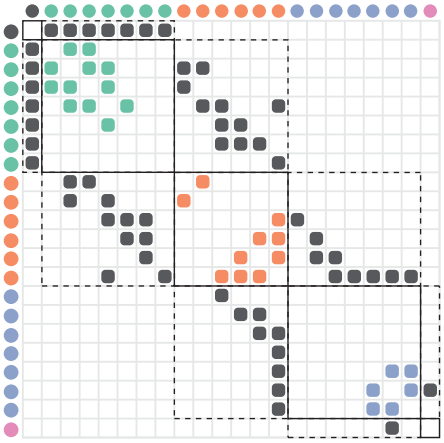
(a) Straight-Line Node-Link Diagram



(b) Radial Node-Link Diagram



(c) Layered Node-Link Diagram



(d) Layered Adjacency Matrix

Figure 4.3: Illustrative visualizations of the implemented ego network representations. Nodes and intra-alter edges are color-coded according to their alter-level using an appropriate color palette. More specifically, 1-alters are colored green, 2-alters orange, 3-alters blue, and 4-alters purple. Inter-alter edges, as well as the ego, are colored dark grey.

given circle. With 19 papers, radial node-link diagrams form the second most common representation of ego networks. In their simplest form, such radial representations display the ego at the center of the circle and its 1-alters radially around it, with both ego-alter and alter-alter edges displayed within the formed circle's area (see Figure 4.3b) [LWB18].

As discussed previously, ego-alter edges are sometimes omitted and only intra-1-alter connectivity is visualized [EHAE16]. Multiple alter-levels can also be visualized as multiple concentric circles, i.e., 1-alters placed along the first ring, 2-alters along the second ring, and so on. In such representations, intra-alter edges are commonly omitted and only inter-alter edges are rendered [LHG+20, LHS+15]. However, these concentric circles need not only denote different alter levels. For example, in the context of dynamic ego network visualization, each ring can indicate a particular time slice instead. An ego's various 1-alters' presences/absences are then indicated for each such time slice [FHQ11]. Lastly, for the sake of completeness, radial layouts are also utilized for the comparison of egos' 1-alters by, for example, juxtaposing their concentric alter rings and rendering ego-alter edges not only within each circle but also between them [CHWO13]. Again, edge weights are encoded using each edge's line segment's opacity.

4.4.3 The Layered Node-Link Diagram

Similar to radial node-link diagrams, layered node-link diagrams (▲) opt to restrict node placement instead of allowing them to be drawn freely in 2D space. Unlike radial diagrams, such representations arrange nodes not along the circumference of concentric circles, but instead along equidistant lines called layers. Edges are commonly represented as straight-line segments between them. For ego networks specifically, layered node-link diagrams are tree-like representations of ego networks, which are notably less popular than straight-line or radial node-link diagrams, with only 3 papers mapping to this particular category. Here, the aforementioned layers in layered ego network node-link diagrams encode the specific alter level, and nodes are placed along it accordingly (see Figure 4.3c). Consider Sayers et al.'s [Say04] RDF graph visualization, which provides a very representative example hereof. While in all examples found, only inter-alter edges were rendered, we opt to render all edges of the ego network to investigate their utility for both inter and intra-alter connectivity analysis. Edge weights are again encoded using each edge's line segment's opacity.

4.4.4 The Layered Adjacency Matrix

Unlike the previously discussed node-link diagrams, an adjacency matrix representation (■) takes the form of a data table, in which nodes are represented both as rows and columns. An undirected edge connecting two particular nodes is then represented by "filling" the two corresponding matrix cells of the symmetric table with a 1 (0 otherwise). For example, should nodes $v_i, v_j \in V$ be connected by some undirected edge $\{v_i, v_j\} \in E$, the matrix cells (i, j) and (j, i) are correspondingly "filled in". Here, in line with the conclusions of previous surveys of network visualization outside of the context of ego network visualization, only very few approaches make use of (adjacency) matrix representations [NMSL19, EMWR24]. Across the 50 collected ego network visualization papers, only one uses an adjacency matrix representation, namely Zhao et al.'s dynamic egocentric network representation *EgoLines* [ZGC+16].

In order to meaningfully exploit and visually display an ego network's topological connectivity as well as its various alter levels, and ensure it is comparable to both radial and layered node-link diagrams, we visualize these networks as *layered* adjacency matrices (see Figure 4.3d), akin to a centered matrix representation [SM07] or a quilt [BW11, BHR17]. In such representations, nodes are grouped by some layer structure (here their k-alter level). This ensures that all intra-k-alter connectivity is displayed only within a block of k-alter nodes. Additionally, because a k-alter can, by definition, only be connected to $(k-1)$ or $(k+1)$ alters, all inter-alter connectivity is displayed within blocks adjacent to their corresponding intra-alter blocks. This, in turn, results in larger portions of the off-diagonal elements remaining empty, resulting in overall less visual clutter. To provide a meaningful one-dimensional ordering of the nodes (an important consideration when evaluating adjacency matrix representations [RMO+19, OJK19, DKMT18]), nodes are sorted according to their graph-theoretic distance to the ego within each alter level. Hop-distance ties within a block of k-alters were broken using their weighted distance to the ego, i.e., the closer their graph-theoretic distance, the closer their visual distance to the ego within their corresponding k-alter block. In line with the previously discussed node-link diagrammatic representations' visualization of edge weight, we opt to encode edge weight using opacity. More specifically, the greater the edge weight of some undirected edge between nodes v_i and v_j, the more opaque the corresponding matrix cells (i, j) and (j, i).

4.4.5 Implementation

For our implementations, many different spring-embedded and force-directed approaches to laying networks as node-link diagrams present themselves [Kob12]. For the straight-line, radial, and layered node-link representations, we make use of *D3js*'s [BOH11] particle-based force-directed algorithm to lay out the networks, as it produces consistently visually pleasing results while remaining computationally tractable for the kinds of undirected, weighted graphs we consider here. It is important to note that while the four node-link diagrammatic representations may, in certain circumstances, share visual similarities, they should never look identical to each other, thereby distinguishing themselves from each other both conceptually and visually. The layered adjacency matrix was also implemented in *D3js*. Additionally, across all representations, in order to visually communicate the alter levels, nodes, and intra-alter edges are color-coded using an appropriate *Color Brewer* [HB03] palette (see Figure 4.3). Specifically, 1-alters are colored **green**, 2-alters **orange**, 3-alters **blue**, and 4-alters **purple**. Inter-alter edges, as well as the ego, are colored dark grey. Similar to previous (ego) network visualization evaluations, we also implemented basic interactivity [RMO+19, NWHL20, OJK19]. Interactions include highlighting (in red) incident edges and adjacent nodes when hovering over a node for all node-link representations. For the adjacency matrix, when hovering over a cell (edge) at location (i, j), row i, column j, and nodes $v_i, v_j \in V$ are also highlighted in red. Additionally, basic navigation, i.e., panning and zooming, is provided. We refer the reader to the previously mentioned OSF repository for visual examples of the implementation as presented to study participants.

4.5 Study

We aim to evaluate the ego network representation's effect on user *performance* and *experience* in a mixed methods approach in order to better understand and contextualize our quantitative results. Given that ego networks are highly specialized tools, a group of expert users would be optimal to test these effects, such as network scientists or domain experts, e.g., social scientists or biologists. However, to achieve adequate statistical power and sufficient numbers of qualitative feedback, we instead opt to conduct an online, crowd-sourced user study.

First, we wish to quantitatively evaluate the effect of ego network representation on participants' performance, i.e., their ability to complete a series of low-level graph analysis tasks as quickly (response times) and correctly (accuracy) as possible. To do so, we employ a between-subjects study design, in which each participant completes six low-level graph analysis tasks for one randomly assigned archetypal representation. Here, a between-subjects study design was selected to ensure the study could be completed in less than 30 minutes in an online setting, which (based on our prior experiences with online studies) should be considered the upper limit for such online studies. Requiring each participant to complete all six tasks for all four representations would have i) introduced a possible learning effect, and ii) taken too long, thereby affecting participant concentration. To curb any further systematic learning effects in our results, the order in which participants are asked to complete these six tasks is randomized, and the presented graph's nodes' labels are randomized for each task, mitigating memorization.

Second, we aim to quantitatively evaluate the effect of these representations on user experience. To assess this aspect, each participant is presented with five rankable statements, relating to the ease of use and learning, participants' perceived accuracy and efficiency, as well as the aesthetic appeal of the presented visualization, which they are required to answer on a 5-point Likert scale.

Third, we aim to enrich these quantitative analyses with additional qualitative data. So, after each of the tasks given to a user, we present the task's description again and ask the users to provide a short comment on how the assigned ego network representation assisted or hindered them in the task performed. Finally, at the end of the survey, we collect participant feedback pertaining to their final thoughts about the assigned ego network representation.

4.5.1 Graph Data

Similar to the evaluation conducted by Okoe et al. [OJK18], we opt to investigate a single network; more specifically, the real-world network representing the "*Les Misérables*" character interaction graph, consisting of $|V| = 77$ nodes and a total of $|E| = 254$ undirected edges, in which edge weights encode the number of times two particular characters co-occurred [Knu93]. To simulate the process of creating an ego network from a larger network, with "*Valjean*" as the selected ego, an ego network is constructed

from this graph by computing the shortest hop-distance paths from the ego to each other node. Ties in path length are broken by their weighted distances. Labels were randomly generated integers to ensure that, even though a user was investigating the same network, the network's labels were never the same across the various tasks investigated (Section 4.5.2). Here, this particular network was chosen for six key reasons. First, by utilizing a real-world network, we avoid non-representativeness often associated with simulated (scale-free) graph data [BC19]. Second, by focusing on a single network, we are able to statistically investigate a larger number of tasks relevant to ego networks without sacrificing statistical power. Third, depending on the particular ego chosen, here "*Valjean*", said dataset features a larger number of alter levels than most investigated data while remaining overall representative: while most (31) ego networks focused on either 1 or 2-alters, several (7) also studied ego networks of 3-alters and higher. Fourth, by evaluating the size of ego networks investigated in previous approaches and applications, we determined this particular dataset's size to be slightly larger than the median graph size studied in ego network literature while remaining representative (see Figure 4.4). Moreover, a graph of this size and density was deemed appropriate for the non-expert user group that would be tasked with analyzing it. Fifth, in order to build on work done previously, this network has already found use as a benchmark in both ego network [LGD+17] as well as more general network visualization evaluations [DKMT18]. Lastly, with $|V| = 77$ nodes and a total of $|E| = 254$ undirected edges, this particular network falls nicely within the most commonly explored/visualized sizes of graphs [YAD+18], thereby further ensuring it is representative in terms of its numbers of nodes and edges.

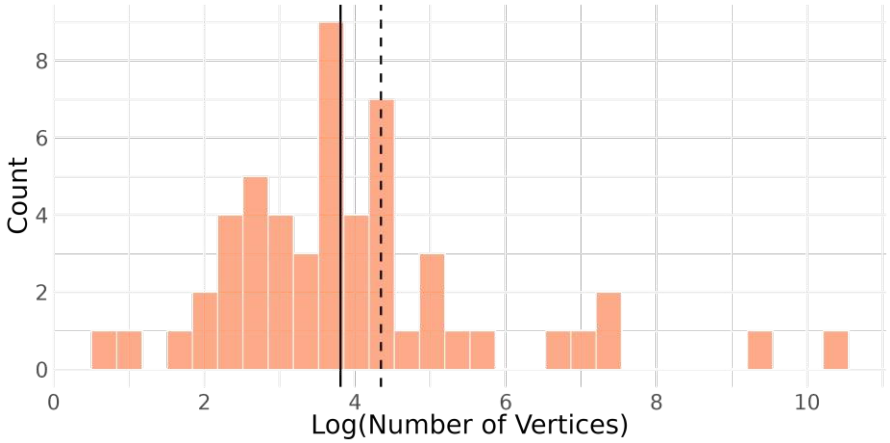


Figure 4.4: Studied network sizes of 53 ego-networks from our gathered 50 papers. For several papers, no specific sizes were specified, nor could they have been estimated from the figures. Network size is displayed on the log scale for readability. The overall median network size, $|V| = 45$, is illustrated by the dashed black line, and the chosen "*Les Misérables*" character interaction graph, $|V| = 77$, by the dashed black line [Knu93].

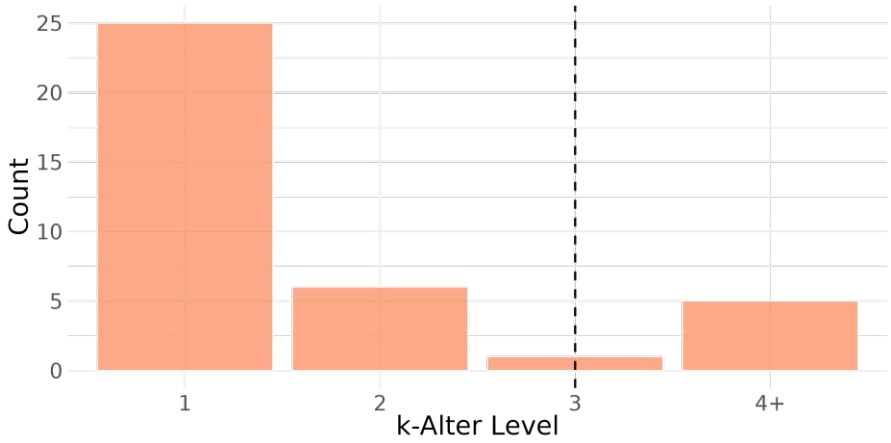


Figure 4.5: Counting of the k-alter levels admitted or demonstrated in 37 of the 50 collected ego network visualization papers. Our chosen "*Les Misérables*" character interaction graph's alter-level of $k = 3$ is illustrated by the dashed black line, chosen as it offered a balance between the relative simplicity of the many $k = 1$ or $k = 2$ networks and the still fairly common larger networks of $k \geq 4$.

4.5.2 Procedure

To conduct our mixed methods analysis of user experience and performance, we must ensure that participants are properly instructed and trained before they commence a series of graph analysis tasks and a final user experience evaluation.

Training In line with other comparative network visualization studies, we employ pre-study training to familiarize participants with the upcoming representations and tasks. Following the definitions of Nobre et al. [NWHL20], we use a mixture of active and passive training. Each participant was presented with both i) a written tutorial explaining the details of ego networks and the assigned network representation to familiarize them with concepts and terminology, as well as ii) an interactive visualization of a simple network using the network representation they could expect during the evaluation to familiarize themselves with the modes of interaction we provide. Key definitions remained available to participants in a small glossary throughout the study to avoid terminology-related errors.

User Performance: Tasks Following existing quantitative (ego) network evaluations (Section 4.3.3), we are interested in statistically comparing how quickly and accurately participants were able to complete a series of low-level graph analysis tasks using the different representations we provide (see Section 4.4). However, a key challenge here lies in the selection of a set of tasks: they must remain general enough for their results to transfer to other network types, yet be specific enough to be meaningful for ego networks specifically. We derive our set of tasks from existing ego network evaluations. Of the 50

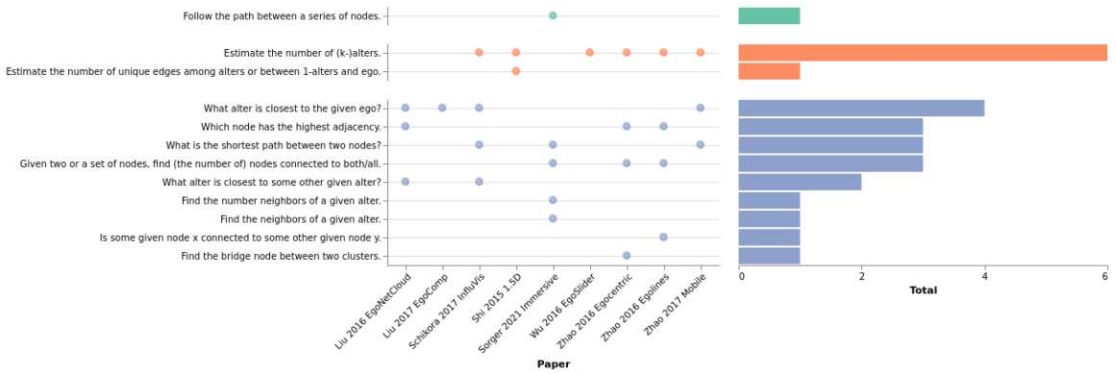


Figure 4.6: Categorization of (non-dynamic and attribute-related) graph analysis tasks according to nine previous works that featured a task-based evaluation of ego networks (i.e., [LHS+15, LGD+17, LGD+17, SI17, SWW+15, SAK+21, WPZ+16, ZGC+16, ZGC+16, ZSZ+17]. i.e., i) Liu et al. [LHS+15]'s study of their EgoNetCloud system for dynamic ego network analysis, ii) Liu et al.'s [LGD+17] visualization approach for the comparison of two ego networks, iii) Shikora et al.'s [SI17] egocentric visualization of artists' relationships to each other, iv) Shi et al.'s [SWW+15] 1.5D visualization of dynamic ego networks, v) Sorger et al.'s [SAK+21] visualization of egocentric networks in VR, vi) Wu et al.'s [WPZ+16] EgoSlider approach to visualizing egocentric network evolution, vii) Zhao et al.'s [ZGC+16] investigation of their matrix-based visualization of dynamic ego networks, viii) Zhao et al.'s [ZGC+16] evaluation of their dynamic adjacency matrix-based technique EgoLines, and ix) Zhao et al.'s [ZSZ+17] evaluation of EgoSlider [WPZ+16] on mobile devices. Additionally, the twelve tasks are categorized, faceted, and color-coded according to Lee et al.'s [LPP+06b] graph task taxonomy, i.e., **Browsing Tasks**, **Overview-based Tasks**, and **Topology-based Tasks**.

collected papers, nine featured a quantitative, task-based evaluation of their proposed techniques.

From these evaluations, we identify twelve common tasks when placed within Lee et al.'s [LPP+06b] low-level graph task taxonomy: one browsing task, two overview tasks, and nine topological tasks (Figure 4.6). More specifically, these twelve tasks include:

1. following the path between a series of nodes,
2. estimating the number of (k-) alters,
3. estimating the number of unique edges among alters or between 1-alters and ego,
4. identifying the alter closest to the ego,
5. identifying the node (ego or alter) with the highest adjacency,
6. identifying the shortest path between two nodes,
7. identifying the alter closest to some other alter,
8. identifying the common neighbors of two given nodes,
9. counting the neighbors of a given alter,
10. identifying the neighbors of a given alter,

11. determining whether an edge between two nodes exists, and
12. finding bridges between clusters.

It should also be noted that several of these dynamic egocentric network visualization system evaluations [HZL+16, LZH+17, WPZ+16, LHG+20] also featured extensive time-dependent tasks, such as estimating changes of alter numbers between time steps [HZL+16] or estimating the number of relationships that lasted for a certain number of time steps [ZGC+16]. However, as the scope of this chapter is explicitly related to investigating differences in the perception and execution of topological ego-centric tasks across different ego network representations, we exclude tasks that focus on either the temporal or attribute-based nature of data. We argue that before one investigates tasks related to more complex multivariate or dynamic graph data, potentially requiring more sophisticated interaction techniques and training, it is necessary to first understand the effect of ego network representation in the context of relatively simpler and mostly static graphs. Therefore, from the thirteen common tasks, we selected the six most common to investigate in our user performance study [LBI+11], i.e., two overview tasks, two topological adjacency tasks, one topological common neighbor task, and one topological association task.

Overview tasks [LPP+06b] can take many different shapes and forms, such as the identification of a network's graph class [ASA+23, YDK+18], the counting or approximation of a network's nodes and/or edges [GFC04, GFC05], or the estimation of a network edge density [ASA+23]. Based on the ego network-specific overview tasks we identified in the literature, we investigated two such tasks:

$\mathbf{T_{edges}}$	Count the intra-alter edges between all 1-alters.
$\mathbf{T_{alters}}$	Count the number of 2-alters in the graph drawing.

Adjacency tasks focus on the immediate adjacency of a node, i.e., its neighbors, and can take several forms, such as counting a given/highlighted node's neighbors [OJ15b] or edges [KEC06], counting incoming and outgoing edges separately [ASA+23], or finding the most connected node [GFC04, GFC05], the highest degree node [CBDM17], or the node with the highest number of edges [HF07]. Based on our literature review, we investigated tasks:

$\mathbf{T_{neighbors}}$	Count the neighbors of a given alter.
$\mathbf{T_{degree}}$	Which 2-alter has the highest degree of all 2-alters?

Common neighbor tasks involve the identification of two (or more) nodes' common neighbors and subsequently listing them by their identifier [HF07, RMO+19] or counting them [KEC06]. Here, we specifically investigated the former:

$\mathbf{T_{common}}$	Which neighbors do two alters have in common?

Finally, **accessibility** tasks [LPP+06b] involve the identification of incident edges between nodes, such as finding the edge between two given nodes [GFC04, GFC05], determining

whether a node is connected to another in two hops or fewer [RMO+19], or determining whether two given nodes are connected or not [OJK18]. Based on our literature review, we specifically tasked participants with:

$\mathbf{T_{association}}$	Which 1-alter's edge to the ego has the largest weight?

Note that for tasks T_{degree} and T_{alters}, $k = 2$ was chosen to ensure there was a layer of alters, i.e., $k = 1$ and $k = 3$, surrounding it to better capture the visual complexity of arbitrary-sized ego networks. Moreover, it should be noted that despite its frequent occurrence in ego network evaluations (Figure 4.6), we opted to ignore shortest path identification tasks owing to the by now well-documented superiority of node-link diagrams over adjacency matrix representations [NWHL20, OJK18, OJK19, DKMT18] which may bias our results. Additionally, as there always exists a path between two nodes through the ego, therefore, the (shortest) path lookup becomes a trivial task.

User Experience: A Likert Scale In order to probe each participant's perceived and subjective effectiveness and preferences [LBI+11], at the end of the study, similar to existing ego network evaluations [ZGC+16, LHG+20], each participant is required to answer five statements on a 5-point Likert scale:

$\mathbf{S_{learn}}$	I found the ego network's visual representation easy to learn.
$\mathbf{S_{use}}$	I found the ego network's visual representation easy to use.
$\mathbf{S_{pleasing}}$	I found the ego network's visual representation aesthetically pleasing.
$\mathbf{S_{accurately}}$	I found the ego network's visual representation allowed me to answer questions accurately.
$\mathbf{S_{quickly}}$	I found the ego network's visual representation allowed me to answer questions quickly.

Qualitative Feedback: Participant Comments Finally, to go beyond a purely quantitative evaluation, we additionally probe both user performance and experience qualitatively. To do so, participants were required to provide written feedback after each completed task as well as at the very end of the survey, together with their Likert-scale-ranked experience. Specifically, we asked for feedback on how the representation was helping with or hindering each task's completion and the participant's general experience with the representation.

4.5.3 Analysis

The collected data must be evaluated both statistically and qualitatively. To do so, we formulate a series of hypotheses regarding our quantitatively evaluated metrics, i.e., task accuracy, task completion time, and user preference. Subsequently, we outline how the

statistical analysis is conducted. Finally, we outline how the coding of the qualitative analysis is performed.

Hypotheses Based on comparisons of node-link diagrams and adjacency matrices (Section 4.3.1), as well as previous comparisons of ego networks (Section 4.3.3), we formulate four hypotheses regarding user performance and experience.

H_1: Counting on Matrices The overall findings in the literature seem to tentatively point towards the superiority of adjacency matrix representations over node-link diagrams for non-path-tracing/finding and accessibility tasks, especially for larger and denser graphs [ASA+23, GFC05]. Subsequently, though we do not study the densest and largest graphs possible to not overwhelm non-expert users, we hypothesize that our layered adjacency matrix representation should result in higher accuracy than the three node-link diagrammatic representations for such tasks, i.e., i) counting the number of edges (T_{edges}) and 2-alters (T_{alters}), ii) estimating the 2-alter with the highest degree (T_{degree}), as well as iii) counting the number of neighbors of a given alter ($T_{neighbors}$). However, given past results comparing node-link diagrams and adjacency matrices [NWHL20, RMO+19, JJHS+22], we also hypothesize that this increased accuracy will require additional time.

H_2: Node-Link Nuance While past evaluations comparing different types of node-link diagrams have not shown meaningful differences [BMK96, PCJ96, MS15], we expect the inherently layered structure of ego networks to tease out differences between the three node-link diagrammatic representations for the investigated overview and adjacency tasks. Specifically, given the 1D arrangement of nodes in both radial and layered node-link diagrams, we anticipate said representations to outperform the classical node-link diagram for the 2-alter-counting task (T_{alters}). For layered node-link diagrams, especially, the clarity of this 1D arrangement of nodes comes at the cost of effectively visualizing intra-alter edges. Subsequently, we anticipate the layered node-link diagram to perform worse than the radial or classical node-link diagrams for the intra-alter edge counting task (T_{edges}). Lastly, while disadvantageous for node/edge counting tasks, the unbounded embedding of the classical node-link diagram should communicate the neighborhood of individual nodes more effectively. As a result, we anticipate this representation to lead to more accurate results than the radial or layered node-link diagrams for the two tasks related to node neighborhood, i.e., T_{degree} and $T_{neighbors}$.

H_3: Topologically Topsy-Turvy Given the findings of individual papers pointing tentatively towards the superiority of node-link diagrams for certain topological tasks [GFC05, RMO+19], we anticipate the three node-link diagrammatic representations to outperform the layered adjacency matrix for the two topological tasks under study, i.e., locating the ego's most closely associated alter ($T_{association}$) and identifying the common neighbors of two given alters (T_{common}). Given this difficulty, we additionally hypothesize that the matrix representation will require participants to spend more time answering

these tasks. Within these three node-link diagrammatic representations, however, we anticipate the straight-line node-link diagram to outperform the other two for task $T_{association}$ in particular, as closely associated nodes can be placed closer to each other in 2D space, whereas layered and radial representation limit node placement along layered lines or concentric circles, respectively.

H_4: Where the Instruction Booklet at Looking at past studies comparing node-link diagrams and adjacency matrix representations, it would appear as though i) node-link diagrams are easier to learn than adjacency matrices [NWHL20, RMO+19], and ii) participants might be more familiar with node-link diagrammatic representations *a priori* [NWHL20, ASA+23]. Subsequently, we hypothesize that participants will find the three different types of node-link diagrams easier to learn (S_{Learn} and S_{Use}) than the adjacency matrix representation. Additionally, owing to the additional layers of complexity they presented, radial and layered node-link diagrams are hypothesized to be more difficult to learn compared to the straight-line layout. However, as these additional layers should allow for certain tasks to be answered more easily, we anticipate them to be easier to use.

Quantitative Evaluation We collect each participant's task answer, task completion times, and user experience questionnaire results. Specifically, for each task, participants' performance was measured using the number of errors made based on their given answers. For example, for T_{alters} or T_{edges}, the error is defined as the difference between the true number of such nodes/edges and the participant's answer. Alternatively, for task $T_{association}$, i.e., identifying the most strongly associated 1-alter with the ego, the error is defined as the difference in edge weights between the participant's answer and the ego, and the ego and the actually closest node. To statistically analyze the impact of ego network representation on these three quantities, we employ Wobbrock et al.'s [WFGH11] non-parametric aligned rank-transformed ANOVA, as standard assumptions of normality could neither be made nor validated when probed with Shapiro-Wilk tests. The overall statistical significance of ego network representations on these quantities is first probed with an omnibus F-test, which, if significant, is followed by a series of pairwise t-tests between individual representations using the computed estimated marginal means [EKHW21], both at an *a priori*, Bonferroni-adjusted family-wise type-I error rate of $\alpha = 0.05$ [Bon36].

Qualitative Evaluation We collect qualitative feedback from participants in the form of open questions after every task and summarily at the end of the session. Given the 120 participants, six task-related questions, and one summary question, 840 qualitative comments are to be collected. These comments are then broken up further into individual utterances. We analyze these utterances in inductive and deductive coding sessions [Chi97], with three independent coders. In the first inductive coding session, each coder assigns a single concept to each utterance. These are ultimately unified into a single set of unique codes. In a second, deductive coding session, every coder assigns one of the thus agreed-upon codes to each utterance. Coders also assign a positive or negative qualifier

to every utterance. In a final discussion, each utterance's coding is discussed until 100% consensus is reached.

4.6 Results

In this section, we enumerate the quantitative and qualitative results collected on user performance and experience.

4.6.1 Participants

In order to evaluate the impact of ego network representation on both graph analysis task completion time and accuracy, as well as user experience, a large-scale, interactive online user study was conducted on the user recruitment platform Prolific [noa14]. Each participant was randomly assigned to one of the four representations, provided said representation had not already received its total number of participants, guaranteeing a balanced number of participants per representation. In total 120 participants, i.e., 30 per representation, were recruited, of which 60 identified as "female", 59 as "male", and 1 as "other". Each participant was paid 10£per hour, slightly above Prolific's recommended hourly rate of 9£per hour. In terms of age, 38 participants were between the ages of $21-25$, 42 between $26-30$, 21 between $31-35$, 7 between $36-40$, 3 between $41-45$, 2 between $46-50$, and 6 older than 50. Given the academic bent of both the conducted literature survey as well as the selected tasks, we opted to (insofar possible) target a non-layperson participant group, i.e., people who had already completed or were in the process of completing some form of higher education. As such, 8 participants were in the process of completing their bachelor's degree, 75 participants had completed their bachelor's degree, and 37 had completed their master's degree. Finally, in terms of self-described previous experience with graph visualizations and analysis, 34 participants reported "no experience", 36 "little experience", 34 "some experience", and 16 "good experience". When probed statistically using Wobbrock et al.'s previously discussed *ART-ANOVA* [WFGH11], no significant association between the expertise of users and their performance was found.

4.6.2 User Performance

Accuracy For each task, participants' performance was measured using the number of errors made, depending on the user's answers and the particular task. Six answers in total had to be manually removed, such as lists of nodes when the question only asked for one, or long textual descriptions that did not actually provide an answer. The calculated error rates are depicted in Figure 4.7. Here, of the six tasks analyzed, we found the impact of ego network representation on performance to be statistically significant for four of them, namely T_{common}, $T_{neighbors}$, T_{edges}, and T_{alters}. For these statistically significant tasks, representations were then compared pairwise. Here, for T_{common}, we observe adjacency matrices (median(■) = 0.35) to produce statistically significantly worse performance than both straight-line (× = 0.0) and radial node-link (● = 0.06)

representations. For $T_{neighbors}$, the adjacency matrix (■ = 0.06) was also statistically significantly less accurate compared to all other representations (✕ = ● = ▲ = 0.0). For T_{edges}, participants using adjacency matrices (✕ = 0.7) were statistically significantly less accurate than those using straight-line (✕ = 0.33) and layered node-link diagrams (▲ = 0.2). Finally, despite notable differences observable visually in Figure 4.7 for T_{alters}, the only statistically significant pairwise differences observed were between adjacency matrices (■ = 0.1) and layered node-link diagrams (▲ = 0.0).

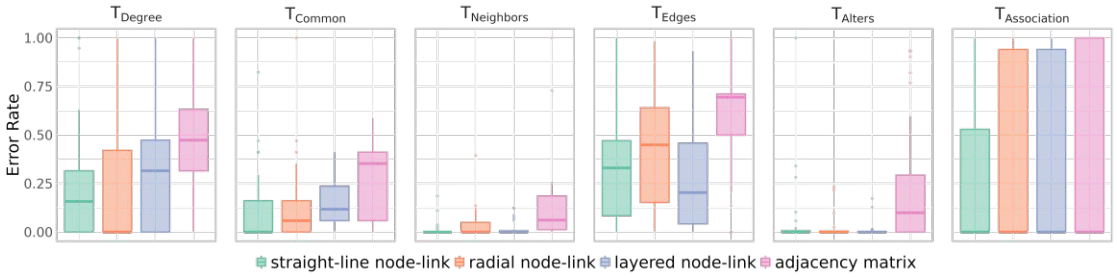


Figure 4.7: Participants' task error rates are visualized per task and represented as a box-and-whisker plot. The center of each boxplot corresponds to the median, and its lower and upper hinges to the first and third quartiles, i.e., the 25th and 75th percentiles, respectively. Finally, the "whiskers" correspond to $1.5 \times IQR$ from the hinge, where IQR denotes the inter-quartile range, i.e., the distance between the first and third quartiles.

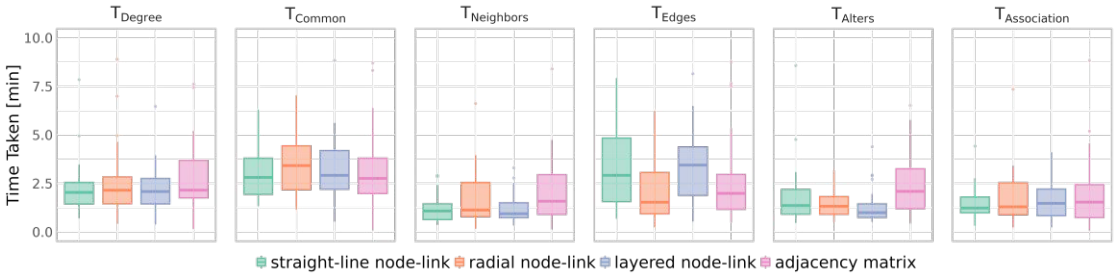


Figure 4.8: Participants' time taken per task is visualized per task and represented as a box-and-whisker plot. The center of each boxplot corresponds to the median, and its lower and upper hinges to the first and third quartiles, i.e., the 25th and 75th percentiles, respectively. Finally, the "whiskers" correspond to $1.5 \times IQR$ from the hinge, where IQR denotes the inter-quartile range, i.e., the distance between the first and third quartiles. For added visual clarity, the y-axis scale was limited to a maximum of 15 minutes, resulting in nine outlying data points not being visualized here.

Time Taken The time needed to complete the survey varied between participants. At the extremes, the fastest participant completed the entire tutorial and survey in 9 minutes, whereas the slowest took 90 minutes. Most participants, i.e., 50%, however, took between 20 and 35 minutes, with a median time taken of 25 minutes. More importantly, we also track the time taken for each participant to complete each task, excluding instructions

and training. Here, differences in time taken per task are fairly small Figure 4.8. The only task for which the ego network representation proved to have a statistically significant impact was T_{Alters}. In the subsequently performed pairwise comparisons, only matrices were statistically significantly slower than the layered node-link diagram.

4.6.3 User Experience

As discussed previously, at the end of the study, each participant was required to answer five statements on a five-point Likert scale. The results can be seen in Figure 4.9. Here, the effect of ego network representation was only found to be statistically significant for two of these five statements, namely S_{learn}, i.e., the ease of learning a particular ego network representation, and S_{use}, i.e., the ease of using an ego network representation. For these two statements, the marginal means of each representation are pairwise compared. For S_{learn} participants rated matrices statistically significantly lower than all three node-link diagrammatic representations. For S_{use}, adjacency matrices were rated statistically significantly lower than both layered and straight-line node-link diagrams. Please note that, as $S_{pleasing}$, $S_{accurately}$, and S_{learn} were not found to be statistically or substantively significant, they will not be further discussed in the results.

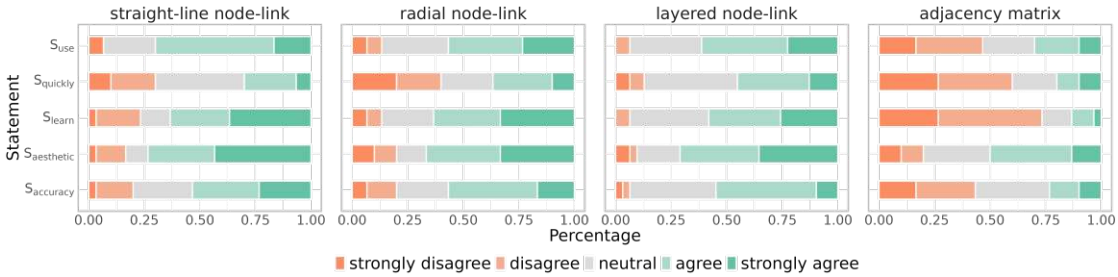


Figure 4.9: 5-Point Likert Scale results visualized as a normalized stacked bar chart. Each statement corresponds to a particular row, across the individual subplots corresponding to the four archetypal ego network representations investigated. Likert scale answers are color-coded, i.e., **strongly disagree**, **disagree**, **neutral**, **agree**, and **strongly agree**.

4.6.4 User Feedback

We collected user feedback per task and summarily after all tasks were completed, resulting in a total of 840 comments, which were ultimately broken down into a total of 1031 utterances. In the first inductive coding round, the three coders independently identified 30, 75, and 73 separate concepts. These emergent concepts were subsequently unified into 17 unique codes, and categorized into five broader classes, namely i) layout, ii) task load, iii) interactivity, iv) comprehension, and v) graph tasks. In the subsequent deductive coding step, the coders individually assigned one of the 17 agreed-upon codes to each of the utterances, ignoring utterances that held no value or were incomprehensible. Additionally, each utterance was also labeled as either positive or negative, denoted here as (***pos***, –***neg***). Ultimately, in a final meeting, the three coders discussed their choices

for each utterance until a consensus for every one of the 837 remaining utterances was reached, the results of which are presented in Figure 4.10.

Layout, the largest class of codes with 299 total utterances, contains sentiments related to the four visual embeddings, i.e., the use of "*Color*" for nodes and edges, the chosen "*Edge Weight Encoding*", node "*Labels*" and "*Layers*", as well as the representations' "*Edge Placement*" and "*Node Placement*". "*Colors*" (105, −5) and "*Edge Weight Encoding*" (56, −7) used in the graphs were regarded generally positive. "*Node Placement*" (7, −19), "*Edge Placement*" (8, −47), and "*Labels*" (3, −5) were often mentioned in connection with visual clutter. Representations using "*Layers*" to separate alter levels were often mentioned in a positive manner (27, −1). Interestingly, the "*Visual Appeal*" of individual representations was sparsely mentioned (8, −1).

Task Load was broken down into three codes derived from the well-known NASA TLX questionnaire [HS88]. Notably, however, "*Physical Demand*", "*Effort*", and "*Performance*" were omitted as utterances related to them could not be identified. In total, 215 utterances related to task load were identified. The largest portion of these utterances describes the "*Cognitive Demand*" that the participants perceived (104, −78) when performing the six low-level graph tasks. The remaining comments in this category describe the participants' subjective "*Time Demand*" (3, −16), or general "*Frustration*" with the task and representation (3, −11).

The **Comprehension** class consists of codes relating to the participants' general experience in the study. In total, 126 utterances are mapped to this class. Several utterances highlighted participants' (lack of) "*Understanding*" as a result of the representation (47, −50). Several participants lamented problems with the clarity of the "*Instructions*" (6, −15), as well as the insufficiency of the "*Training*" (−8) that was provided.

The **Graph Tasks** class comprised codes referring to challenges with specific low-level graph tasks, totaling a sum of 62 utterances. Several participants had problems with the "*Comparison*" of two nodes' adjacencies (6, −33), the "*Localization*" of specific nodes in the graph drawing (−12), or the "*Counting*" of nodes and edges (5, −6).

Finally, the **Interaction** class was made up of solely a single code, namely the interactive "*Highlighting*" of nodes and, in the case of the adjacency matrix, edges. In total, 135 utterances are mapped to this particular class. In general, comments were overwhelmingly positive (124), with only a few comments left complaining about the implementation of interaction (−11).

4.7 Discussion

In the following section, we discuss the quantitative and qualitative differences between straight-line node-link diagrams (×), radial node-link diagrams (●), layered node-link diagrams (▲), and adjacency matrices (■) within the broader context of related literature.

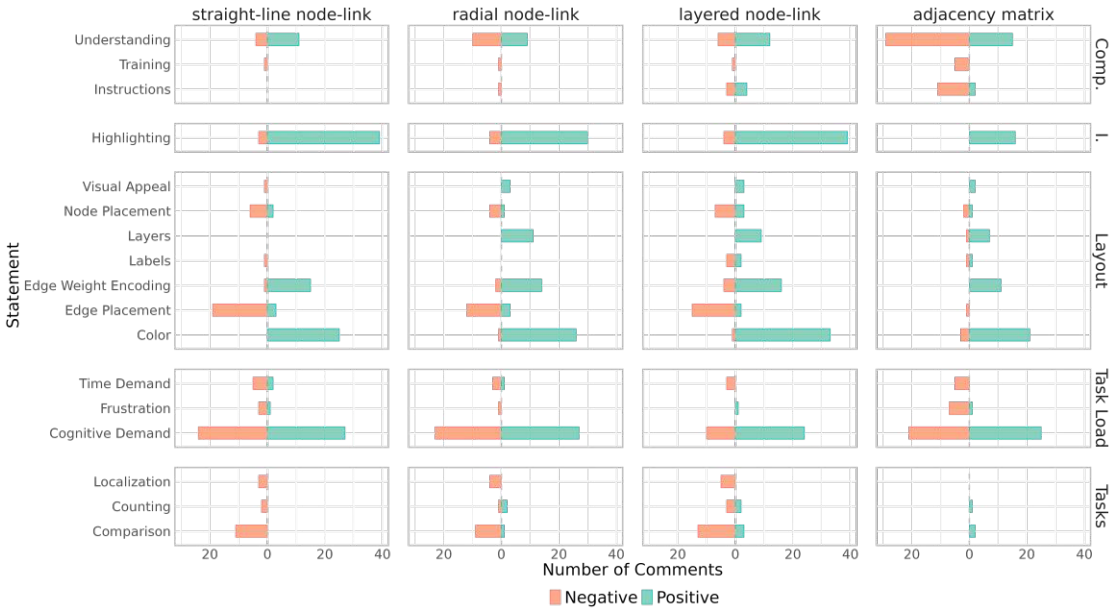


Figure 4.10: The number of negative and positive statements made by participants relating to i) comprehension, labeled as "Comp.", ii) nodes of interaction, labeled as "I.", iii) the layout, iv) participants' perceived task load, and v) particular graph tasks, labeled as "Tasks", organized by each of the four ego network visualizations. Comment counts by statement type are color-coded based on whether they are **negative** or **positive**.

4.7.1 The Layered Adjacency Matrix

As previously discussed, (conflicting) results of previous quantitative comparisons of participants' performance between adjacency matrices and node-link diagrams indicate that i) observed differences are highly task-dependent, and ii) for many tasks, no clear "winner" can be determined. Here, we discuss this study's observed differences in participant performance across the six tasks investigated. Additionally, we also discuss the differences in understanding that participants exhibited between adjacency matrices and node-link diagrams.

Adjacency and Location Tasks The tabular nature of adjacency matrices, i.e., its linear arrangement of nodes and non-obscured display of edges, has been speculated (and partially shown) to make certain tasks easier, namely the localizing of nodes, based on identifier and degree, as well as the general estimation of a node's degree and the counting of edges, respectively [GFC04, GFC05]. It is subsequently surprising that our results do not agree with the results of related work on adjacency matrices. Specifically, for the overview (T_{edges} and T_{alters}) and adjacency (T_{degree} and $T_{neighbors}$) tasks, we hypothesized that the tabular nature and layered grouping of alter-levels in the adjacency matrix representations would be advantageous. Participants using layered adjacency matrices either performed equivalently to those using node-link diagrams (T_{alters}) or

systematically statistically significantly worse (T_{edges}, T_{degree}, and $T_{neighbors}$). These quantitative results indicate that we are unable to produce the evidence necessary to confirm (parts of) our performance-oriented hypotheses stated in $\mathbf{H_1}$.

Looking at the feedback provided by participants (see Figure 4.10), we note some interesting differences that provide insights into these quantitative results. For example, as hypothesized by Ghoniem et al. [GFC05] and categorized here under "*Understanding*", we noted several (11) participants explicitly voice their confusion when (double) counting (intra/inter-layer) edges in the layered adjacency matrix representation during task T_{edges}, possibly explaining the observed poor participant performance for this specific task. Additionally, despite the differences in user performance (see Figure 4.7) across tasks, it is interesting that participants were critical of all three node-link diagrammatic representations, specifically their "*Edge Placement*" (✕: −19, ●: −12, ▲: −15), and to a lesser extent "*Node Placement*" (✕: −6, ●: −4, ▲: −7). This was not the case for the adjacency matrix, i.e., (■: −1) and (■: −2), respectively (Figure 4.10). This gives credence to the notion that matrices are possibly preferable for larger and denser graphs owing to the absence of edge and node occlusions [GFC05, OJK19, ABHR+13]. Similarly, the adjacency matrix participants were also less critical of both node "*Localization*" (■: 0) and neighbor "*Counting*" (■: 0) tasks (Figure 4.10) compared to those using node-link diagrams, i.e., (✕: −3, ●: −4, ▲: −5) and (✕: −2, ●: −1, ▲: −3) respectively. This further highlights the previously theorized advantages of adjacency matrices [GFC04, GFC05]. Nonetheless, at least for the chosen graph data, provided training, and user group, these conceptual advantages of adjacency matrices were not beneficial enough to positively impact participant performance for the overview and adjacency tasks. A possible explanation, based on the self-reported user experience (see Figure 4.9), could be the fact that adjacency matrices were statistically significantly harder to learn and use, i.e., S_{Learn} and S_{Use} respectively, compared to their node-link diagrammatic counterparts. This, coupled with the disproportionate number of comments illustrating the poor "*Understanding*" of participants using adjacency matrices (■: −29), compared to the three node-link diagrams (✕: −4, ●: −10, ▲: −6), indicates that participants were not fully equipped to utilize adjacency matrices, despite their having received equivalent training.

Pathfinding and Common Neighbors While the tabular nature of adjacency matrices can make certain tasks conceptually simpler, it can also make others more difficult, such as the investigated common neighbor (T_{common}) and accessibility ($T_{association}$) tasks. In our study, in line with inconclusive related literature [MOB+12, KEC06, ABHR+13], we demonstrate i) the statistical inferiority of adjacency matrices to straight-line and radial node-link diagrams for T_{common}, and ii) the statistical inferiority of adjacency matrices to layered node-link diagrams for $T_{association}$. This, in turn, provides partial evidence for our hypotheses stated in $\mathbf{H_3}$.

Looking at the results of $T_{association}$ more closely (see Figure 4.7), it is notable how many participants, especially those using adjacency matrices, failed to select a 1-alter

node, thereby incurring a maximal error. On the one hand, participants found adjacency matrices statistically significantly more difficult to learn and use. On the other hand, many participants positively and correctly described the representation's "*Edge Weight Encoding*" (■: 11) and its utility, thereby clearly demonstrating an understanding of the task and representation. Perhaps these observations highlight the divide between those participants demonstrating a good and bad "*Understanding*" of the representation, i.e., (■: 15) and (■: −29), respectively. It is also possible that the "*Cognitive Demand*" (■: −21) was high for the participants using adjacency matrices, especially for the complex T_{common} task for which five adjacency matrix users explicitly voiced the high cognitive demand.

User Experience Looking at user experience (see Figure 4.9), participants found adjacency matrices both statistically significantly more difficult to learn (compared to all three node-link diagrammatic representations) and use (compared to layered and straight-line node-link diagrams). Looking at the provided user comments (see Figure 4.10), the most striking difference between adjacency matrices and all three node-link diagrams across all tasks investigated, is the many negative comments left regarding participant "*Understanding*", i.e., (■: −29) and (✕: −4, ●: −10, ▲: −6) respectively. Here, previous comparisons of straight-line node-link diagrams and adjacency matrices have speculated that differences in performance may be, at least partially, attributable to participants' lack of *a priori* familiarity and experience with adjacency matrix representations of graphs [CBDM17, BSC13]. Moreover, past studies have shown that extensive training [NWHL20] as well as gaining familiarity over the course of a longer study [RMO+19] positively impacted participant performance using adjacency matrices. In line with these previous studies, participants using adjacency matrices did respond more critically regarding the provided "*Training*" (■: −5) and "*Instructions*" (■: −11) compared to those using the three node-link diagrams, i.e., (✕: −1, ●: −1, ▲: −1) and (✕: 0, ●: −1, ▲: −3) respectively. This could indicate that longer and more in-depth pre-study training could have assisted participants in better understanding the presented layered adjacency matrix representation. This, in turn, could have assisted them in overcoming the hypothesized *a priori* unfamiliarity with the representation, thereby allowing them also to perform better.

4.7.2 Node-Link Diagrams

Here, given the previously enumerated quantitative and qualitative results, we discuss the differences between the investigated node-link diagrams.

Task-based Performance In line with previous studies conducted comparing different node-link diagram representations and layouts [MS15, HHE06], we observe a statistically significant effect of network representation on user performance generally, namely for T_{common}, $T_{neighbors}$, T_{edges}, and T_{alters}, but ultimately cannot detect any statistically significant differences between individual node-link diagrammatic representations. Addi-

tionally, differences in the time taken per task proved to be fairly small. Specifically, for the ego network representation, only one task had a significant impact on the time taken, namely for T_{alters}, and, again, no significant pairwise differences between the three node-link diagrammatic representations could be found. Beyond statistical evaluations, visual inspection of participant performance (see Figure 4.7) does not reveal any meaningful trends in the differences between these three node-link representations either. Ultimately, it would appear as though we must concur with past evaluations [MS15, HHE06], i.e., differences between different node-link diagrams are present, though very small and are not statistically detectable. This also means we cannot find evidence for our hypotheses of **H2** and **H3**. Similarly, it is interesting how, for most categories (Figure 4.10), all three node-link diagrams proved fairly comparable, with some particular exceptions to be discussed below.

The Value of Structure Looking at participants' comments across the six tasks (Figure 4.10), it is notable how, for both the radial and layered representations, "*Layers*" were mentioned exclusively positively, i.e., (●: 11) and (▲: 9) respectively. This indicates that the partitioning of nodes, in addition to their "*Color*"-coding (✕: 25, ●: 26, ▲: 33), by alter-level was indeed helpful to participants. Relatedly, negative comments regarding the three representations' "*Edge Placement*" were slightly more frequent for the straight-line node-link diagram (✕: −19) than either the radial (●: −12) or layered node-link diagram (▲: −15), potentially further highlighting the utility of a more structured representation. Here, we speculate that the more structured and straightforward representation of the layered node-link diagram could explain the difference in negative "*Cognitive Demand*" comments left (▲: −10), compared to node-link (✕: −24) and radial (●: −23) representations.

User Experience Looking at differences in user experience, it is interesting that the ego network representation only had a statistically significant impact on participants' self-reported ease of use, S_{use}, and ease of learning, S_{learn}. However, for those two statements, there were no statistically significant pairwise differences between node-link diagrams, which, in turn, means that we are unable to provide evidence for our hypothesis **H4**. Looking at the qualitative feedback provided by participants, we note some interesting visual observations. As discussed previously, while participants were positive regarding the use of "*Color*" across node-link representations, it is notable how frequently participants commented positively about radial and layered node-link diagrams' "*Layers*". Specifically, these comments often described how helpful these layers were in distinguishing alter levels or inter- and intra-alter edges from each other. This could explain the minor visual differences between layered and the other two node-link diagrammatic representations across statements (see Figure 4.9). For S_{learn}, this could indicate that layers assisted participants in understanding the fundamentally layered structure of ego-networks. This, to a lesser extent, could also explain the visual differences observed in $S_{quickly}$ and $S_{accuracy}$. At least for the data presented, these hypothesized benefits did not translate to statistically significant differences in user experience.

4.7.3 Interaction

While not directly related to the representations investigated, many participants left positive comments on the "*Interactive Highlighting*" that each visual representation featured, i.e., 124 in total. These comments were especially numerous across the three node-link diagrammatic representations (✕: 39, ●: 30, ▲: 39) compared to the adjacency matrix (■: 16), pointing out that certain tasks would have been much more difficult or even impossible without such interaction.

We note several negative comments regarding user interaction, often regarding the specific graph tasks, i.e., "*Localization*" (total = −12), "*Counting*" (total = −6), and "*Comparison*" (total = −33). Again, such comments were more numerous for the three node-link diagrammatic representations, with no single negative utterance across these categories from users of the adjacency matrix. Participants often voiced their desire for some form of automation to make such tasks less complex or arduous. They often wished for certain features such as the ability to filter nodes and edges, a node-lookup feature, the automatic counting of a node's neighbors, or the ability to select and compare multiple nodes at once. Coupled with some of the previously discussed difficulties participants had answering certain questions and the subsequently increased "*Cognitive Demand*" required, these results further point towards the need for meaningful interaction and automated analysis tools in (seemingly especially in node-link diagrammatic) ego network visualizations designed for application and production [WEF+14, YKSJ07, Fig15].

4.7.4 Validating Our Hypotheses

Based on previous work, particularly non-ego-network comparisons of adjacency matrices and various node-link diagrams (Section 4.3.1), we formulated a series of hypotheses regarding the performance and preferences of users (Section 4.5.3). However, we were unable to find evidence for all posited hypotheses. Most notably, we expected adjacency matrices to perform *better* than the three node-link diagrammatic representations for all tasks (**H1**) but $T_{association}$ and T_{common} (**H3**). We found our layered adjacency matrices to perform (statistically significantly) worse across all six tasks (Section 4.7). We posit that, in addition to the usual difficulties of learning and understanding adjacency matrices (Section 4.7.1), the complexity introduced by an ego network's alter layers exacerbates these issues. While the layered structure of the implemented layered ego network adjacency matrix *should* have made certain tasks easier, it was even more complex to understand than conventional, non-ego-network adjacency matrices. Given the limited work on centered/layered adjacency matrices, it is difficult to relate these findings to previous work. However, in Bae et al.'s [BW11] comparison of centered adjacency matrices, quilts, and node-link diagrams, adjacency matrices consistently required more time from participants to complete the low-level graph tasks presented. As also speculated by Bae et al. [BW11], this could indicate that certain tasks are indeed harder to complete with a layered adjacency matrix than other embedding types. The layered structure of ego networks appears to have been understood much better in both radial and layered node-link diagrams. This indicates that more research in this direction

is necessary to fully understand how an ego network's layered structure can be most effectively communicated in adjacency matrices, specifically, and what lessons can be learned from node-link diagrams.

4.8 Limitations and Future Work

Using Additional Visual Channels In this study, in the interest of comparability across the three node-link diagrammatic representations and (layered) adjacency matrix, we have focused on one particular set of visual mappings implemented in *D3.js* (Section 4.4). Given its preattentive properties, we selected color as the visual variable to encode alter levels, i.e., embedded node attributes [EMWR24], and (where possible) node position, as well as edge weight using line segment opacity. However, several visual channels were not utilized at all, such as node size, node shape, or edge thickness, which could have additionally been utilized to further communicate topological properties. For example, the alter/ego degree could have been communicated using node size. Alternatively, the alter level could have been redundantly encoded in the nodes' shapes. Here, in both the interest of simplicity and comparability to adjacency matrices, such additional mappings were not considered. As our design choices would have impacted a user's interpretation of the data and identification of patterns, we opted to keep these visual attributes constant across all representations.

Future work, as well as application-driven implementations of the network representations featured here, should make appropriate use of the visual channels to communicate as effectively as possible the necessary topological information. For example, redundantly encoding the alter level with node color and shape could allow for greater accessibility for color-blind users. Additionally, layout approaches outside of *D3.js*' force-based layout should be considered, such as Sugiyama-based algorithms [STT81] for layered node-link diagrams, or frameworks akin to *Circos* [KSB+09] for radial node-link diagrams. We opted to utilize *D3.js* for all representations to ensure aesthetic and run-time comparability across all representations. Future research endeavors and application-driven implementations should consider and investigate further these alternative layout approaches.

More Complex Data Given the hypothesized and shown poor scalability of node-link diagrams (compared to adjacency matrices) [KPS14, EVRW23, GFC04, ASA+23, ABHR+13] and the many negative comments left regarding node-link diagrams but not adjacency matrices, it is interesting that i) participants using adjacency matrices performed systematically worse and that ii) no statistical differences between node-link diagrammatic representations could be detected. We speculate that the selected network size and density may have played a role in this. Increased numbers of nodes and edges could render straight-line node-link diagrams increasingly unreadable, complicating counting, adjacency, and look-up tasks. On the other hand, the orderly arrangements of nodes in radial and layered node-link diagrams, as well as adjacency matrices, could

mitigate these effects. We, therefore, recommend that future works investigate the differences in participant performance and experience, taking also into account graphs of greater sizes and complexities with different selected egos in an even larger follow-up user study.

Better Training Participants appear to have had less of an "*Understanding*" of the representation and tasks when using adjacent matrices (Figure 4.10); an observation underscored by their systematically poor performance. It is worth asking to what extent this observation is owed to the representation itself, the training and instructions participants received, or, as hypothesized in previously conducted studies [CBDM17, BSC13, ABZD13, GFC05], participants' *a priori* lack of familiarity with adjacency matrices. Given previous studies highlighting the benefits of more extensive training [RMO+19, NWHL20] as well as the participants' negative comments on using the adjacency matrices related to "*Training*" and "*Instructions*", we wonder whether our results would have painted a more favorable picture, had we employed longer and more in-depth training. We argue that, despite participants' poor performance using adjacency matrices, the previously discussed conceptual advantages of adjacency matrices over node-link diagrams still hold and are a direction worth pursuing. Future research efforts should continue to investigate this representation as a potentially powerful alternative to node-link diagrams. To make such adjacency representations more useful, it would be interesting to thoroughly investigate the measurable effect of different types of training on participant performance when using adjacency matrices. Such results would be invaluable in guiding researchers and engineers alike in selecting their ego network representation and choosing appropriate methods of training.

More Complex Graph Types In this chapter, we focus exclusively on an undirected, weighted ego network. While the majority of the findings presented here should hold for unweighted graphs as well, not all do. Most notably, the tie-breaking mechanism of the layered adjacency matrix requires weights to determine the order of the ego network's nodes within each k-alter block (Section 4.4.4). Future work should certainly make an effort to study the impact these visual representations could have on different types of networks, which feature directed or unweighted edges, multiple node or edge types, (hierarchical) group structures, or multivariate attributes that need to be displayed.

Furhter Testing While great care was taken in selecting a representative dataset (Section 4.5.1), it must naturally be acknowledged that a single dataset cannot fully capture the complexities of all possible ego networks. For example, one could have selected multiple different egos to investigate within the investigated "*Les Misérables*" dataset, resulting in different maximum values of k, different topologies, and subsequently different network visualizations. Alternatively, multiple different datasets of different node sizes $|N|$ and edge numbers $|E|$ could be investigated, each with its own set of ego, k, and visualizations. However, within the context of the labor-intensive mixed methods analysis conducted here, such larger-scale studies were beyond the capabilities of this

first systematic look at different layout approaches to ego networks. Future work should investigate such scenarios (perhaps purely quantitatively) to paint a fuller picture of when these different visualizations are to be put to best use.

4.9 Conclusion

In this chapter, we compile a list of 50 ego network visualizations in order to identify the most common approaches, i.e., straight-line, radial, and layered node-link diagrams, as well as adjacency matrices. We then study these approaches' quantitative and qualitative impact across six different ego network-specific graph analysis tasks on user performance and user experience in a large-scale, crowd-sourced user study of 120 participants on a single, intermediate-sized, representative graph dataset. Our results indicate that:

1. Participants using *adjacency matrices performed systematically worse* than those using node-link representations, despite the former's conceptual advantages highlighted both in the literature and in the participants' comments.
2. In line with previous studies, all three *node-link diagrams performed very similarly to each other*, despite the many positive comments highlighting the conceptual benefits of alter layers on both user performance and learning.
3. The representation had *hardly any impact on the time* participants needed to complete the six tasks investigated.
4. There is a *need for better training and appropriate instructions* in order to overcome participants' lack of familiarity with adjacency matrix representations in particular.

Ultimately, unless an ego network's user group is already familiar with or receives extensive training on a particular representation, we recommend the use of a layered node-link diagram, as these proved statistically comparable to other node-link diagrams and superior to adjacency matrices in terms of both user performance and experience. Notable differences in user experience and comments left by our study participants hint at the value of the layered node-link diagram's more structured and straightforward representation of the ego, its alters, and their inter- and intra-alter relationships, compared to radial and straight-line node-link diagrams.

CHAPTER 5

Are You Sure? Node-Attribute Uncertainty in Network Visualization

Biological graphs are rarely purely topological in nature but can feature a variety of multivariate attributes attached to nodes, edges, or both. These attributes can represent different aspects of the data, such as gene expression levels in gene-gene interaction networks or protein abundances in protein-protein interaction networks. Visualizing such multivariate networks is challenging as interest simultaneously lies in both the network's topology as well as its attributes. Complicating matters even further, these attributes may be uncertain, e.g., measurement errors, analysis errors, or irreducible uncertainty inherent in the process. Here, it may be valuable to visualize these uncertainties alongside the network's multivariate attributes in order to enhance user decision-making, e.g., selecting a particular biological entity for experimental follow-up study. However, many different visual channels for the visualization of these uncertainties present themselves, each with conceptual advantages and disadvantages. One particular approach remains relatively underexplored and controversial, despite its conceptual usefulness: animation.

Here, we investigate animation (in the form of node "wiggliness") as a means of encoding node attribute uncertainty, and compare this novel approach to a set of carefully selected alternatives: node enclosure, node saturation, and node sharpness/fuzziness. The contents of this chapter are based on the full paper *"Wiggle! Wiggle! Wiggle! Visualizing Uncertainty in Node Attributes in Straight-Line Node-Link Diagrams Using Animated Wiggliness"* [EPdB+25a], published in the journal *Computer & Graphics*, with myself as first author, in collaboration with Daniel Pahr, Sara di Bartolomeo, Velitchko Filipov, Hsiang-Yun Wu, and Renata Raidou.

5.1 Overview

Uncertainty is common to most types of data, from meteorology to the biomedical sciences. Here, we are particularly interested in the visualization of uncertainty within the context of multivariate graphs, specifically the visualization of uncertainty attached to node attributes. Many visual channels offer themselves up for the visualization of node attributes and their uncertainty. One channel, however, that has remained under-explored, and has actually been criticized in the past, is animation, despite its conceptual advantages. In this chapter, we thus investigate node "wiggliness", i.e., uncertainty-dependent pseudo-random motion of nodes, as a potential new visual channel with which to communicate node attribute uncertainty. In order to study wiggliness' effectiveness, we compare it with three other visual channels identified from a thorough review of uncertainty visualization literature — namely node enclosure, node fuzziness, and node color saturation. We compare these four uncertainty encodings in a larger-scale, mixed-method, *Prolific*-crowd-sourced, online user study of 160 participants. In this study, we quantitatively and qualitatively compare these four uncertainty encodings across eight low-level graph analysis tasks that probe participants' abilities to parse the presented networks both on an attribute and topological level. We ultimately conclude that all four uncertainty encodings appear comparably useful—as opposed to previous findings. Wiggliness may be a suitable and effective visual channel with which to communicate node attribute uncertainty, at least for the kinds of data and tasks considered in our study.

5.2 Introduction

Uncertainty is common to most types of data and can affect the visualization pipeline at any stage, from data acquisition, through data transformation, to rendering [PWL97]. Effective visual communication of uncertainty in data is essential for a user's accurate interpretation and informed decision-making [PKH20, CCP07]. While common in some fields, such as meteorology [CHL13], climatology [HVCM17], and biomedical sciences [HMT+17], the visualization of uncertainty remains underutilized across many other fields [Hul20], such as network visualization.

In this work, we thus focus on the field of network visualization, in which uncertainty is often discussed, but rarely visualized [CGH+24]. Uncertainty in network visualization can take many different forms, such as geometrical (embedding) uncertainty [YC17], topological (edge and node) uncertainty [SNG+17], edge attribute [SSSE16, SS18] uncertainty, or node attribute uncertainty [VHK+12]. We are particularly interested in the visualization of node attribute uncertainty, as it remains understudied despite node attributes being (the most) commonly included in many multivariate networks [NMSL19].

For the visualization of node attributes and their uncertainty, several visual channels present themselves, as outlined by Conroy et al. [CGH+24] and summarized in Figure 5.1. The actual effectiveness of these visual channels' abilities to draw user attention to

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 5.1: Various visual channels with which to communicate node attributes, and uncertainty more specifically, according to Conroy et al. [CGH+24]. Nine possible channels are identified, namely a) node size, b) node color hue, c) node shape, d) saturation, e) node sharpness or fuzziness, f) node texture, g) node enclosure, h) node gradient, and i) node splatting.

areas of uncertainty remains unexplored [CGH+24]. In this work, we highlight an often-overlooked uncertainty visualization method—animation—which remains uncommon even beyond network visualization [CGH+24]. We additionally compare it against other visual channels in terms of effectiveness.

We argue that animation holds significant potential for uncertainty visualization. While not universally applicable, common guidelines suggest emphasizing regions of high uncertainty [BHJ+14, CGH+24]. Movement effectively draws attention without increasing cognitive load [BWC01], and animation may help explain complex concepts like uncertainty while engaging users [TLLS20]. It also "frees up" other visual channels, which may already be in use for representing different data dimensions. While animation has been criticized in dynamic network contexts for being too complex for accurate interpretation [TMB02], such concerns may be less relevant for uncertainty visualization, where conveying relative rather than exact values is often sufficient [CCP07].

We hypothesize that animation, while controversial, could be highly effective at visualizing uncertainty in node attributes and warrants further investigation. In contexts where uncertainty is critical, animation could be a valuable alternative to existing visual channels, as it could draw strong attention to regions of high uncertainty. However, whether animation can form an effective alternative to commonly employed visual encodings, both in terms of user preference and performance, remains largely unexplored. Similarly, the relative utility of standard uncertainty encodings in network visualization is still not well understood [CGH+24]. This work investigates the impact of different visual channels for uncertainty visualization on both user experience and performance.

In this chapter, inspired by previous biomedical work [LLPY07, Bro04, BTM+19], we thus present a novel approach to visualizing uncertainty in networks' node attributes using "*wiggliness*" (Ⓦ), i.e., animated pseudo-random motion. To compare this novel network uncertainty visualization approach to meaningful alternatives, we perform a thorough survey of uncertainty visualization literature. This corpus of literature is then categorized by application domain and uncertainty visualization approach to better understand the current landscape of the field (outside of the context of network visualization). From the thus identified approaches, we select three representative and effective uncertainty visualization techniques that can be applied to network visualization

[CGH+24]: saturation (S), enclosure (E), and fuzziness (F). To determine whether wiggliness holds value as an uncertainty visualization technique, we then conduct a large-scale, online, crowd-sourced user study in which we probe user performance and preference both quantitatively and qualitatively in a mixed-method setup.

In summary, we introduce animated node "wiggliness", a novel technique with which to represent attribute uncertainty in node-link visualizations, and compare it to enclosure, fuzziness, and saturation. A user study of 160 participants shows that all four methods offer comparable performance and perceived learnability, intuitiveness, and understandability.

5.3 Related Work

5.3.1 Visualizing Uncertainty in Networks

While uncertainty visualization has received notable attention in fields such as biomedical or meteorological visualization, the same cannot be said for network visualization [CGH+24]. Here, we outline what has been done in regard to uncertainty visualization in networks, specifically uncertainty in graph layouts, uncertainty in edges, and uncertainty in nodes.

Uncertain Layouts Unless nodes have some intrinsic 2D/3D positional information, the network's embedding is merely a function of the selected automatic layout algorithm. There is, thus, ambiguity in these layouts, depending on which layout algorithm is chosen or what hyperparameters are chosen for a specific layout algorithm. To address this ambiguity, Yan and Cui [YC17] opted to visualize the resulting embeddings side-by-side as an ensemble in order to allow users to investigate differences in the produced clustering of nodes. Similarly, Wang et al. [WSA+16] also visualize multiple such produced layouts, but instead of visualizing them side-by-side, they visualized them overlayed atop each other.

Uncertain Edges Beyond the entire layout being uncertain, a graph's edges may have uncertainty attached to them and their weights. For an application-driven example hereof, Vehlow et al. [VHK+12], within the context of biochemical reaction network analysis, visualized the uncertainty in an edge using color saturation, i.e., the more certain an edge, the more saturated it is. Within the context of flow diagram visualization, Vosough et al. [VKKG17, VKKG19] investigated multiple different encodings of edge uncertainty, i.e., color saturation, edge gradient (fuzziness), and color hue. With the results of their conducted user study in hand, the authors ultimately conclude that color saturation works best for their purposes. More abstractly, some have also investigated different approaches to edge uncertainty visualization outside of any particular application domain. Here, Schwank et al. [SS18, SSSE16] investigated four possible edge encodings, namely edge dashes, waves, stripes, and blurring. They ultimately concluded that dashed edges communicated uncertainty most effectively. Finally, Guo et al. [GHL15]

investigated multiple pairs of visual encodings to communicate edge uncertainty, i.e., lightness+saturation, fuzziness+saturation, lightness+width, and fuzziness+width.

Uncertain Nodes and Node Attributes Finally, and least common, a network's nodes and their attributes may also be uncertain. Cesario et al.'s [CPS11] opted to encode the (un)certainty in a network's node attributes using the nodes' positions in 2D space. Returning to Vehlow et al.'s [VHK+12] biochemical reaction visualization, uncertainty in nodes' attribute uncertainty is visualized using color saturation; the more saturated, the more certain. Lastly, within the context of lattices, Collins et al. [CCP07] visualized node uncertainty using transparency, border fuzziness, and position. Animation has, to the best of our knowledge, never been used to visually communicate node or edge attribute uncertainty.

5.3.2 Animation

Within the context of medical visualization, animation has been used for a variety of reasons, such as viewpoint selection, camera path planning, and focus+context visualizations [PM20b]. Here, we aim to highlight the work done using animation for the purposes of uncertainty and network visualization.

Uncertainty Visualization Using Animation Examples of animation to visually communicate uncertainty (relevant to our own proposed approach) include i) Blenkinsop et al.'s [BFBW00] random animations to communicate the results of fuzzy satellite classification data, ii) Lundstrom et al.'s [LLPY07] probabilistic animation for medical volume visualization , iii) Brown et al.'s [Bro04] animated oscillation between two states to indicate uncertainty between them, iv) Akiba et al.'s [AWM10] animated transfer functions in medical volume visualization, iv) Kale et al.'s [KNKH19] animated Hypothetical Outcome Plots, Ma et al.'s [MCC+20] dynamic visualization of uncertainty in medical features of interest, or Hermosilla et al.'s [HBVV12] visualization of uncertainty visualization of brain fibers.

Animation and Network Visualization While an example of animation for uncertainty visualization in network visualization can be found [ZQL+18], the majority of applications of animation center around dynamic, i.e., time-dependent, network visualization [MMB05]. Very simply, such approaches map the time dimension of a network to time, i.e., animation frames [BBDW14]. Many examples of animated dynamic network visualization approaches exist, all of which aim to preserve a user's mental map between time steps while still highlighting changes between steps meaningfully, as seen in *GraphAEL* [FKN+05, EHK+04] or *Visione* [BS08]. As a complete discussion of such animated dynamic graph visualization approaches is beyond the scope of this chapter, we refer the interested reader to Beck et al.'s review of the topic [BBDW14]. However, while animation has found frequent use, empirical comparisons of animated with static/non-animated visualizations do not always paint a favorable picture of animation.

For example, Robertson et al. [RFF+08] and Archambault et al. [APP11] both found (for certain tasks) small multiples to outperform animations in their evaluations. Similarly, Farrugia et al. [FQ11] found static visualizations to generally outperform animation. However, these evaluations are seldom clear-cut. For example, Saraiya et al. [SLN05] found animation to outperform interactive timeline visualizations, at least when the number of time points was few. Boyandin et al. [BBL12] showed that animation led to more findings on adjacent time steps than small multiples. Finally, animation has also been shown to be useful for highly specific tasks, such as highlighting short transitions, detecting small changes in large data, and highlighting nodes [WB05]. In general, the question of whether animation is beneficial to users in dynamic (and subsequently uncertain) network visualizations is, in our estimation, still far from settled.

5.4 Study

In this study, we aim to quantitatively and qualitatively evaluate the effect of node attributes' uncertainty representation on user performance, user experience, and the overall "value" of the visualization. In order to ensure sufficient statistical power and qualitative feedback, we conduct a large-scale, between-subjects, online, crowd-sourced study. A between-subjects study design was chosen to ensure the study would take no longer than 30 minutes. Based on our prior experience with such online studies, this is the upper limit for online crowd-sourced studies, as longer studies negatively impact participant concentration, user performance, and the quality of qualitative comments received. Note that participants were required to complete the survey on a desktop or laptop computer with a display resolution of at least *1080p*.

First, we quantitatively evaluate the impact of four selected node attribute uncertainty representations on user performance. Specifically, we study the accuracy with which participants are able to answer a series of low-level graph analysis tasks and the time necessary to do so. Second, we also quantitatively evaluate the impact of these representations on user experience. Each participant is required to answer five questions regarding their experience on a 7-Point Likert scale at the end of the study, probing each participant's i) perceived accuracy and efficiency, ii) the ease with which participants used and learned the uncertainty representation, and iii) the aesthetic appeal of the visualization. Finally, to go beyond a purely quantitative evaluation, we enrich said analysis with an additional qualitative analysis. To do so, after each completed task, participants were presented with optional feedback regarding the uncertainty visualization, as well as required feedback summarily at the end of the study.

In this section, we outline and motivate i) the selection of uncertainty representations, ii) the study's research questions, iii) the data utilized for the study, iv) the selection of low-level graph analysis tasks we investigate, and v) the mixed-methods analysis of the produced data. The implementation, as well as our classification of the collected papers, has been made available on the Open Science Framework[1].

[1] https://osf.io/e8927/?view_only=6295953d19ef439a8e9c11d5469f9310

5.4.1 Uncertainty Visualizations

In this section, we describe how we selected and implemented our four uncertainty visualization channels: saturation (S), fuzziness (F), enclosure (E), and wiggliness (W).

Design Rationale

Limited work has been done on uncertainty visualization in networks (Section 5.3.1). Thus, to select popular, representative, and effective visual channels for comparison, we characterize the landscape of uncertainty visualization outside of the context of networks. To use as representative a sample of papers as possible, we collected all references from Jena et al. [JED+20], who compiled a total of 286 papers from two previously published surveys [Pot13, HQC+19] and through their own systematic search of the literature. We excluded papers that were not focused on visualization specifically (e.g., theoretical discussions of uncertainty), or not relevant to our survey of literature (e.g., review papers), resulting in a total of 191 papers. These papers were manually categorized by their application domain as well as their chosen visual encoding of uncertainty. We drew upon Jena et al.'s [JED+20] already existing categorization of application domains, and supplemented our own categories where needed. For our categorization of chosen visual encodings, we drew upon work by Weiskopf [Wei22], as it encapsulates many previously published taxonomies [PWL97, GS06, PRJ12, MRO+12, BHJ+14, PKH20, KDJ+21]. Here, Weiskopf categorizes visual approaches into three broad categories: **Hybrids and Systems**, **Display of Distribution**, **Summary Statistics**. Because Weiskopf's taxonomy is non-exhaustive, we, where necessary, added visual encodings from previous work, e.g., Padilla et al. [PKH20], yielding a total of 31 unique approaches to uncertainty visualization. Figure 5.2 shows the results of the categorization, as well as the total number of papers that featured each visual encoding. The categorization of these papers 5.2 has been made available through the Open Science Framework.

Some of these encodings have been argued to map more naturally and intuitively to uncertainty than others [PKH20], such as fuzziness/fogginess [RLBS03, MRH+05], out-of-focus blur [JOK95], transparency [GR04], location [MRO+12], color saturation [Hen03], sketchiness [BBIF12], and noise textures [HM96]. Some of these visualization approaches (e.g., positional blur [MHBG16] or value-suppressing color-palettes [CMH18]) literally obfuscate the value in question: the more uncertain it is, the harder to read it becomes. In our selection of visual encodings, we aim to use those that map the most naturally to uncertainty.

Here, not all of the identified 31 approaches to uncertainty visualization (Figure 5.2) are (straightforwardly) applicable to the visualization of node attribute uncertainty. We, therefore, applied our findings to the nine methods identified by Conroy et al. [CGH+24] (Figure 5.1) and settled on three intuitive and popular visualization approaches to compare to animated wiggliness (W): i) saturation (S), ii) fuzziness (F), and iii) enclosure (E). In summary, these three uncertainty encodings were selected as they are i) applicable to node attribute uncertainty visualization [CGH+24], ii) commonly utilized outside of

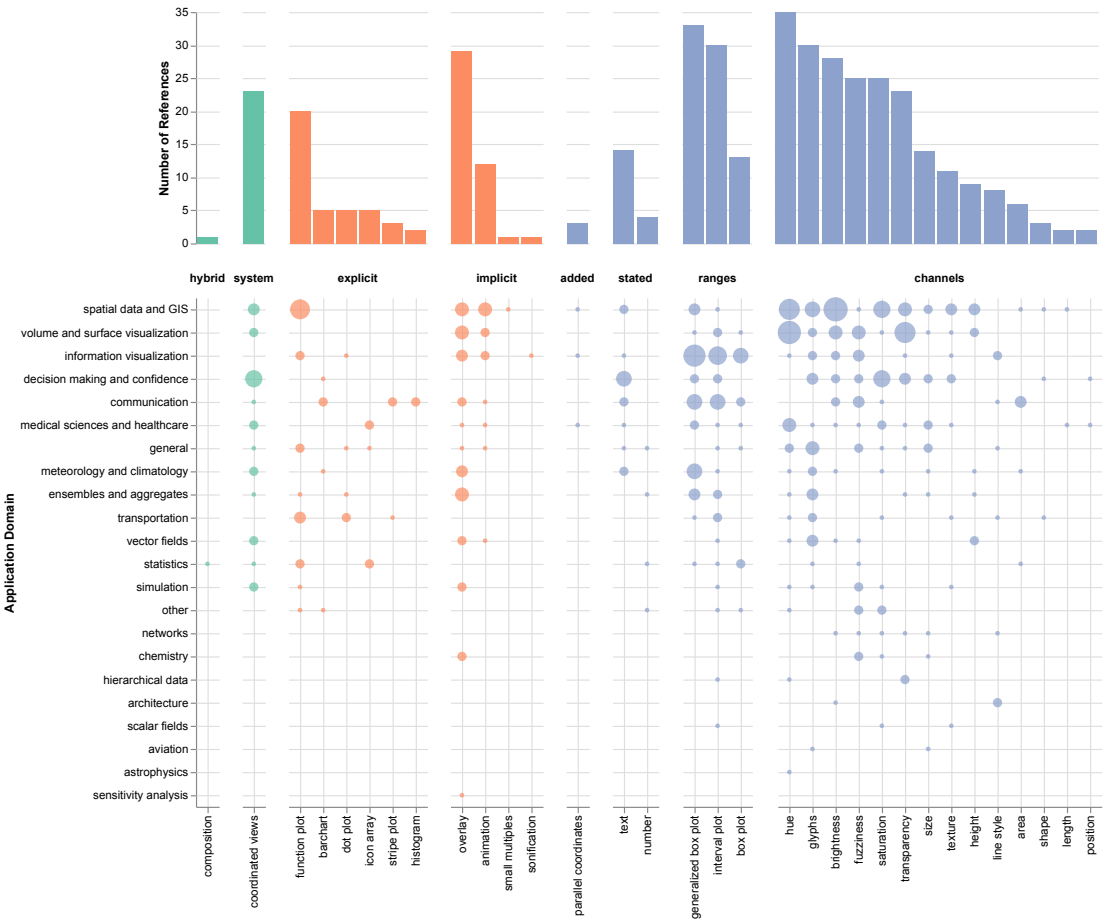


Figure 5.2: Categorization of collected uncertainty visualization papers, visualized as a dot plot. The number of papers that mapped to a particular combination of application domain and visual encoding is visualized as a dot, whose surface area encodes the number of papers. Our classification of application domains was, where available, based on the work of Jena et al. [JED+20]. Our classification of visual encodings is based on the taxonomy of Weiskopf [Wei22]: All possible visual encodings fall within one of three categories, namely **Hybrids and Systems**, **Display of Distribution**, and **Summary Statistics**. Each of these categories is further broken down into subcategories, which in turn encapsulate individual visual encodings. Rows, i.e., application domains, are sorted in descending order of the number of papers that mapped to said application domain. Within each subcategory facet, columns, i.e., visual encodings, are also sorted in descending order of the number of papers that mapped to said visual encoding. The total number of papers that mapped to a particular visual encoding across application domains is visualized in the corresponding bar in the bar chart atop the dot plot. Papers could map to multiple application domains and visual encodings.

the context of network visualization (Figure 5.2), and iii) deemed intuitive or effective in the context of uncertainty visualization [PKH20]. Their selection ensures a fair and meaningful comparison, focusing on the most compelling alternatives currently available.

Selected Approaches

In this section, we discuss the four selected approaches to uncertainty visualization that we use in this study. Additional (dis)advantages as discussed in the literature are further elaborated in Section 5.6.1.

Saturation (S) With 24 occurrences across 14 different application domains (Figure 5.2), such as medical volume visualization [Kni08] and spatial data analysis [LCBD11], saturation is one of the most commonly utilized Summary Statistics approaches to uncertainty visualization. Relatedly, brightness, sometimes called "*lightness*", is also frequently employed, with 28 occurrences across 10 application domains. Saturation is implemented through either a continuous [ZA11, GHL15, HMT+17] or ordinal [PHR+15, GBFM16, VKKG17] color scale, respectively indicating continuous or discretized levels of (un)certainty. Compared empirically to encodings within and outside of the context of uncertainty visualization, saturation i) shows potential as an effective encoding [GHL15, PHR+15, Mun14, GF09], ii) is liked by users [SL14], and iii) deemed easily understandable [BBIF12]. We opt for a continuous saturation color scale, which denotes entities of low uncertainty with low saturation, and entities of high uncertainty with high saturation (Figure 5.3a) [SZB+09, BVPtHR09].

Fuzziness (F) With 25 papers across 13 different application domains (Figure 5.2), such as volume rendering [FM12], flow diagrams [VKKG17] and lattice graphs [CCP07], fuzziness is also a widespread Summary Statistics approach to uncertainty visualization. In empirical and qualitative comparisons of fuzziness against other visual encodings of uncertainty, fuzziness is generally recommended [PHR+15, TTvE14, GBFM16], described as intuitive and easily associated with uncertainty [BBIF12], and explicitly shown to be superior to other encodings in certain settings [CG14, GES+22]. As described by Bonneau et al. [BHJ+14], increased fuzziness intuitively communicates lower confidence, i.e., higher uncertainty. Here, we thus map increasing levels of uncertainty to increasing levels of fuzziness (Figure 5.3b).

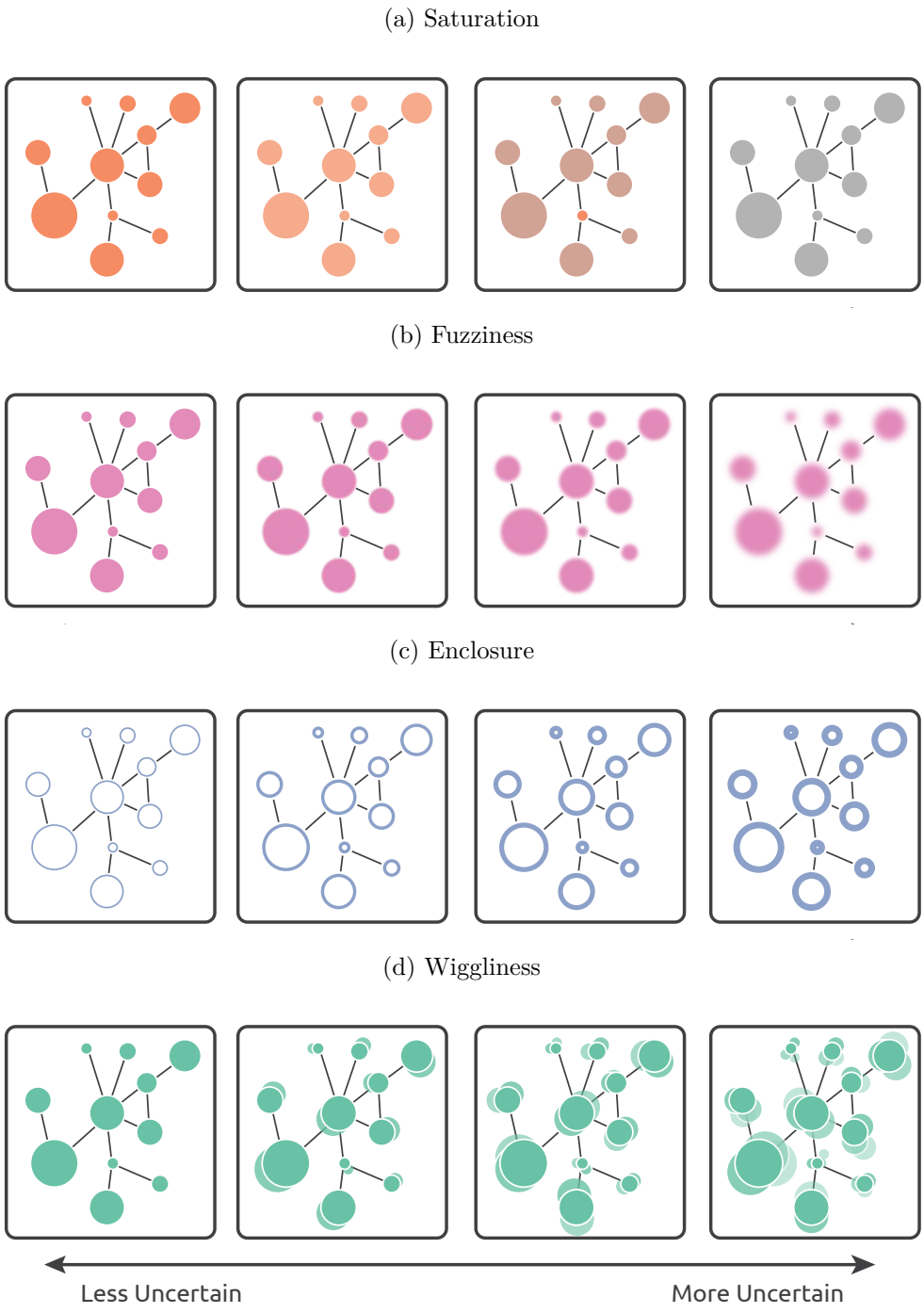


Figure 5.3: Illustrative examples of a) saturation, b) fuzziness, c) enclosure, and d) wiggliness encoding uncertainty in a synthetic network of size $|V| = 9$ and $|E| = 11$. Note that, for illustration purposes, all nodes share the same degree of uncertainty for their attributes.

Enclosure (E) Node enclosure describes the style of the line border enclosing the surface of the (circular) node (Figure 5.1g). Here, we define node enclosure as the *thickness* of this enclosing border. More specifically, akin to various types of *generalized box and interval plots* (Figure 5.2), we map uncertainty to this thickness: the more uncertain a node's attribute, the thicker its border, in the same way, that an interval or boxplot becomes wider/larger for less certain data. Such (generalized) interval and box plots (**Summary Statistics**) also form a very popular approach to uncertainty visualization outside of the context of networks (Figure 5.3c), with 76 papers across 16 application domains mapping to the category as whole, and, more specifically, box plots, interval plots, and generalized box plots featured in 13, 33, and 30 papers, respectively. These types of range plots have found application/study across domains [BKC+12], such as information visualization [RRT99], (non-expert) communications [GOHS15], and meteorology and climatology [SZD+10].

Wiggliness (W) Finally, animation, with only 12 occurrences across 7 application domains (Figure 5.2), remains one of the least popular **Display of Distribution** visual encodings for uncertainty visualization. As discussed previously (Section 5.3.2), animation has found some use, particularly in the visualization of time-dependent phenomena in medical [PM20b] and spatial data [ESG97] visualization. Even in the context of uncertainty visualization, animation has found some, if limited, use, both within the context of network visualization [ZAH22] and outside of it [KNKH19]. Here, inspired specifically by promising previous works on random [BFBW00], procedural [LLPY07], and looping [Bro04] animation, we propose *wiggliness* as a new channel for uncertainty visualization in networks. Intuitively, wiggliness maps uncertainty to animated motion: the less certain the attribute of a node, the more that node moves randomly in (2D) space, drawing user attention to it (Figure 5.3d). Wiggliness visually conveys uncertainty by adding dynamic motion to uncertain nodes, conceptually making it easier for users to detect regions of high uncertainty at a glance. This approach leverages the human ability to notice movement, thereby minimizing cognitive effort while maximizing awareness of uncertainty. More specifically, each node $v \in V$ is initially located at some layout-algorithm-derived mid-point $p_v = (x_v, y_v)$. Per frame, this node is allowed to move randomly around said mid-point to some new position $p'_v = (x'_v, y'_v)$, as a function of its uncertainty-defined radius r; the greater the uncertainty, the greater the radius. This then requires the redrawing of both the node v as well as all edges connected to it, i.e., $\{\{v, w\} : \forall w \in V, w \neq v\} \cap E$. Prior testing showed that sampling x'_v and y'_v from uniform random distributions (independent of previous node locations) resulted in the clearest form of wiggliness. In contrast, position-dependent sampling (e.g., Gaussian noise or random walks) introduced apparent spatial patterns, which misleadingly suggested structure in the uncertainty. We also found that animating at 20 frames per second (fps) offered a smooth visual experience and consistent performance across systems. The combination of a uniform random node movement at 20 fps most effectively conveyed random, structureless uncertainty through visual jitter.

Implementation

All four uncertainty representations and network visualizations were implemented in *D3.js* [BOH11]. More specifically, all graphs were laid out using *D3.js*' particle-based force-directed simulation algorithm, as it produced consistently visually appealing results very quickly. All four representations were implemented monochromatically. All four representations did not feature any interactivity, in order to ensure we were only studying the effects of the uncertainty visualization, instead of inadvertently their interactions with certain modes of user interactivity [LLPY07]. Moreover, interaction has been noted to increase the cognitive demand on users, especially non-expert ones [SR05, SBL93]. Again, as we wish to avoid such cognitive load differences, or, more precisely, avoid their effect on our results, we implement static visualization in this study. Examples of the produced network visualizations, as they were presented to user study participants, can be found in Figure 3 of the supplement.

5.4.2 Research Questions

Here, based on both previous research (see Section 5.3) as well as our own expectations, we formulate two research questions to be answered both quantitatively and qualitatively.

Q_1: Differences in Performance *Which low-level graph analytical tasks are best supported by which uncertainty visualization?* As pointed out by Conroy et al. [CGH+24], little work has been done to compare different uncertainty visualization strategies in networks. It is hence difficult to *a priori* hypothesize how the four selected uncertainty visualization strategies will fare in terms of user performance. Moreover, studies comparing animated solutions to uncertainty visualization to static approaches are also rare (Section 5.3.2). We thus ask ourselves which uncertainty visualization approach will prove most effective for certain low-level graph analysis tasks.

Q_2: Differences in Preferences *Which representation is preferred by users?* It has been argued that certain visual channels, such as fuzziness, transparency, or location, map more intuitively to uncertainty than others [PKH20, BHJ+14]. Here, while all four uncertainty visualization approaches were selected for their intuitiveness, we ask ourselves which representation will be preferred by users.

5.4.3 Data

In the following section, we briefly describe the graph data used in our study. Specifically, we discuss the process of generating graph topologies, node attributes, and node attribute uncertainties.

Topology Similar to past comparative graph studies [EPF+24], we perform our user study using real graph topologies instead of simulated ones to avoid the non-representativeness often associated with simulated graphs [BC19]. As we aim to investi-

gate eight topology and attribute-based tasks (see Section 5.4.5), a selected graph must have multiple realizations, i.e., one per task, to curb any learning effect across tasks. Additionally, to investigate the effect of the number of vertices on user performance and experience, we aim to investigate two sets of graphs: one small and one medium in size [YAD+18]. We select two sets of animal social networks of (roughly) two different sizes and densities, both collected from the *Network Repository* [RN15] that fulfill the selection criteria stated above.

First, we investigate a set of **raccoon proximity networks**. Each node represents a particular raccoon, and each undirected, weighted edge represents the number of times two raccoons were close to each other over the course of a day [MCK13]. This dataset consists originally of 52 networks, each representing one day of a 52-day experiment. From these 52 graphs, eight were selected for their overall similarity (in terms of their number of edges and nodes), i.e., one per graph task. The number of vertices of the thus selected eight graphs ranges from $21 \leq |V| \leq 23$ and the number of edges from $50 \leq |E| \leq 82$, resulting in densities [KKPEP20] ranging from $0.198 \leq d \leq 0.355$ (Figure 5.4a).

Second, we investigate a set of **ant interaction networks**. Each node represents an ant of a particular colony, and each undirected, weighted edge is the number of mandible-to-mandible contacts between two ants over the course of a day [Que15]. This dataset consists of 8 graphs, each representing one day of an eight-day experiment. The number of vertices ranges from $24 \leq |V| \leq 31$ and the number of edges from $30 \leq |E| \leq 52$, resulting in densities [KKPEP20] ranging from $0.073 \leq d \leq 0.137$ (Figure 5.4a).

Node Attributes In our study setup, each node is to have one attribute attached to it: a single positive, continuous variable. Such a variable could, in a real dataset, represent a person's height in a social network or a gene's expression level in a gene-gene interaction network. In this study, however, this quantity is a purely abstract and theoretical one in order to avoid familiarizing users with a real dataset. Thus, for each node, we draw a single value, y, from a standard half-normal distribution (i.e., $y \sim |N(\mu, \sigma^2)|$, where mean $\mu = 0$ and standard deviation $\sigma^2 = 1$) (Figure 5.4c). To visually communicate the node attribute's magnitude to the user, it is mapped to each node's surface area, i.e., the larger the node's attribute's magnitude, the larger its surface area.

Node Attribute Uncertainty For each such drawn attribute magnitude, we must now simulate an uncertainty around its value. In a real-world dataset, such uncertainties could represent anything from measurement error to inherent variation in the data [GS06]. Here, it is, again, a purely abstract measure to avoid confusing users with a real dataset. Thus, for each node, we simulate a single positive value, s, between zero and one using a random uniform distribution, i.e., $s \sim U(a, b)$, where bound $a = 0$ and bound $b = 1$ (Figure 5.4d). While a uniform distribution may not necessarily be representative of real-world uncertainties, it does ensure a sufficient amount of variation in uncertainty within each dataset to make each task sufficiently challenging and sufficiently different.

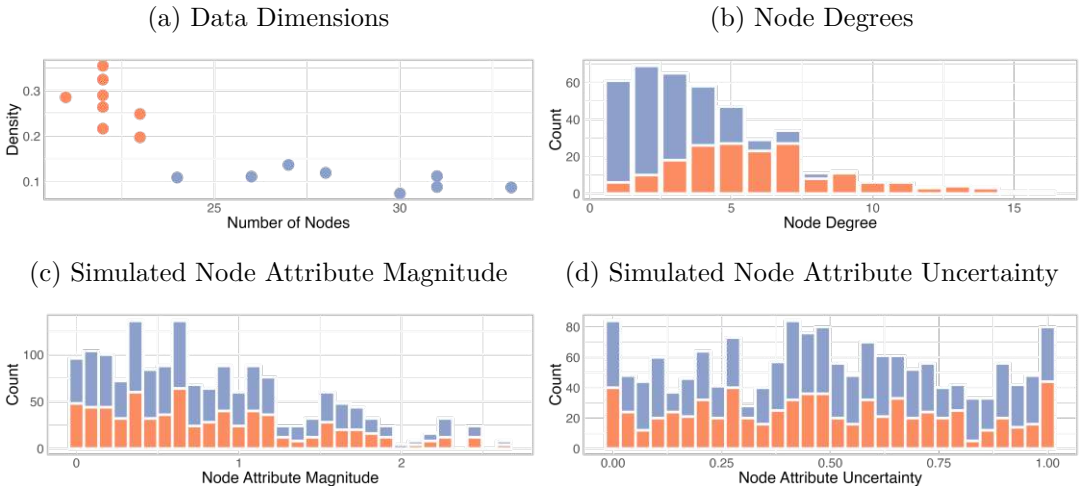


Figure 5.4: Chosen/Simulated graph datasets' properties, specifically their a) number of nodes and density, b) nodes' degrees, c) half-normally simulated node attribute magnitude, and d) uniform-randomly simulated node attribute uncertainty. Data properties are color-coded depending on the dataset, i.e., raccoons and ants.

5.4.4 Training

To prepare participants for the study, we employ pre-study training. Following the terminology laid out by Nobre et al. [NWHL20], we make use of *Passive Training*, i.e., text-based tutorials. Specifically, in these tutorials, we i) lay out how to read straight-line node-link diagrammatic network visualizations, and ii) define node attributes and their uncertainty, as well as iii) how these attributes and their uncertainties are visually represented to the user.

5.4.5 Study Procedure

In this section, we outline the various aspects of user performance and experience that were measured.

User Performance and Tasks Following the taxonomy of Lam et al. [LBI+11], we aim to evaluate user performance, i.e., study the speed and accuracy with which users are able to answer a series of low-level graph analysis tasks using a particular node attribute uncertainty visualization. Ultimately, we aim to quantitatively compare these speeds and accuracies between the four previously discussed uncertainty encodings (Section 5.4.1): The faster and more accurately participants are able to answer tasks with a particular visual encoding, the more "readable" that particular encoding is [GFC04, GFC05] A key challenge lies in the selection of appropriate tasks for participants to complete. We investigate eight tasks in total, four attribute tasks relating to attribute magnitude and their uncertainty, and four tasks relating to the topology of the provided graphs.

First, we aim to investigate the user's ability to visually **identify and locate nodes of extreme uncertainty and attribute magnitude**. Following the taxonomy of Lee et al. [LPP+06b], we investigate four "*Attribute Tasks*" "*On the Nodes*". Following the taxonomy of Amar et al. [AES05], we are particularly interested in tasking users with "*Finding Extrema*" in the data. Specifically, we aim to investigate four key tasks, namely

$\mathbf{A_{max}}$	Locate the node with the largest attribute magnitude.
$\mathbf{A_{min}}$	Locate the node with the smallest attribute magnitude.
$\mathbf{U_{max}}$	Locate the node whose attribute is the most uncertain.
$\mathbf{U_{min}}$	Locate the node whose attribute is the least uncertain.

As we anticipate nodes of greater attribute magnitude to be more readable than those of smaller magnitude, owing to their greater surface area, we deliberately control the assignment of attribute magnitude to attribute uncertainty. Additionally, we also anticipate the identification of particular nodes with higher attribute uncertainties to be more difficult, as these will be blurrier or wigglier. To control for these anticipated differences, a participant (for a particular uncertainty visualization approach) is assigned to either a "large" or "small" study run. In a "large" study run, the answers to A_{max} and A_{min} are those nodes with the *largest* attribute uncertainty. Similarly, for U_{max} and U_{min}, the answers are those nodes with the *largest* attribute magnitude. Conversely, in a "small" run, the answers to A_{max}/A_{min} and U_{max}/U_{min} are the nodes with the *smallest* attribute uncertainty and node attributes, respectively. In this way, all possible combinations of small and large attribute magnitudes and attribute uncertainties are controlled for and investigated.

Beyond communicating attribute uncertainty, an effective uncertainty encoding should not interfere with a user's ability to make sense of a graph's topology. Hence, second, we investigate the user's ability to **visually parse the topology of the presented graph** for a given uncertainty visualization encoding. Specifically, following the topology of Lee et al. [LPP+06b], we investigate two "*Overview*" and two "*Topology*" tasks:

$\mathbf{O_{nodes}}$	Estimate the number of nodes in the graph.
$\mathbf{O_{edges}}$	Estimate the number of edges in the graph.
$\mathbf{T_{max}}$	Identify the node with the most neighbors in the graph.
$\mathbf{T_{min}}$	Identify the node with the fewest neighbors in the graph.

The answers to the previously discussed "*Attribute-based* Tasks" should have no bearing on users' abilities to make sense of the graph's topology. Thus, for each task O_{nodes}, O_{edges}, T_{max}, and T_{min}, users are randomly assigned on of the previously discussed "*large*" or "*small*" study runs. For the exact phrasing of these questions, i.e., as they were presented to users, please refer to Table 4 in the supplement.

User Experience Following Lam et al.'s [LBI+11] seven scenarios, we also aim to investigate user experience, i.e., perceived effectiveness and preferences. To do so, at the end of the study, users are presented with five statements to be answered on a 7-point Likert Scale:

$\mathbf{S_{learn}}$	The uncertainty visualization was easy to learn.
$\mathbf{S_{use}}$	The uncertainty visualization was easy to use.
$\mathbf{S_{pleas.}}$	The uncertainty visualization was aesthetically pleasing.
$\mathbf{S_{quick}}$	The uncertainty visualization allowed me to answer questions quickly.
$\mathbf{S_{acc.}}$	The uncertainty visualization allowed me to answer questions accurately.

Participants were presented with exactly these statements.

Qualitative Feedback Finally, we investigate user performance and experience qualitatively to understand *why* we observe certain quantitative trends. After each task (Section 5.4.5), users were able to optionally explain how the assigned uncertainty visualization either assisted or hindered them in answering the task. At the end of the study, users provide two points of feedback either in favor or against the uncertainty visualization, explaining how it either assisted or hindered them.

5.4.6 Analysis

Quantitative Evaluation For each task, we record the participant's answer as well as the time taken to submit said answer. Based on the thus collected answers, the accuracy can be calculated for each task. For example, for task A_{max}, the accuracy is defined as the submitted answer's attribute magnitude divided by the actual (ground truth's) attribute magnitude. Alternatively, for task T_{min}, the accuracy is defined as the ground truth node's degree divided by the submitted node's degree.

Here, as assumptions of normality could neither be made nor (as will be discussed in Section 5.5) validated when probed with Shapiro-Wilk tests, we cannot make use of a conventional *ANOVA* for our statistical analysis. Instead, we turn to Wobbrock et al.'s *non-parametric aligned rank-transformed ANOVA* for our analysis [WFGH11]. The overall statistical impact of the uncertainty visualization approach on the time taken per task, task accuracy, and user experience is then assessed using an omnibus F-test. If significant, said F-test is followed by a series of pairwise comparisons of all uncertainty visualizations' estimated marginal means [EKHW21]. All tests were performed with a standard *a priori* Bonferroni-adjusted family-wise type-I error rate of $\alpha = 0.05$ [Bon36].

Qualitative Evaluation For each participant, we collect up to nine pieces of qualitative feedback, i.e., comments. These collected comments are then broken up into individual

utterances. Note that some comments may also consist of only a single utterance. These utterances are then analyzed initially individually and then jointly by three coders in iterative coding sessions [Chi97]. During the first individual coding session, each coder assigns a single code to each utterance as well as a positive or negative qualifier. These individual utterances and qualifiers are iteratively unified across coders until all coders reach 100% consensus.

5.5 Results

Here, we describe the quantitative and qualitative results obtained from the previously described user study.

5.5.1 Participants

In order to study the quantitative and qualitative effects of uncertainty encoding on user performance and experience, we conducted a large-scale, online user study. More specifically, we recruited 160 paid participants using the Prolific platform [noa14]: 20 per representation and dataset, i.e., 40 per uncertainty encoding total. Each participant was paid 11£per hour for about $15 - 20$ minutes of work. Of the 160 recruited participants, 83 self-identified as "male", 77 as "female". The average participant's age was 33. The youngest recruit was 18 years old, and the eldest was 72 years old. In terms of the highest completed academic degree, 38 had completed high school, 76 had obtained a Bachelor's degree, 44 had finished a Master's program, and 3 had completed a PhD. Finally, 29 participants report "No Experience" with networks or graphs, 44 "Little Experience", 43 "Some Experience", 37 "Good Experience", and 8 "Extensive Experience". When probed statistically, no significant association between experience level and task accuracy or time could be established.

5.5.2 User Performance

For each of the eight investigated tasks (Section 5.4.5), each participant's time as well as task accuracy was recorded, as visualized in 5.6 and 5.5, respectively. When statistically probed (Section 5.4.6), no statistically significant effects of uncertainty representation on either time or task accuracy could be detected.

5.5.3 User Experience

As discussed previously (Section 5.4.5), participants were required to answer five questions regarding their experience on a 7-point Likert scale. When probed statistically (Section 5.4.6), no statistically significant effect of uncertainty representation on user experience could be detected. The reader is directed to Figure 5.7 for more details.

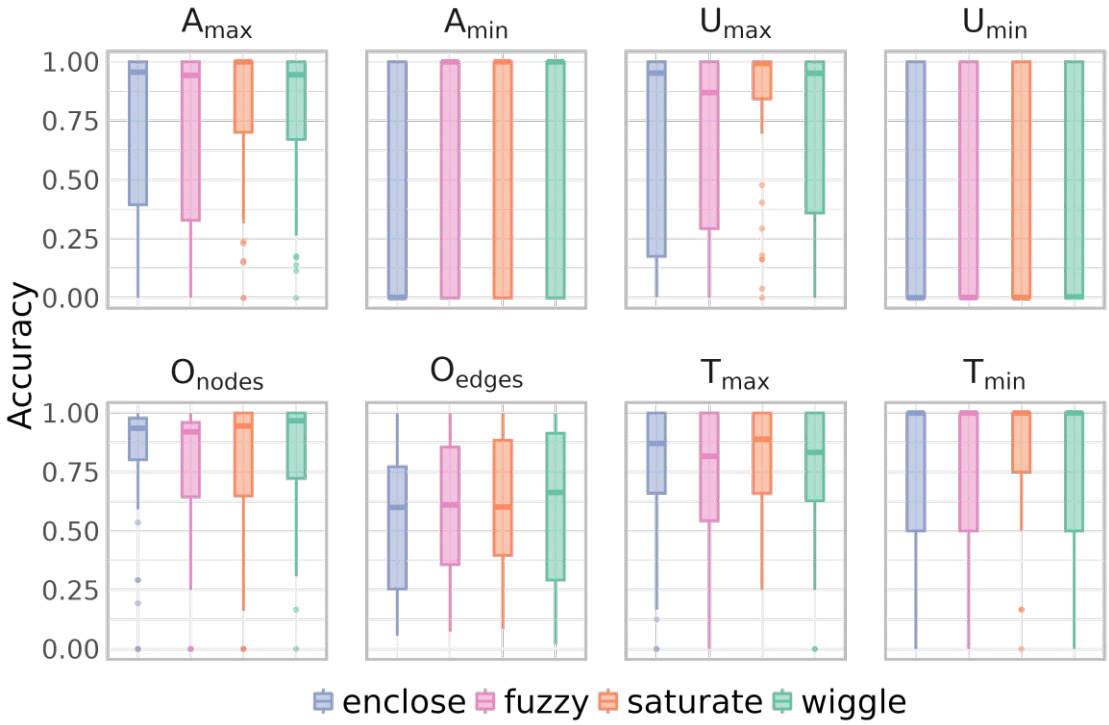


Figure 5.5: Participants' task accuracies visualized per task and per uncertainty encoding, represented as a box-and-whisker plot. The box's center represents the values' median, and its hinges the first and third quartiles. The whiskers represent 1.5 times the interquartile range.

5.5.4 User Feedback

As described previously (Section 5.4.5), we collected qualitative user feedback per task (optional) and summarily at the end of the study (mandatory). In total, 726 comments were collected from the 160 participants throughout the study. These comments were then broken down further into 893 utterances. During the first round of inductive coding, the three coders individually came up with 35, 99, and 27 different codes for the identified 893 utterances. These codes were discussed and unified until coders agreed upon 15 unique codes, grouped in 4 broader categories, namely *layout*, *effort*, *engagement*, and *usability* (Figure 5.8). During said discussion, utterances of no or little value were also removed (majority vote), leaving a total of 512 utterances. In the final deductive coding step, each coder assigned one of the 15 agreed-upon codes to each of the remaining 512 utterances. Moreover, each utterance was also labeled as either positive or negative (*−negative*, *+positive*). During a final meeting, the three coders discussed each of their deductively coded utterances until, for each of the remaining utterances, consensus was achieved. The results of this coding are presented in Figure 5.8.

Figure 5.6: Participants' task time taken visualized per task and per uncertainty encoding, represented as a box-and-whisker plot, in which the box's center represents the values' median and its hinges the first and third quartiles. The whiskers represent 1.5 times the interquartile range.

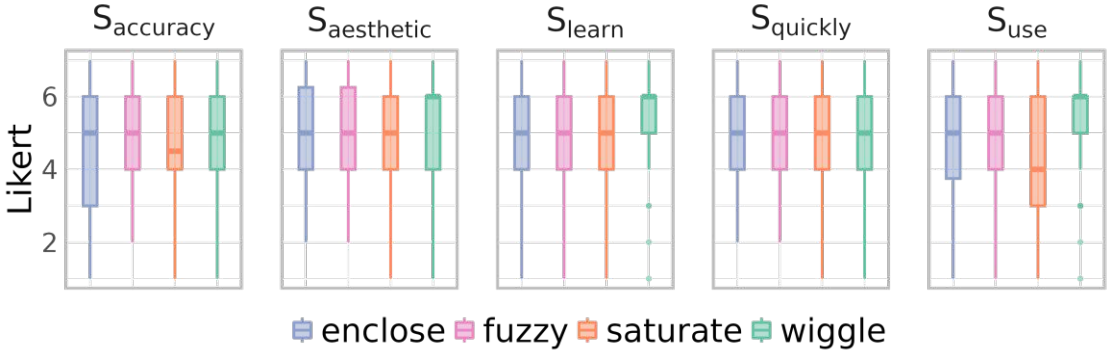


Figure 5.7: 7-point Likert scale results per question and per uncertainty encoding, visualized as a box-and-whisker plot, in which the box's center represents the values' median and its hinges the first and third quartiles. The whiskers represent 1.5 times the interquartile range.

Layout Layout (38 utterances) describes how negatively or positively the topological properties were perceived by users. This consists of the *node placement* (−11, +1) and *edge placement* (−25, +1). Examples hereof included comments such as "*[The visualization] hindered me just because there were a lot [of nodes]*" or "*[Edges] all look the same and sometimes almost overlap, I wasn't able to accurately count them*".

Effort Effort (69 utterances) describes how much effort a participant had to invest to answer questions using the presented uncertainty encoding. This includes the *cognitive load* (−47, +11), the *temporal effort* (−2, +5) required, and the *physical effort* (−4, +0) needed, e.g., "*[...] Negative: one has to give a maximum concentration to avoid mistakes.*", "*[...] it [took] time to find the correct answer*", and, "*[...] Negative: Some parts were straining on the eyes.*" respectively.

Engagement Engagement, the smallest category with only 35 utterances, describes utterances relating to active participation, commitment, or involvement. This comprises how *intellectually engaging* (−0, +13) and *affectively engaging* (−0, +9) it was, as well as how *aesthetically pleasing* (−1, +12) the uncertainty encoding was. Examples include, respectively, "*It was a very stimulating study [...]*", "*I really enjoyed this! [...]*", and "*I found the uncertainty visualization aesthetically pleasing [...]*".

Usability Finally, usability, the largest category with 214 utterances total, contains utterances related to how usable a particular uncertainty encoding is. This includes how *confident* (−26, +2) users were in their given answers, e.g., "*[the visualization] enabled me to provide accurate answers [...]*". It also included how easy or difficult users found *comparing* (−35, +8) different attribute levels or certainty, e.g., "*it was difficult to compare the [enclosure's] thicknesses*". Additionally, usability included how *intuitive* (−4, +8) the uncertainty encoding was, e.g., "*[...] Having never seen this before it was easy to pick up*". Moreover, it included how *understandable* (−18, +9) the encoding was, e.g., "*[The visualization led to] confusion or misinterpretation.*". Penultimately, it included how *readable* (−23, +45) the encoding was, e.g., "*The uncertainty visualization effectively communicated the uncertainty levels, making it easy to identify nodes with high or low certainty.*". Finally, it also included how easy or difficult it was to gain an *overview* (−2, +16) of the data, e.g., "*[...] it was easy to access the information with a quick look.*".

5.6 Discussion

In this section, we discuss the results, both quantitative and qualitative, of our crowdsourced user study.

5.6.1 Quantitative Results

Surprisingly, the four investigated uncertainty encodings (Section 5.4.1) fared equivalently in terms of their produced user performance (Section 5.5.2). Indeed, no statistically

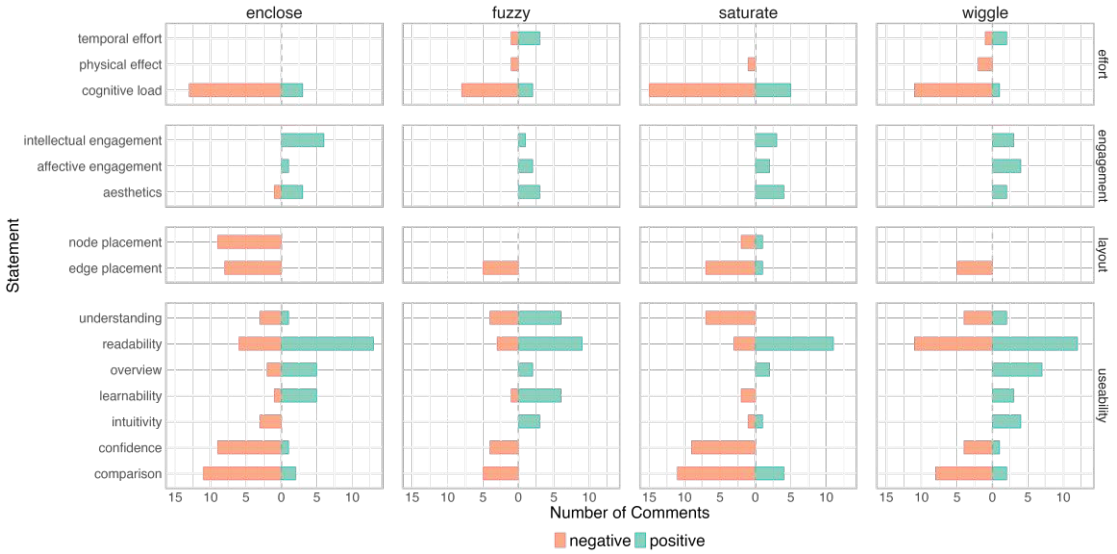


Figure 5.8: The number of negative and positive statements made by participants relating to i) effort, ii) engagement, iii) layout, and iv) usability (rows) for each of the four representations (columns). Comment counts by statement type are color-coded based on whether they are negative or positive.

significant effects of uncertainty encoding on either time taken or task accuracy, and subsequently no statistically notable pairwise differences, could be detected. These findings are corroborated by a visual inspection of the results (Figures 5.5 and 5.6). These results are interesting, as past investigations and evaluations have been rather critical of animation of a visual channel, especially within the context of uncertainty visualization. One possible explanation is that noted limitations, not only of animation, but also of saturation and fuzziness, ultimately balance each other out. Here, we briefly discuss the discussed disadvantages of these techniques in the literature in order to explain our observed results.

Limitations of Fuzziness First, fuzziness makes precise quantification difficult both from a mapping and cognitive perspective, i.e., the exact level of uncertainty may not be clear [GR04, CR00]. This is corroborated by utterances left by participants regarding the comparability of levels of uncertainty, e.g., "*There is not enough contrast to be accurate [when comparing relative levels of uncertainty.*" or "[...] *nodes can confuse you [as they look] same*". Second, fuzziness does not always work well in conjunction with other visual variables, e.g., shape or transparency: While intuitively communicating the presence of uncertainty, fuzziness may interfere with other visual channels and obfuscate their meaning and mapping [BKC+12]. Indeed, one participant explicitly mentioned that "*It's hard to tell apart the same uncertain level of attribute when [attribute] values are very much different*", highlighting the negative interaction between the uncertainty's fuzziness and the attribute level's node size. Finally, small differences may be hard to visualize

and detect by users of the visualization [GR04]. In line herewith, it would indeed appear as though detecting the least uncertain, i.e., the least fuzzy, node (U_{min}) proved difficult for participants (Figure 5.5), though this particular task proved difficult across all four uncertainty encodings.

Limitations of Saturation First, human perception of brightness/saturation is not linear, i.e., small changes are not perceived equally along the spectrum, which can lead to misinterpretation of uncertainty [War21]. Moreover, the number of meaningfully distinguishable levels of brightness/saturation is relatively small and thus not advisable for larger ranges [War21]. Looking at the quantitative results of our study (Figure 5.5), it is noticeable how, for all four visual encodings, participants struggled to identify the least uncertain node (U_{min}). For saturation specifically, this could be explained by this aforementioned non-linear perception as well as the poor distinguishability of different levels. Indeed, it is striking how many utterances were left lamenting the poor comparability of different saturation levels, e.g., "*Minor differences in [saturation are] not be visible to the naked eye*" or "[...] *it was hard to identify the saturation of some shades when it came to the nodes with least certainty* [...]". Third, brightness/saturation may also be easily confused with other variables, such as intensity, and may also interfere with chosen (background) colors [War21], though we noted no results, either quantitative or qualitative, in this regard. Fourth, brightness/saturation, when combined with other visual channels may be overwhelming to the user, i.e., it does not necessarily match well with other encodings [War21]. Again, looking at the aforementioned numbers of utterances left regarding comparability, it is interesting that a number of them complained specifically about this effect, e.g., "[...] *bigger nodes may appear more saturated than smaller nodes of the same saturation.*" or "*The more saturated, the larger the appearance.*".

Limitations of Animation First, animation has been argued to increase the cognitive load a user experiences, especially when multiple moving elements are present, i.e., as is the case with wiggliness [LLPY07]. With multiple nodes all wiggling simultaneously, a user could have had trouble comparing uncertainty levels across/between them. Looking at the qualitatively coded utterances left by participants (Figure 5.8), wiggliness proved to not be any more cognitively challenging than any of the other uncertainty encodings, though some participants did explicitly mention the added effort required, e.g., "[...] *Excessive wiggliness can become visually distracting or overwhelming, especially in dense or complex networks* [...]". Second, similar to fuzziness, the mapping of uncertainty to wiggliness may make quantifying exact levels of uncertainty difficult [LLPY07]. Here, in line herewith, it is interesting to note that a number of participants did indeed leave comments such as "*There wasn't much of a difference in wiggliness between a few nodes, so it was hard to determine which was moving the most*". Third, animation is fundamentally limited by its temporal resolution with regard to a user's cognitive limits. If the speed of the animation does not match a user's viewing speed or attention, the user may miss critical details [HR07, TMB02]. However, we could not identify any particular qualitatively coded utterances that would support a claim for additional

temporal demand. Fourth, motion and animation, more generally, are strongly associated with changes over time. This may cause some confusion in users with such preconceived associations [HR07, TMB02], though we did not observe any comments corroborating this claim. Finally, animation has been found to induce visual fatigue over time, which may negatively impact the accuracy and speed with which users are able to navigate and parse the visualization [LLPY07]. While one participant did explicitly describe the experience of working with wiggliness as "[...] *straining on the eyes.*", we were unable to identify any additional cognitive demand, physical effect, or cognitive load.

5.6.2 Qualitative Commentary

Here, we enumerate several interesting observations regarding the qualitative feedback received from users. Where possible, we use these qualitative results to inform or explain our observed quantitative results.

Learnability and Intuitivity A key question underlying any (novel) uncertainty encoding is how easily it is to comprehend and whether it is truthful to the data it should actually represent. Here, we look to users' responses regarding an encoding's *learnability* and *intuitivity*. Interestingly, wiggliness was viewed exclusively positively in terms of its learnability (W: +3) and intuitivity (W: +4), described by two participants as "[...] *easy to figure out*" or "[...] *easy [to] interpret*". Comments left regarding enclosure, fuzziness, and enclosure, on the other hand, were not always positive regarding their learnability (E: [−1, +5], S: −2, F: [−1, +6]) and intuivity (E: −3, S: [−1, +1], F: +3). Given these small differences, however, we argue that the four uncertainty representations are ultimately comparable in terms of both their learnability and intuitivity. Unless the target user group is strongly familiar with a particular representation of uncertainty, we argue that animated wiggliness could form an equally learnable/intuitive representation as conventional alternatives, at least for the kinds of data and tasks presented here, i.e., relatively simple, low-level tasks [LPP+06b] applied to small to intermediate-sized graphs [YAD+18]. It remains to be seen whether wiggliness (and animation more generally) is a suitable representation of uncertainty for more complex datasets and tasks.

Understanding and Readability Related to intuitivity and learnability, *understanding* and *readability* describe how well participants were able to utilize the presented uncertainty encoding. Regarding understanding, the four representations are fairly comparable (E: [−3, +1], F: [−4, +6], S: −7, W: [−4, +2]). However, when looking at readability, we note that those using wiggliness were more critical (W: [−11, +12]) than those using the other three encodings (E: [−6, +13], F: [−3, +9], S: [−3, +11]). Comments left regarding wiggliness' poorer readability included "*certain movements/wobbles can create an optical illusion almost* [...]" or "[...] *the visualization did not clearly show the uncertainty or variability in the data* [...]" However, for the smaller to medium-sized graphs presented here, this difference in perceived readability had no impact on users' quantitative results (Figure 5.5). However, it is worth asking whether, with larger and

denser graphs, wiggliness could lead to a statistically notable decrease in either user performance or experience. We argue that this is broady in line with the work of Lundström et al. [LLPY07] who concluded that "*animation methods [were an] effective approach for uncertainty visualization*". We thus argue that experimental follow-up work is needed to study the impact of wiggliness on user understanding and performance for more complex datasets and tasks.

Overview *Overview* describes how easily a participant was able to quickly navigate the visualization in order to identify core aspects of the data. Here, we note that wiggliness (W: +7) was noted exclusively positively and more often than the other uncertainty encodings investigated (E: [−2, +5], F: +2, S: +2). Comments left by users of wiggliness noted the ease with which they were able to quickly identify regions of high and low uncertainty, e.g., "[...] *at a quick glance large differences in uncertainty can be easy and obvious to spot due to differences in wiggling* [...]" or "[...] *The movement draws attention to uncertain nodes, helping prioritize focus during analysis.*". Such utterances are in line with the speculation of Ehlschlager et al. [ESG97], who posited that audiences may find animation useful for "quick qualitative impressions of the magnitude of uncertainty". As noted previously, (smaller) differences were more difficult to detect using wiggliness. This could point towards wiggliness being useful for big-picture, qualitative impressions of uncertainty instead of fine-grained comparisons of smaller differences, i.e., allowing users to quickly identify (groups of) nodes with high or low uncertainty.

Confidence and Comparability *Confidence* and *Comparability* describe the degree to which study participants were sure of their answers, particularly when comparing nodes' attribute levels and attribute uncertainties. In general, users expressed low confidence in their given answers across uncertainty encodings (E: [−9, +1], F: −4, S: −9, W: [−4, +1]). Across representations, participants also expressed difficulty in comparing attribute levels and uncertainties (E: [−11, +2], F: −5, S: [−11, +4], W: [−8, +2]). This low confidence and difficulties in comparing node attribute levels and uncertainty may explain the generally poor results of participants for identifying the smallest attribute node (A_{min}) and the least certain node (U_{min}) (Figure 5.5). As visual representations of uncertainty often aim to communicate relative levels of uncertainty [CCP07], these poor results are somewhat disconcerting. The visualization shown to users was static, limiting the ease of comparison. An interactive system, e.g., with hover or click-based details, could aid comparison of uncertainty levels and boost user confidence.

Effort As observed by Lundström et al. [LLPY07], animation (for the purposes of uncertainty animation) caused more visual fatigue owing to the "movement and flickering of the image". However, when looking at the number of comments left by users regarding the *cognitive load*, wiggliness (W: [−11, +1]) fared equivalently to the other three representations (E: [−13, +3], F: [−8, +2], S: [−15, +5]). Regarding wiggliness' cognitive load, one participant did write that "*[wiggliness was an] information overload.*", while another commented that "*There was a lot going on to focus on which can make*

decision making difficult [...]". Interestingly, negatively coded utterances regarding the cognitive load for the other three uncertainty encodings were less specific, describing their experiences generally as "*hard*", "*tough*", or "*difficult*". We also noted no meaningful differences in the perceived *physical effect* (E: −1, F: −0, S: −1, W: −2). However, one participant, in line with Lundström et al.'s [LLPY07] observations, did explicitly mention that wiggliness proved to be "[...] *straining on the eyes.*". Here, we speculate, based on other qualitatively coded utterances, that participants (unfamiliar with network visualizations) were already overwhelmed by the complexity of the graph's topology (discussed next), meaning the uncertainty encoding itself did not add any notable additional cognitive load or physical effort. Here, it is worth asking whether a larger and more complex network, with more complex patterns of node attribute uncertainty, could have revealed differences in the effort required as well as the physical effect on users. A graph of hundreds of nodes, each wiggling at different rates, could indeed prove taxing to parse visually.

Topology Utterances related to *node placement* and *edge placement* described the ease or difficulty with which users were able to parse the topology of the presented graph. Comments were mostly negative across all four representations, for both node (E: −9, S: [−2, +1]) and edge placement(E: −8, F: −5, S: [−7, +1], W: −5). Interestingly, enclosure was the only uncertainty encoding with many negative comments relating to node placement. However, making sense of the edges in the drawing was generally regarded as challenging, i.e., negative. In the case of wiggliness specifically, one participant expressed the difficulty they experienced given the constant movement of the edges. In the case of fuzziness, one participant remarked that the fuzzy border made it difficult to determine whether an edge was or was not incident to a particular node. A denser graph, i.e., a graph with more edges, and subsequently more edge movement per frame, would probably make parsing the graph's topology even more challenging. It is thus worth asking whether, in the context of topological analyses of networks, animated wiggliness would be best suited for either smaller graphs or subgraph views only.

5.7 Limitations and Future Work

Alternative Encodings In this chapter, based on the findings of our literature survey (Section 5.4.1), we focused explicitly on three non-animated visual encodings of uncertainty: saturation (S), enclosure (E), and fuzziness (F) However, as made clear in Figure 5.2, many other possible (static) representations of uncertainty present themselves, such as color hue, direct text labels, or opacity. It is hence important to note that our evaluation of wiggliness is by no means exhaustive. Follow-up studies should additionally investigate and compare such visual channels in order to more completely gauge the effectiveness of these representations for uncertainty visualization in networks.

Interactions between Encodings In this chapter, we focused on the visualization of one node attribute and its attached uncertainty in isolation. However, in real (domain-

specific) applications, a visualization may need to visually map not simply one but multiple node attributes to various visual channels. Additionally, each of these attributes may bring with it its own uncertainty. Alternatively, a single attribute may have multiple (types of) uncertainties attached to it. In such cases, the interaction between these selected visual channels becomes important. For example, the use of node fuzziness may make understanding node shape difficult. Alternatively, the use of node brightness may complicate a user's understanding of node color. Future investigations of node attribute uncertainty visualization should include such interactions for the sake of completeness. Here, it is worth asking whether and how wiggliness interacts with other common visual channels.

Investigating Cognitive Task Load In this chapter, we made an effort to thoroughly investigate both user performance and experience from a quantitative and qualitative perspective. However, one aspect that has not been explicitly investigated was the cognitive task load of these individual uncertainty visualizations. According to past studies, though not our own qualitative results, animated wiggliness should have conceptually resulted in a greater cognitive load compared to static representations. Future work should investigate the effect of representation on (cognitive) task load using, for example, NASA's TLX [HS88], to better understand when and how to use them. Additionally, future work could combine task-based user studies with eye-tracking in order to understand where a user focuses their attention.

Task and Data Complexity In this chapter, we deliberately focused on simple, low-level graph analysis tasks and small, abstract networks to evaluate the baseline effectiveness of wiggliness as an uncertainty encoding. While our results suggest that wiggliness can be a conceptually useful approach in these controlled settings, future work must explore its applicability under greater complexity—both in task demands and data scale. For example, domain-specific tasks such as identifying the most certain paths or clusters may reveal different strengths or limitations of wiggliness. Likewise, larger and denser graphs introduce new challenges: as visual complexity increases, wiggliness may either become imperceptible or introduce excessive node occlusion due to increased spatial demands. Understanding how uncertainty visualizations scale with data and task complexity is essential to their broader adoption in real-world applications.

5.8 Conclusion

In this paper, we studied the impact of animated wiggliness for node attribute uncertainty visualization on user performance and experience in a mixed methods setup in a large-scale, crowd-sourced user study of 160 participants. Despite animation's infrequent use and recommendation in literature, we find animated wiggliness to perform equivalently to saturation, enclosure, and fuzziness. Interestingly, when investigating qualitative feedback received from participants, we find that, at least for the comparably simple datasets and tasks presented here, wiggliness is as learnable, intuitive, and understandable as

all other investigated representations. We do, however, note several comments by users that hint at wiggliness' possible shortcomings, such as inducing visual fatigue and being poorly readable in certain situations. These shortcomings may limit wiggliness' utility for larger and more complex graphs and tasks. Nonetheless, the results of this study point towards the potential utility of wiggliness for uncertainty visualization in networks. In future work, we aim to investigate the efficacy of animated wiggliness in larger and denser networks in order to evaluate the approach's scalability and utility in practice.

CHAPTER 6

Beyond the Individual: The Current State of Compound Graph Visualization

In multivariate biological networks, node attributes frequently take the form of group structures. In addition to topological relationships, nodes share group-level relationships. Examples of such group structures may include so-called "Single Nucleotide Polymorphisms" (or SNPs) that fall within genes, or genes and metabolites that fall within metabolic pathways. The key challenge here lies in clearly communicating group-level associations and relationships, such as inclusions, exclusions, and intersections, while also effectively visualizing topological relationships.

In order to characterize the current state-of-the-art we conduct a thorough survey of the literature, consisting of both newly collected papers as well as papers collected by past reviews, and categorize them within a newly developed taxonomy which explicitly differentiates between i) the *visual relationship* between graph topology and group structure, and ii) the *visual encoding* chosen for the group structure. With the results of this survey, we hope to locate notable gaps, also relevant to the domain of biological network visualization, ultimately to assist the development of novel (application-driven) techniques. The contents of this chapter are based on the position paper *"Visualizing Group Structure in Compound Graphs: the Current State, Lessons Learned, and Outstanding Opportunities"* [EMWR24], published in the conference proceedings of the *2024 VISIGRAPP* conference, with myself as first author, in collaboration with Diana Marin, Hsiang-Yun Wu, and Renata G. Raidou.

6.1 Overview

Compound graphs are common across domains, from social science to biochemical pathway studies, and their visualization is important to both their exploration and analysis. However, effectively visualizing a compound graph's topology and group structure requires careful consideration, as evident by the many different approaches to this particular problem. To better understand the current advancements in compound graph visualization, we have consolidated and streamlined the existing surveys' taxonomies. More specifically, we aim to disentangle the visual relationship between graph topology and group structure from the visual encoding used to visualize its group structure in order to identify interesting *gaps in the literature*. In so doing, we are able to enumerate a number of *lessons learned* and gain a better understanding of the outstanding *research opportunities* and *practical implications* across domains.

6.2 Introduction

Compound graph data, their visualization, and analysis are common across many different fields: from social networks [FAM+11], to biochemical pathways [PLS+13], to business analytics [AB08] and transportation logistics [MAU15]. In social network analysis, for example, researchers are interested in understanding a person's role across social groups. Here, compound graphs can be used to model individuals as nodes and their relationships as edges, as well as their place in various types of groups, such as circles of friends, roles within a workplace, or associations with organizations. For another example, in the context of biological pathways, domain experts are interested in understanding the mechanistic relationships between individual genes and groups of genes and metabolites, in order to understand the biochemical underpinnings of disease or cell function. No matter their application area, compound graphs are a useful framework for probing and understanding networks whose nodes also share group-level relationships.

However, compound graphs are challenging to visualize, as researchers are interested in understanding such graphs on both a topological and a group level for increasingly large datasets. Thus, any visualization of compound graphs must tackle the challenge born of trying to balance the visual communication of both entity topology and group structure. Various visualization approaches and systems have been put forth in the literature, each tackling this challenge differently: Some forgo interactivity in the interest of scalability [DDPA15], others aim to combine the two using summarization or linked views [DS13], while others yet build upon domain-specific visual conventions to better serve a particular user group [LSKS10].

While existing reviews have taxonomically summarized these various visualization approaches for compound [VBW15], multivariate [NMSL19], multilayered [MGM+19], and dynamic [BBDW14] graphs, we note that none had classified compound graph visualization strategies along the more abstract axes of general (comparative) visualization [KCK17]. We argue that doing so allows for a more general understanding of literature gaps and the identification of new, interesting research opportunities.

In this chapter, we identify and discuss lessons learned and novel research opportunities within the context of group structure visualization in compound graphs. To better understand the current state-of-the-art, we draw upon existing taxonomies developed in recent surveys, namely i) Vehlow et al.'s [VBW15], Beck et al.'s [BBDW14],Nobre et al.'s [NMSL19], and McGee et al.'s [MGM+19] reports on (particular types of) compound graphs, ii) Alsallakh et al.'s [AMA+16] survey on set-typed data, and finally iii) in order to position compound graph visualization within a comparison-oriented framework, Kim et al.'s [KCK17] report on comparative strategies. These reviews were selected as, based on their impact on the field, we believe them to be authoritative and representative. These state-of-the-art reports and their taxonomies share notable overlap, and all shed light on various facets of the group structure visualization challenge in compound graphs. We argue, however, that none, by themselves, provide a complete view of the current opportunities. Instead of providing a meta-review, here, we consolidate and extend these taxonomies to provide a comprehensive overview of the topic. This is to aid us in identifying both major lessons learned from past research in the field as well as open challenges across compound graph visualization domains not yet discussed by these aforementioned reports.

Specifically, we identify three core axes along which to categorize literature, namely

1. the *visual relationship* between group and topological information, inspired by Kim et al. [KCK17],
2. the *visual encoding* chosen for the graph's group structure, inspired by Alsallakh et al. [AMA+16],
3. the kind of *group-level relationships* admitted by the approach in question, as investigated by Vehlow et al. [VBW15].

We collect 167 references, partially based on the bibliographies of existing reviews, and place them within our taxonomy. Based on our findings, we identify gaps in the literature, outstanding challenges, and lessons that can be useful for domain researchers. We also identify several novel research opportunities.

6.3 Related Work

6.3.1 Compound Graph Visualization

Dynamic Graphs Dynamic graphs describe the evolution of entities and their relationships over time. Beck et al. [BBDW14] survey and taxonomically classify current

approaches to the visualization of such graphs. They identify three families of approaches, namely i) *animation*, i.e., the mapping of time to time, as seen in the work of Ma et al. [MKF+15], ii) *timelines*, i.e., the mapping of time to space, exemplified by *MatrixFlow* [PS12], and iii) *hybrid* approaches that combine the two, such as *Small Multiples* [BHRD+15]. While these families of approaches can indeed be utilized for general compound graph visualization, their work naturally does not discuss this in much detail. Nonetheless, they identify a large set of relevant techniques, approaches, and applications of dynamic graph visualization, and a number of their findings are mirrored in more general surveys [VBW15, MGM+19].

Multilayer Graphs Multilayer networks are a general framework describing various group-level relationships of both edges and nodes [KAB+14]. McGee et al. [MGM+19] survey the state-of-the-art of visualizing such networks. They classify papers collected based on the visualization method: i) *1D representations*, e.g., circular [BSH13] or axis-based [KBJM12] approaches, ii) *2, 2.5, and 3D node-link diagrams* which often use color [AMA07b] or linked views [RMM15] to communicate group structure, iii) *matrix-based visualizations*, such as *Termite* [CMH12] or *MuxVis* [DDPA15], iv) *hybrid* approaches, such as the matrix/node-link diagram *NodeTrix* [HFM07], and v) *summary* approaches, such as *Graph Thumbnails* [YDK+18].

Multivariate Graphs Multivariate graphs are collections of nodes and edges with additional data attached to them, such as group-level associations. Here, Nobre et al. [NMSL19] classify existing multivariate graph techniques along their view-, layout-, data-operations, and layouts. While they do not specifically address the visualization of group-level information, a number of their taxonomy categories are mirrored by other surveys discussed here. Most notably, they discuss three types of view operations: juxtaposed, integrated, and overloaded. Nonetheless, their recent publication contains a number of papers featured in other state-of-the-art reports that are worth classifying in our own taxonomy.

Compound Graphs Finally, Vehlow et al. [VBW15] survey approaches to the visual communication of the group structure of compound graphs. They ultimately identify four meta-categories to describe different approaches with which to visualize such group structures, namely i) *visual node attributes*, i.e., the encoding of group structure in the form of glyphs or color in the compound graph's embedding, exemplified by *TopicPanorama* [WLL+16] and *NetworkAnalyst* [XGH15], respectively, ii) *juxtaposition*, i.e., the separate visualizing group structure in either a linked view or attached to the graph's embedding, as seen in the works of Burch et al. [BSW13] and Zhou et al. [ZXQ15], respectively, iii) *superimposition*, i.e., the overlaying of group structure atop the graph's embedding using regions or lines, such as *Kelp Diagrams* [DvKSW12] and *LineSets* [PF15], respectively, iv) *embedding*, i.e., the drawing of a separate graph with which to communicate group structure relations, using, for example, hypernode summarization [CDA+14] or hybrid graph embeddings [HFM07, ADM+19].

Summary All four of these categorizations, while useful, are complementary to each other, and all, to some extent or another, conflate i) the *visual relationship* between topological and group-level information, i.e., how a graph's group structure is visualized relative to the topology's visualization, and ii) the group structure's chosen *visual encoding*, i.e., how a graph's group structure, not its topology, is visually represented. Subsequently, we aim to build upon, unify, and extend them to produce a novel perspective on the current state of compound graph visualization.

6.3.2 Visualizing Group Structure

Alsallakh et al. [AMA+16] study different strategies for visualizing group-level information for sets and their elements. While this taxonomy was not developed with compound graphs in mind, there is notable overlap with, and opportunities for extending, the aforementioned compound graph visualization taxonomies (Section 6.3.1). They identify six types of visualization strategies with which to encode group structure, namely i) *Euler/Venn* diagrams that represent each set as a closed curve and intersections as the overlap between them, ii) *Overlays*, which, given some embedding of each element as a point in 2D (or 3D) space, overlay set-membership using (colored) glyphs, lines, or region overlays, iii) *node-link diagrams* that represent each set and element as a node and draw bipartite edges between them to communicate set-element-membership, iv) *matrices* which place either elements and sets along rows and columns to communicate identity, or place sets/elements along both rows and columns to communicate similarity, v) *aggregation-based representations* that do not visualize all elements, but instead only show the number of elements per set in order to provide an effective overview of larger datasets, vi) *scatter plots*, a special type of aggregation, which embeds each set as a single point in 2D or 3D space as a function of some similarity metric or data attribute.

6.3.3 Comparative Visualization

Lastly, Kim et al. [KCK17] describe four different approaches to comparative visualization of 3D and 4D spatial data. While not complete on their own, we argue that these abstract classes of comparative visualization can be meaningfully mapped to group-structure visualization approaches in compound graphs. More specifically, we argue that the compound graph visualization problem can be understood as a comparison of graph topology on the one hand and group structure on the other. Four classes of approaches to comparative visualization are identified. *Juxtaposition* describes the side-by-side visualization of objects to be compared. *Superimposition* describes overlaying the objects to be compared. *Interchangeability* describes the interactive or animated scrubbing through objects. Finally, *explicit visualization* is the visualization of some derived (summary) quantity instead of the objects directly.

6.4 Paper Collection

In order to obtain a representative set of papers and applications, key references selected from the bibliographies of four relevant state-of-the-art reports, i.e., Nobre et al. [NMSL19], McGee et al. [MGM+19], Vehlow et al. [VBW15], and Beck et al. [BBDW14] were consolidated. Interestingly, despite the relatedness of the topics, we note little overlap in general between these four reviews, with the possible exception of Beck et al. [BBDW14] and Vehlow et al. [VBW15] (Figure 6.1). It should, however, be noted that the inclusion of references from existing works may introduce a certain bias to our literature review. Thus, in order to include more recent papers not present in these aforementioned works, we additionally manually curated an additional set of references from relevant venues, such as *IEEE TVCG*, *Computer Graphics Forum*, *PacificVis*, *Graph Drawing and Network Visualization*, and *Information Visualization*. In total, 167 papers were collected and subsequently manually categorized based on their visual relationship (Section 6.5.1), visual encoding (Section 6.5.2), group structure (Section 6.5.3), as well as their application area. The final set of papers and their categorization have been made publicly available on GitHub.

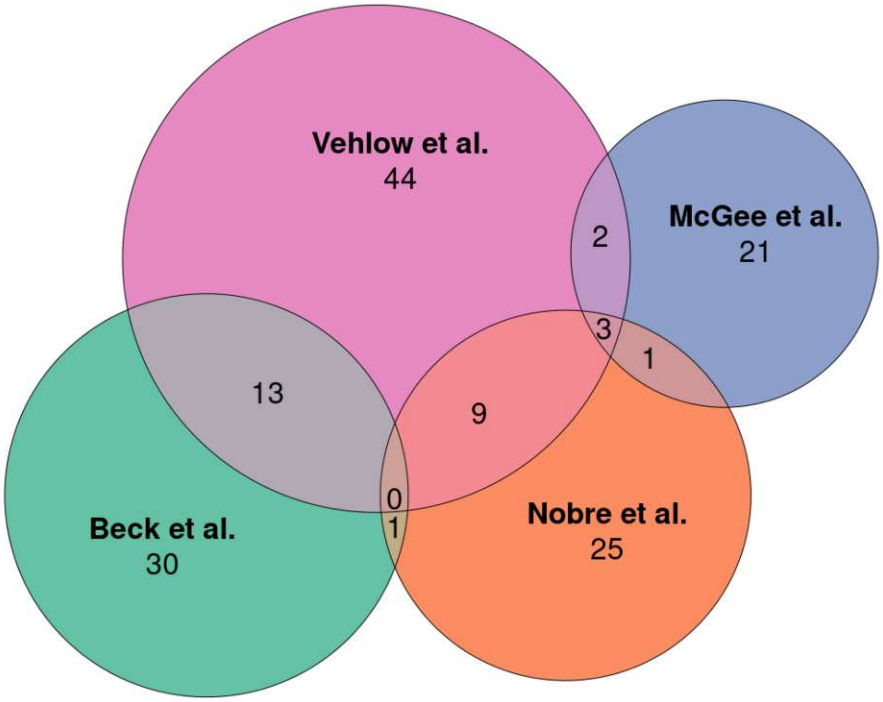
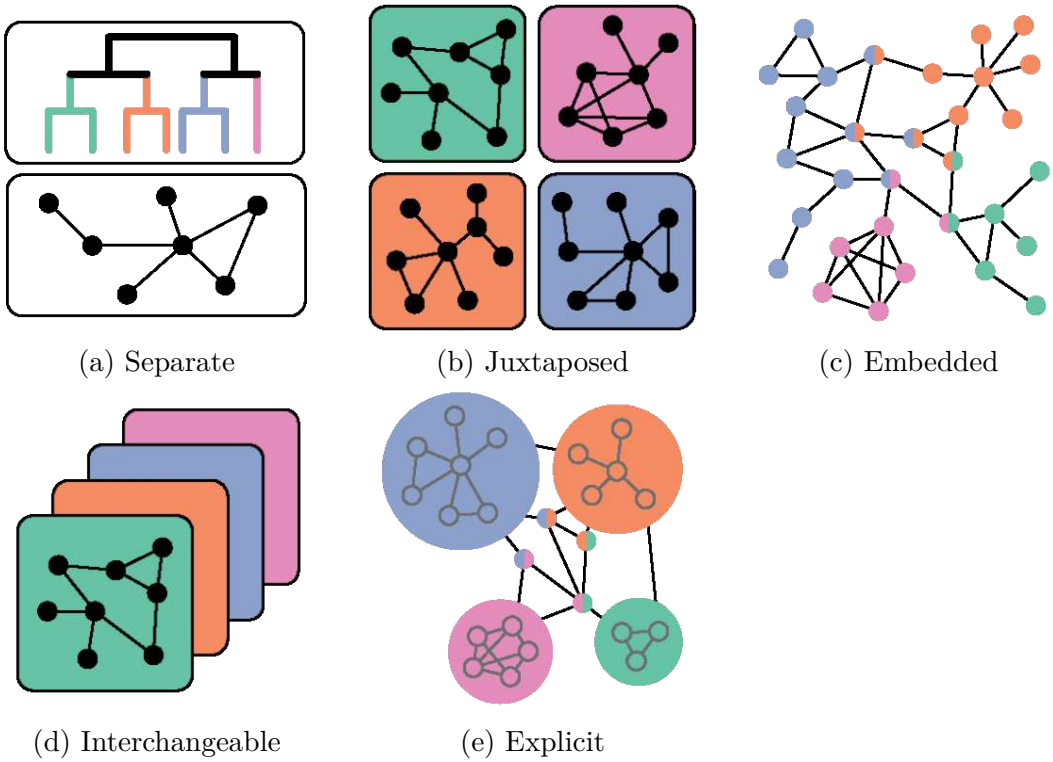


Figure 6.1: Sources and overlap of the selected papers, collected from Beck et al. [BBDW14], Nobre et al. [NMSL19], Vehlow et al. [VBW15], and McGee et al. [MGM+19], illustrated as an area-proportional Euler diagram. Among other goals, we aim to unify their collected references in addition to extending this collection of literature.

6.5 The Current State

Drawing upon these taxonomies, we propose to categorize papers along three main "axes", namely i) the chosen *visual relationship* between groups and their elements, ii) the selected *visual encoding* with which to communicate the graph's group structure itself, and iii) the *group structure* of the data.

6.5.1 Visual Relationships



(a) Separate (b) Juxtaposed (c) Embedded

(d) Interchangeable (e) Explicit

Figure 6.2: Examples showcasing the five identified visual relationships between group-level and topological encodings. **(a) Separate** relationships describe the visualization of group structure independently of any (topological) embedding of the compound graph. **(b) Juxtaposed** relationships place each group's subgraph side-by-side, allowing for straightforward comparisons between them. **(c) Embedded** relationships embed group-level information within/atop a compound graph's drawing. If a particular node forms an intersection between two groups, its glyph is accordingly colored. **(d) Interchangeable** relationships only visualize one group's subgraph's topology at a time, but allow for a user to scrub through them linearly. **(e) Explicit** relationships, rather than displaying all topological and group-level relationships, visualize only a composite of the two.

A compound graph's visual relationship describes how a graph's group structure is visualized relative to its topology (Figure 6.2). Here, combining Kim et al.'s [KCK17] and Vehlow et al.'s [VBW15] taxonomies, we identify five possible visual relationships.

Separate Defined by Vehlow et al. [VBW15] as "*Partitioned*", a *separate* visual relationship describes the visualization of group structure in a separate (possibly linked) view, such that the graph's global topology and group structure can be investigated independently. Such representations are especially useful if the graph's group structure is too complex to visualize atop or within its topological embedding. Consider, for example, the tree attached to a node-link diagram seen in *ASK-GraphView* [AvHK06].

Juxtaposed Equivalently found in Vehlow et al.'s [VBW15] taxonomy under "*Superimposed / Partitioned*", a *juxtaposed* visual relationship [KCK17] describes the side-by-side visualization of each group's subgraph's embedding in separate, possibly linked, views. Such representations are beneficial when a set of topologically similar graphs is to be compared side-by-side. For example, *Entourage* [LPK+13] opts to visualize different pathways side-by-side in a juxtaposed manner.

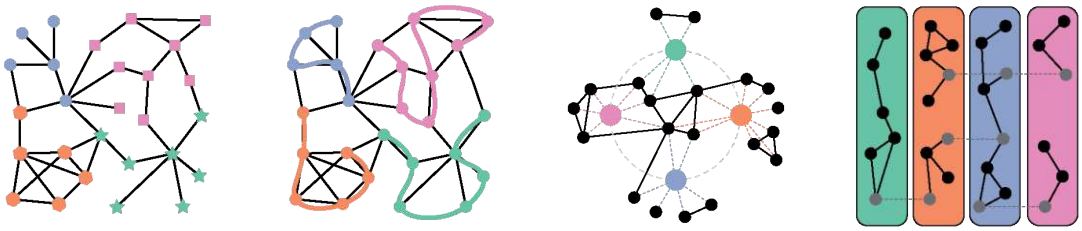
Embedded Defined as "*Superimposed*" by Kim et al. [KCK17] and "*Visual Node Attribute*" as well as "*Superimposed / Overlay*" by Vehlow et al. [VBW15], an *embedded* visual relationship defines the simultaneous visualization of graph topology and group structure in a single view, be it through color or region overlays or an explicit axis. Such representations can be useful, especially in explorative analysis endeavors, where one must first locate areas of simultaneous topological or group-structural interest. *TopicPanorama* [WLL+16], for example, visualizes group assignments as embedded glyphs.

Interchangeable Not explicitly found in Vehlow et al. [VBW15], an *interchangeable* visual relationship [KCK17] describes the visualization of each group's subgraph as a separate "slice" in a pile of linearly arranged slices that are traversed interactively, or automatically, using animation. Similar to juxtaposed representations, such visual relationships lend themselves well to immediate comparisons of pairs of subgraphs, especially when a compound graph's group structure can be linearly arranged, e.g., time slices in dynamic graphs. Consider, for example, the small multiples of adjacency matrices presented in the work of Bach et al. [BHRD+15].

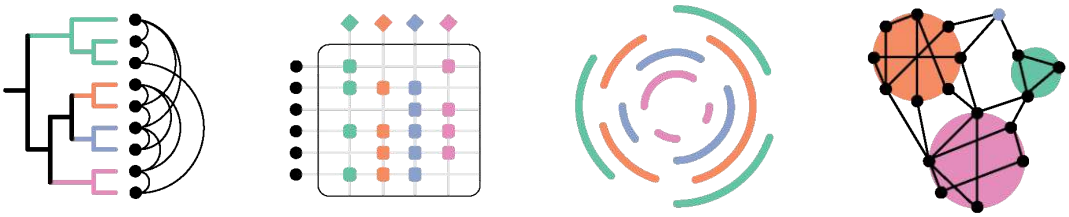
Explicit Defined by Vehlow et al. [VBW15] as "*Embedded*", an *explicit* visual relationship Kim et al. [KCK17] describes the visualization of some computed characteristic, such as differences or averages, or similarity, in order to provide a simpler summary visualization. These representations are especially useful for highly complex or large compound graphs, where the simultaneous visualization of group structure and topology (in a single view) is not feasible or useful. For example, Sallaberry et al. [SZPM10] opt to visualize each group and their intersections as hypernodes and regular nodes, respectively.

6.5.2 Visual Encoding

The visual encoding describes here how a graph's group structure—not its topology—is visually represented, i.e., how set membership, intersections, and exclusions are visually communicated to the user, as seen in Figure 6.3. Drawing primarily from Vehlow et al. [VBW15] and [AMA+16], we identify and define eight approaches with which to visually communicate nodes' group associations.



(a) Node Attribute (b) Overlay (c) Bipartite N.L.D. (d) Multiples



(e) Trees (f) Matrix (g) Abstraction (h) Hybrid

Figure 6.3: Examples of the eight identified visual encodings of group structure. **(a) Node-Attribute** encodings simply visualize nodes' group memberships using those nodes' color and/or shape. **(b) Overlay** encodings visualize group membership as regions or lines atop the nodes in the graph embedding. **(c) Bipartite node-link diagram** encodings visualize groups as additional nodes within the graph's embedding. Group membership is communicated using bipartite edges connecting group nodes to topological nodes. **(d) Multiples** encodings visualize each group's subgraph separately in its own "tile". **(e) Tree** encodings communicate (hierarchical) group structure between entities, visualized as leaf nodes, as hierarchical nodes whose edges indicate super/sub-set relationship between them. **(f)** Identity **matrix** encodings communicate set membership tabularly, where if a node (row) pertains to a group (column), the respective cell is filled. **(g) Abstraction** encodings provide an overview of groups' cardinalities and their relationships by abstracting away the individual elements that make up said group. Intersections are communicated through angular overlap between lines. Finally, **(h) hybrid** encodings combine any of the above encoding types, e.g., a combination of multiples with node-link diagrams.

Node Attribute Described as "*Overlays*" by Alsallakh et al. [AMA+16], *node attributes* [VBW15] are characteristics, here group membership(s), of a compound graph's

elements, which can be utilized to alter the visual attributes of each node during the embedding of the graph. *Color* is arguably the most common such node attribute with which to communicate group/set/cluster membership, as it is simple to implement and understand. Similarly, *shapes* can also be used. Lastly, *glyphs* are useful for more complex element-set memberships and relationships, as they can, for example, express such relationships as pie charts instead of simple shapes or colors, as seen in *TopicPanorama* [WLL+16]. In general, node attributes are a non-invasive, if limited, approach to visually communicating group memberships and relations without affecting the graph's topological embedding.

Overlay Given some embedding of elements in 2D, *overlays*, described by Vehlow et al. [VBW15] as "*Line/Region-based overlays*", add visual information atop elements/nodes to communicate set membership. This commonly takes the form of a *hulls/regions* overlaid atop grouped vertices [AMA+16], as seen in [PLS+13]'s *enRoute*. A key disadvantage of region-based overlays is the ambiguity that can arise from overlapping, but non-intersecting, contiguous regions [ARRC11]. *Line* overlays avoid such ambiguity by encoding group membership to line-node-intersections, as seen in *LineSets* [ARRC11]. Lines, however, can require a lot of "ink" when connecting vertices close to each other, resulting in more visual clutter than a simple region. In an effort to leverage the advantages of both, *hybrid* overlays combine both regions and lines, as exemplified by *KelpFusion* [MRS+13]. In general, overlays can be an effective means of communicating (disjoint) vertex-group membership, though they struggle to effectively communicate denser or more complex group-level relationships.

Bipartite Node-Link Diagram Thus far, group-level relationships have not affected the topological embedding, thereby communicating topological relationships clearly at the potential cost of group-level clarity. Here, *(bipartite) node-link diagrams* [AMA+16, VBW15] represent both groups and their elements as different vertices in the same embedding, and visualize their associations as bipartite edges connecting them. The produced node-link diagram thus has two types of edges: topological edges connecting element vertices to each other, and bipartite edges connecting group and element edges. This can allow group-level clustering to be more apparent as the group structure now directly affects the embeddings seen in *Origraph* [BNML19].

Multiple Instead of visualizing group information within, atop, or next to a graph's embedding, *multiples*, described as "*Partitioning*" by Vehlow et al. [VBW15], visualize each group's topology separately. Here, each such subgraph is displayed in its own "tile", arranged most commonly either in a juxtaposed or interchangeable manner. Juxtaposed multiples, such as *Small MultiPiles* [BHRD+15], allow for a straightforward side-by-side comparison of subgraphs. Interchangeable multiples require (animated) transitions from one tile to the next, as seen in *Graph Diaries* [BPF14a]. On the one hand, multiples allow for a clear visualization of each group's subgraph's topology and its element-group memberships. On the other hand, however, the visual communication of group-level

intersections in non-disjoint graphs is complicated, as vertices that map to multiple groups must be duplicated; once per tile.

Tree *Trees* are prevalent in the visualization of disjoint, hierarchical group structures. Elements are represented as the tree's leaves, and the various levels of the tree represent the various (hierarchically related) sets. Edges encode set-element membership as well as set-set hierarchies. Most commonly, trees are visualized in separate views, e.g., *ASK-Graph* [AvHK06]. as visualizing topological edges between the tree's leaf nodes can be visually difficult. For example, *CodeFlows* [TA08] addresses this challenge by visualizing topological relationships as a bipartite graph and duplicating the group's hierarchical tree structure along each bipartition.

Matrix *Matrices* [AMA+16] can communicate set membership and relationships tabularly in one of two ways. On the one hand, an identity matrix arranges elements and sets along its rows and columns, respectively, and "fills" a corresponding matrix cell if an element maps to that particular set. Intersections are communicated by a specific row (element) mapping to multiple columns (sets), e.g., Chuang et al.'s [CMH12] *Termite*. On the other hand, a similarity matrix places either sets or elements along both rows and columns and fills each cell with some measure of similarity, thereby communicating set or element relationships, but being unable to communicate element-set-membership on its own.

Abstraction *Abstraction*-based techniques, described by Alsallakh et al. [AMA+16] as "*Aggregation*", opt to visualize not all sets, elements, and their relationships, but instead provide an overview of one, at the expense of the others. Here, given the scope of this chapter, we define abstraction-based techniques as techniques that abstract away element information in favor of more clearly communicating group-level information. Quite naturally, this fairly broad category can encompass many different approaches; from the relatively simple *Linear Diagram* [RSC15], through the interactive *TugGraphs* [AMA11] or *Grouse(Flocks)* [AMA07a, AMA08], to the much more complex *Graph Thumbnail* [YDK+18].

Hybrid Finally, *hybrid* approaches [VBW15] combine the aforenoted representations in unique ways, exemplified by Henry et al.'s [HFM07] fusion of node-link diagrams and matrices, *NodeTrix*, or Angori et al.'s [ADM+19] combination of chord and node-link diagrams, *ChordLink*.

6.5.3 Group Structure

Lastly, continuing the work of Vehlow et al. [VBW15], we categorize collected techniques based on the type of group structure they are designed to visualize. While they focused on distinguishing not only *hierarchical* from *flat* and *disjoint* from *overlapping* group structures, these authors additionally noted *crisp* from *fuzzy* groupings. Given the scope

of our work, we omit the latter from our own categorization of literature. Thus, for each combination of visual encoding and visual relationship, we count the total number of approaches that apply to each considered type of data in order to identify potential gaps (Figure 6.4).

6.6 Lessons Learned

Following our taxonomy of visual encoding and visual relationships described in Sections 6.5.2 and 6.5.1, we classify the collected corpus accordingly, the results of which are presented in Figure 6.5.

Foregone Conclusions Somewhat unsurprisingly, many combinations of visual encodings pair (almost) exclusively with certain visual relationships (Figure 6.5). Most notably, overlays and node attributes, by definition, are embedded encodings, visualized within the graph's topological embedding. Thus, the 39 overlay and 22 node-attribute techniques map exclusively to the embedded relationship category. Similarly, multiples, if presented in 2D, would most commonly only be visualized either interchangeably or juxtaposed to each other, or separately from some other topological representation.

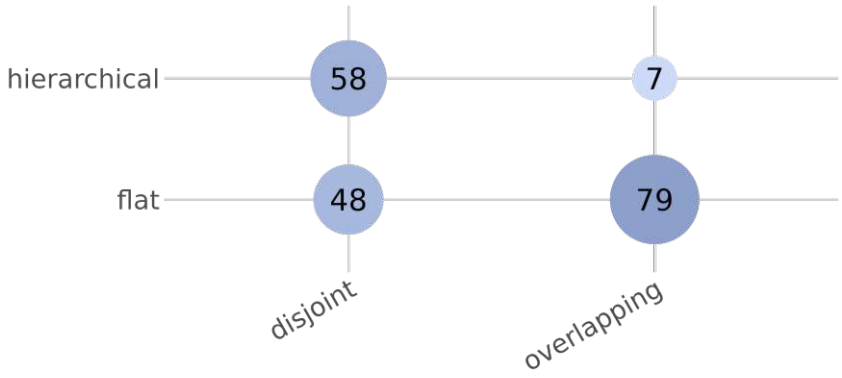


Figure 6.4: Group structures compatible with the visualization approaches collected from the selected papers. Circle area and color intensity encode the number of papers and techniques that map to a particular combination of group structure and "overlappedness". The exact number of papers is displayed at the center of each circle in black. Papers could map to more than one such combination of categories.

Embedded Relationships Of the 167 papers collected, the majority (98 references) featured an embedded relationship between graph topology and group structure (Figure 6.5). Most commonly, these papers represented their graphs' group structure using overlay, node-attribute, or hybrid techniques. Overlays and node attributes appear to be especially popular when the graph's topology is of greater importance than its group

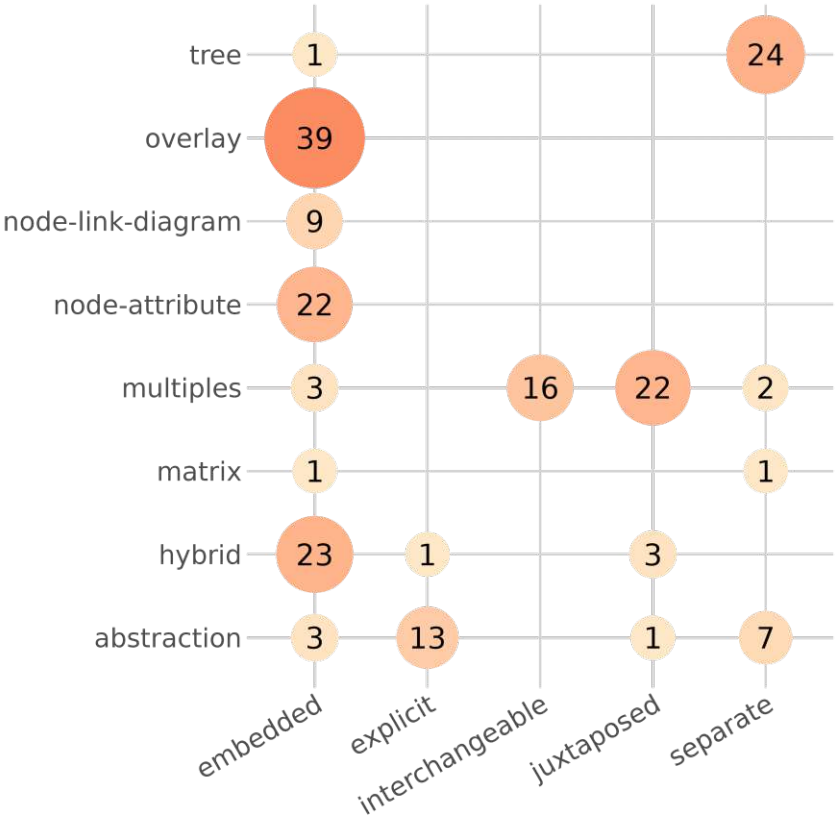


Figure 6.5: Co-occurrence of visual encodings and visual relationships of the selected papers. Circle area and color intensity encode the number of papers and techniques that map to a particular combination of encoding and relationship. The exact number of papers is displayed at the center of each circle in black. Some papers featured multiple visual encodings and/or relationships and were thus mapped to more than one combination of such categories.

structure, as they do not alter the topology representation. For completeness, however, it should be noted that several *region-based overlays* specifically, such as *Polyptychon* [DWS+14] and *H-BLOB* [SBG00], are algorithmically incorporated during the (spring-)embedding of the graph and thus do affect the graph's visualization. Their popularity seems to also be connected to their conceptual simplicity and ease of implementation.

Embedded relationships were used for all group structures, though primarily for *flat* ones. 37 papers were applied to *disjoint/flat* group structures, such as the region-based overlay *MapSets* [EHKP14] or the hybrid technique *ChordLink* [ADM+19], and 36 papers mapped to *overlapping/flat* group structures, as seen in hybrid tool *eXamine* [DEKB+14] or node-attribute approach [VHTW13].

This combination of *embedded* encoding and *flat* groupings is perhaps unsurprising, as the use of node attributes or overlays — the two most common embedded encoding techniques — do not straightforwardly allow for the encoding of hierarchical relationships on their own. Additionally, as discussed in Section 6.5.2, *embedded* approaches' ability to visually communicate overlap between more complex groups is also somewhat limited. Thus, most combinations of *embedded* encoding and *overlapping* groupings limit themselves to fairly simple cases, i.e., either few groups or few intersections.

However, *embedded* approaches were also utilized for hierarchical relationships, though primarily disjoint ones, with 24 instances of *disjoint/hierarchical* and 4 *overlapping/hierarchical*. *Disjoint/hierarchical* relationships mostly focus on visualizing a single layer of the hierarchy within the compound graph's embedding while visualizing the full hierarchy in a separate (linked) view, as seen in *OntoTrix* [BPL13], *HybridVis* [BPL13], or *TreeMatrix* [RMF12].

Lastly, for the few *overlapping-hierarchical embedded* visualizations, [WLL+16] and [NIST12] opted to visualize one layer at a time and communicate overlap using colored glyphs while the complete hierarchy was visualized separately, while [JKZ13] opted to visualize group-entity associations as a node-link diagram.

Trees *Tree*-style encodings were applied predominantly to *disjoint/hierarchical* group structures with 21 instances thereof, while all other group structures featured 2 instances each. Here, tree representations of group structure are most commonly visualized separately from the graph's topology: of these 21 *disjoint/hierarchical* group structures, 20 were visualized separately (Figure 6.5). This can be attributed to the visual complexity that such trees introduce on their own, i.e., a whole set of nodes and edges representing set relationships between groups in addition to topological ones between entities. By visualizing topology and group structure separately, equal weight can be given to both without one affecting the other, which allows for such complex, hierarchical relationships to be visualized more easily.

Indeed, looking at techniques intended for larger networks, such as *ASK-GraphView* [AvHK06] or Abello et al.'s [AKY05a] combination of fisheye views and treemaps, the graph's hierarchical group structure becomes complex enough to require a separate view. For simpler hierarchies and smaller graphs, a separate tree representation has been combined with abstracted graph embeddings, such as Archambault et al.'s [AMA07a] *Grouse* framework. Combining topological and group node-link diagrams in a single integrated view is possible as well [PVW06]. However, even in the examples given, understanding the graph's topology and group structure is challenging.

Mutliples Multiples are a reasonably popular approach for representing group structures, as they allow for each group's subgraph to be embedded (partially) independently of other groups' topologies. With 37 counts, they are primarily used for the visualization of *overlapping/flat* group structure, though they are also applied to 6 *disjoint/flat* groupings. Of these 37 *overlapping/flat* applications, 14 visualized these multiples *interchangeably*,

such as Erten et al.'s [EHK+04] *GraphAEL*, and 19 *juxtaposed* [FAM+11] (Figure 6.5). Especially for dynamic graphs, the use of interchangeable multiples, akin to the mapping of time to time [BBDW14], is a popular choice, with 12 of the 14 interchangeable papers using the visualization of dynamic graphs [MKF+15, RM14].

If the comparison of two or more graphs is of interest, the juxtaposition of subgraphs as multiples allows for a clear and uncluttered view of their differences, as seen in the works of Yoghourdjian et al. [YDK+18], Bach et al. [BHRD+15], and Behrisch et al. [BDF+14]. However, multiples can also be valuable in providing users with a small "thumbnail" representation of a subgraph to aid in exploration [AABS+14, BMGK08].

Lastly, beyond such 2D arrangements, 3 papers opted to present multiples in *embedded* relationships. Here, multiples are presented in 3D "cubes", formed by arranging them independently as seen in *MatrixCubes* [BPF14b], or arranging them in a juxtaposed manner as seen in *Caleydo* [LSKS10].

6.7 Outstanding Opportunities

Matrix Representations Adjacency matrices are used to visually communicate graph topology, which is expected given their advantages over common node-link diagrams [GFC04]. Surprisingly, matrices, despite being a straightforward and scalable method for the visualization of group structure, are hardly used (Figure 6.5). Though it should also be mentioned that some hybrid techniques, most notably *NodeTrix* [HFM07] and *Anchord+Matrix Diagrams* [MZ11], combine topological adjacency matrices and node-link diagrams to communicate topology and structure simultaneously. This has also been noted by Nobre et al. [NMSL19] within the context of multivariate graphs, Beck et al. [BBDW14] within the context of dynamic graphs, as well as Alsallakh et al. [AMA+16] within the context of set-typed data visualization. We also see ample opportunity to reap the benefits of (interactive) matrix visualization for the communication of compound graph group structures, as they are *simple to implement* and *understand*.

Node-Link Diagrams Somewhat surprisingly, *bipartite node-link diagrams* are infrequently used to communicate group structure, be it embedded or separately (Figure 6.5). Similar to trees (Section 6.6), it is possible that the additional complexity introduced by a second set of nodes for groups and edges, representing group memberships, simply makes them unsuitable for graphs with more complex topologies. Indeed, as seen in the works of Bigelow et al. [BNML19], Ahmed et al. [ABF+07], and Pienta et al. [PHE+18], the graphs studied are relatively small and simple. Moreover, color and/or shape are necessary to distinguish between topological and group nodes, further adding visual complexity. Nonetheless, this particular gap strikes us as worth investigating with scalable, and presumably interactive, bipartite node-link diagrams that combine topology and group structure in a single embedding. Specifically, for more clustered group structures, this could potentially allow for very clear visual distinctions between nodes that map to exclusively one group and those that map to multiple.

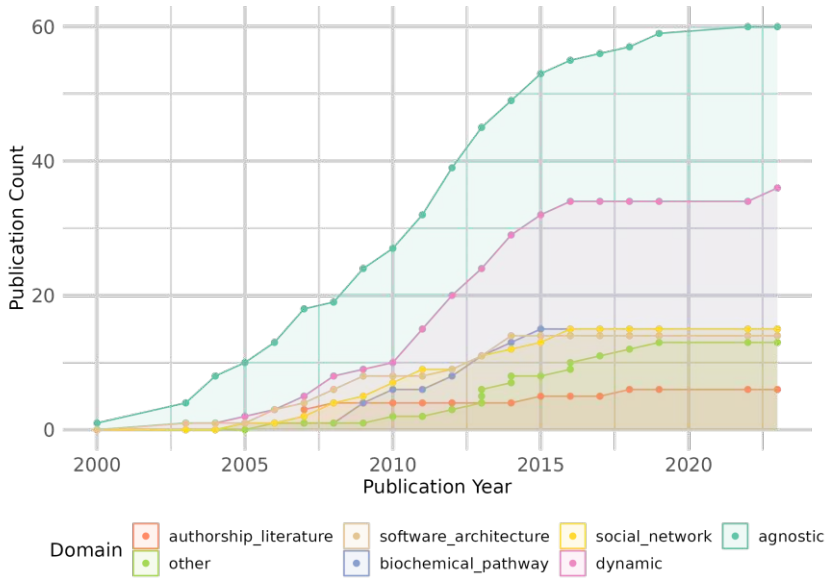


Figure 6.6: Cumulative number of the selected papers mapped to application area across time. Papers could map to multiple such application areas.

Represented Group Structures Mirroring the findings of Vehlow et al. [VBW15] *disjoint/flat* and *disjoint/hierarchical* are well represented with 48 and 58 papers, respectively, and the *overlapping/flat* category is the most represented with 79 entries. Interestingly, only 7 of the collected papers were applied to *overlapping/hierarchical* group structure (Figure 6.4). Most such papers opted to visualize only two or three group hierarchies at a time, usually separately, and link these to the graph's topology using either color [WLL+16] or edges [DWS+14]. Since overlapping, hierarchical groupings are common in ontologies and clusterings [VBW15], we see a great opportunity to tackle the challenges that such group structures present in the context of compound graphs, such as how to concurrently visualize different levels of the overlapping hierarchies, or how to best visualize each node's mapping to multiple categories of different hierarchies.

Hybrid Representations 16 hybrid visualization techniques (defined as any combination of visual encodings) were found in our paper set (Figure 6.5). Interestingly, however, all collected hybrid techniques combine specifically node-link diagrams with other visualization encodings, such as i) matrices [HFM07, MZ11, BPL13], ii) treemaps [BD05], iii) bounded node-link diagram embeddings [VHTW13, CDA+14], and iv) chord diagrams [ADM+22]. However, node-link diagrams are known to suffer from certain visual aesthetic limitations that can make them difficult to read when applied naively to larger and complex graphs, such as node/node occlusions, heterogeneous node density, edge crossings, or incident edge angles. Here, we see an opportunity for interesting and

novel hybrid visualization strategies with which to communicate the group structure of compound graphs that go beyond node-link diagrams, thereby potentially opening the door for more visually scalable approaches.

Underexplored Application Domains Certain application domains are notably more represented than others in the paper set collected (Figure 6.6). More specifically, domain-agnostic techniques and dynamic graphs appear to far outnumber all others. Application areas, such as authorship and citation networks, appear not to have been studied as extensively or recently. Often, as seen in biochemical application areas, a particular application area is constrained by the visual conventions of the domain and/or the (perceived) visual literacy of its users [LSKS10, PLS+13]. This, at least in our estimation, points to an opportunity to revisit such domains, to identify potentially unanswered challenges.

6.8 Conclusion & Future Work

We have surveyed literature based on both an independent collection of references and the existing works of Nobre et al. [NMSL19], McGee et al. [MGM+19], Beck et al. [BBDW14], and Vehlow et al. [VBW15]. The collected corpus of application and technique papers were then categorized within a comprehensive taxonomy that, inspired by the works of Kim et al. [KCK17] and Alsallakh et al. [AMA+16], disentangles the visual relationship between the graph's topology and group structure, as well as the chosen visual encoding of the graph's group structure, respectively. Based on this classification of literature, several lessons and outstanding research opportunities were identified:

1. the under-utilization of identity and similarity matrices,
2. the under-representation of bipartite node-link diagrams,
3. the under-studied visualization of *overlapping/hierarchical* data sources,
4. the over-representation of node-link diagram-based hybrid visualization approaches,
5. the understudied nature of certain application domains.

A future elaboration of this chapter in the shape of a formal state-of-the-art report should aim for a completely independent, large-scale collection of literature, and/or the collection of additional authoritative reviews to avoid any possible bias in the collected corpus of literature, to, in turn, draw potentially broader and more general conclusions.

CHAPTER 7

Putting it All Together: Penta

Throughout this dissertation, we have aimed to lay the foundations for several outstanding challenges that complicate the visualization of complex biological networks. Here, we aim to bring two key topics, the visualization of compound graphs (Chapter 6) and ego networks (Chapter 4), together by developing a novel prototypical dashboard for the visualization of compound graphs, including biological networks. Inspired by set visualization efforts, the dashboard *Penta* aims to decompose the compound graph visualization problem into separate but linked components. We ultimately demonstrate the utility of this dashboard using three case studies, including the visualization of biochemical *KEGG* pathway network data [KG00]. While still a prototype, *Penta* could form the basis of a more complete and fully-featured tool with which to visualize compound graphs. Future work should focus on improving visual scalability, incorporating more forms of interactivity, and allowing for the (visual) analysis of the kinds of multivariate networks common to biological usage scenarios.

The contents of this chapter are based on the short paper *"Penta: Towards Visualizing Compound Graphs as Set-Typed Data"* [EKR25], published in the conference proceedings of the *2025 VISIGRAPP* conference, with myself as first author, in collaboration with Mario Kerndler and Renata Raidou.

7.1 Overview

Compound graphs are graphs whose nodes, in addition to topological connections, share group-level relationships. The need to incorporate both topological and group-level relationships makes them inherently challenging to visualize, especially for large datasets. We present *Penta*, a prototypical dashboard that, by combining elements of compound graph and set visualization, provides a complete view of both types of relationships. To this end, we employ five linked views that provide insight into a compound graph's i) *global and set-local topology* using both hypernode and traditional node-link diagrams, respectively, ii) *set and entity-level relationship and identity* using similarity matrices linked by a bipartite node-link diagram, as well as iii) *node-centric topology* across sets visualized as a layered node-link diagram. We demonstrate the workflow and advantages of *Penta* in three small-scale case studies, using character co-occurrence networks as well as biochemical pathway data. While still a prototype, the proposed dashboard shows promise in facilitating a complete visual exploration of the topology and group-level relationships present in compound graphs, simultaneously.

7.2 Introduction

A clustered or compound graph is a graph whose nodes, in addition to topological connections, share group-level relationships [VBW15]. In the context of social networks, for example, nodes may represent individual people, edges (different types of) relationships between them, and groups, circles of friends. Given their general utility and applicability, compound graphs, their visualization, and their analysis are common across many different domains: from social sciences [HEAE16], through biochemistry [PF15] and neurology [AABS+14], to transportation logistics [Duc17]. Subsequently, various visualization approaches and systems have been put forth to tackle the visualization of compound graphs differently: Some of these forgo interactivity in the interest of scalability [XGH15], others combine the two using summarization techniques [PF15] or linked views [AABS+14], while others yet build upon domain-specific visual conventions to better serve a particular user group [HEAE16]. Ultimately, all these approaches address the challenge of simultaneously visualizing graph topology and group structure.

A compound graph's group structure can be conceptualized as a collection of sets; their elements formed by the graph's nodes, and their intersections by nodes present in two or more groups [PFH+18]. When visualizing or analyzing sets, we are interested in understanding their relations, such as containment, exclusions, or intersections, to better understand the role or importance of their elements [AMA+16]. However, even when the underlying data is not a graph, such set-typed data can become challenging to visualize. Depending on the complexity and size of the data, such visualizations often require the use of abstraction [RSC15] or summarization [LT20] to remain legible. Domain-specific applications may additionally require the interactive visualization of metadata attached to sets' and their elements [LG14]. Many different families of techniques and tools have been developed to tackle these challenges, both for generic set-typed and application-area-

specific data [AMA+16]. However, few — if any — of these techniques can be applied to compound graphs, and fewer still scale to large datasets, common in modern biochemical and social network analysis.

In this chapter, we present *Penta*, a prototype dashboard for the visualization of non-hierarchical, non-disjoint compound graphs as set-typed data. With this prototypical implementation, we aim to address several outstanding gaps in the fields of both set visualization [AMA+16] and compound graph visualization [VBW15, EMWR24], by combining multiple linked views, (similarity) matrix visualizations, and ego network representations. More specifically, the contribution of *Penta*'s linked views is that they provide insight into i) the group's global and local topology using both hypernode and traditional node-link diagrams, respectively, ii) set and vertex-centric relationship and identity using similarity matrices linked by a bipartite node-link diagram, as well as iii) node-centric topology across sets visualized as a layered node-link diagram.

7.3 Related Work

As *Penta*'s five linked views draw from conventions of compound graph visualizations, matrix-based set-visualization, as well as ego-network visualization, we here discuss the state-of-the-art of each.

Compound Graph Visualization Compound graphs have been visualized in a multitude of ways [EMWR24, VBW15], using node attributes, overlays, bipartite node-link diagrams, multiples, trees, matrices, abstractions, and hybrid techniques. The most common way of visualizing group-level memberships in compound graphs, node attributes take the form of, for example, colors [VBAW15] or glyphs [WLL+16]. Overlay techniques add group-level associations atop a graph's embedding, using, for example, regions [PLS+13], lines [ARRC11], or combinations of the two [MRS+13]. Less common than either overlays or node attributes, bipartite node-link diagrams represent both nodes and groups as (different types) of vertices with topological and group-level-association edges connecting them [BNML19]. Multiples opt to visualize each group's subgraph separately, as seen in Bach et al.'s *Small Multiples* [BHRD+15] or *Graph Diaries* [BPF14a]. Trees, commonly visualized alongside a compound graph's topology, visualize elements, i.e., vertices, as the leaves of the tree and their (hierarchical) set-element relationships as edges [AvHK06]. (Biadjacency) matrices, also visualized separately from a compound graph's topology, depict vertex-group memberships tabularly where each row corresponds to a vertex and each group to a column [CMH12]. Abstractions do not visualize all sets, elements, and their relationships but provide a view into one by abstracting away the others. This broad category can unsurprisingly include many different types of approaches [RSC15, YDK+18] Finally, hybrid techniques combine any of these aforementioned techniques to visualize group membership and topology simultaneously [ADM+19, HFM07].

Matrices for Set-Typed Data While several examples of set visualization and their elements as *biadjacency*-based matrices can be found in the literature [SMDS14, LGS+14], *similarity*-based matrix representations are less common. We could only identify two examples for the explicit purpose of similarity-matrix-based set visualization. First, Liu et. al's *Similarity Lattice* [XMB05] represents sets' pairwise *Completeness* and *Intensity* similarity as a non-symmetric similarity matrix. Second, *Intervene* [KM17] features similarity-based matrix representations alongside more conventional biadjacency-based ones. As also discussed in reviews of both compound graph visualization [VBW15, EMWR24] and set visualization [AMA+16], the use of (similarity) matrices is a novel avenue with which to communicate group and entity-level relationships in both sets and compound graphs.

Ego Network Visualizations Ego network representations visualize graph topology relative to some selected node of interest—the so-called "*ego*" [EPF+24]. Ego networks may be used to simplify the visualization of a graph's topology by only visualizing those nodes and edges of immediate importance to the selected ego. Examples of node-centric visualizations of larger graphs can be found across several domains [EPF+24]. Here, we are particularly interested in the use of layered (tree-like) node-link diagrammatic representations, as they promise an intuitive and orderly view into the intrinsically layered topology of ego networks. Such layered representations align nodes along "layers" representing the distance from the selected ego [Say04]. While not related to ego networks directly, similar ideas of edge scaling and distance-based node embedding can be found in the visualization of phylogenetic trees [SWKP18], evolutionary graphs [MMTM11], or synaptic/brain graphs [AABS+14]. To the best of our knowledge, such ego network visualizations have not yet found application in compound graph visualization.

7.4 The Five Facets of Penta

Visualizing compound graphs is challenging, as one needs to visually communicate both graph topology and group-level relationships simultaneously. Specifically, supporting the following aspects is necessary:

1. viewing all groups' **global topology**,
2. investigating the particular **set-relative topology** of selected groups,
3. understanding the **similarities of all groups and nodes** relative to each other,
4. viewing all selected **set-element mappings**, and
5. investigating the **node-relative topology** of the graph, i.e., topology relative to a selected ego.

To integrate the aforementioned channels of information, we propose the prototypical web-based *Penta* dashboard, which aims to visualize different facets of a compound graph's group structure and topology in five linked views. Users are expected to upload their compound graph's topology and group-level associations they wish to investigate to

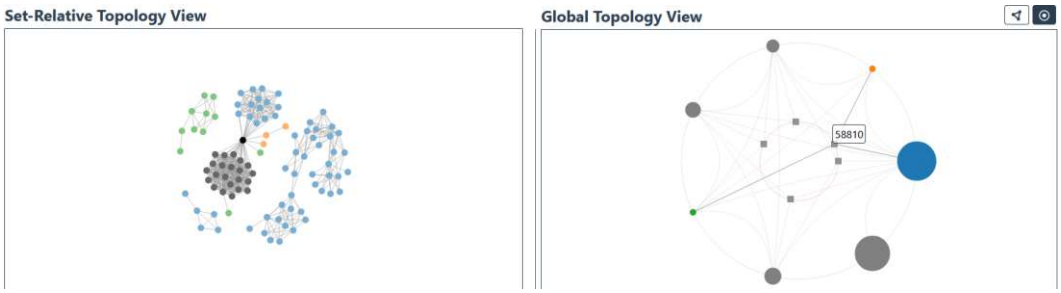
Penta, within which they then interactively filter and explore their dataset. This may entail selecting entities or sets within its multiple linked views or looking them up based on their unique identifiers. More specifically, in accordance with Shneiderman's [Shn96] mantra of "*Overview First,* [...] *filter, then details on demand*", the dashboard aims to funnel users from a *global view of topology*, through the *topology* and *similarity of selected sets*, to the immediate *topology of a particular node*. Interactive selections of vertices and sets in any of these five views are reflected in all other views through brushing and linking [BMMS91]. *Penta*, available on GitHub, was developed in `Svelte` using `D3.js` [BOH11].

Global Topology For some compound graph $G(V, E, S)$, let V denote its total set of elements/nodes, E its total set of edges, and S its elements' non-hierarchical, non-disjoint group structure. For a set $s \in S$, node-set pairs (v_a, s) and (v_b, s) in V_G are connected by edges $E_G \subseteq V_G \times V_G$, where $V_G \subseteq V \times S$. Here, the input graph's global topology can be either represented as a force-directed or (as presented here) a radial node-link diagram (Figure 7.1b). As straight-line node-link diagrams do not tend to produce aesthetically pleasing or readable results for graphs of larger sizes and greater complexity [EVRW23], we do not visualize nodes $v \in V$ and their edges $e \in E$ directly. Instead, each set of nodes V_G is visualized as its own circular hypernode [TBNV10]. Hovering over such hypernodes reveals their labels. To communicate which groups of nodes are present across sets, groups of nodes that form intersections between sets of nodes are represented by separate, square hypernodes (Figure 7.1b). Hypernodes' sizes encode the cardinality of their sets or intersections, i.e., the more elements a set contains, the larger the set's corresponding hypernode.

In the particular example shown in Figure 7.1b, seven groups, representing seven *Mus musculus* KEGG pathways, are depicted—one circular node per group. The size of each node represents the number of genes (nodes) in each pathway. Between these seven groups, there are five unique intersections, represented by five square nodes. One particular intersection, consisting only of a single node with identifier *58810*, is highlighted. The said node is the intersection between the three selected, i.e., colored, groups, indicated by the edges connecting the intersection's square node to the three groups' circular nodes.

Set-Relative Topology From the aforementioned global topological view, hypernodes can be selected, thereby adding their elements, i.e., vertices, to the set-relative topological view. This is represented as a straight-line node-link diagram (Figure 7.1a), laid out using `D3.js`'s particle-based force-directed algorithm [BOH11]. Each added group of nodes is color-coded, using an appropriate *ColorBrewer* color palette [HB03] used consistently across all views. To ensure colors remain visually distinct from each other [Hea96], users can only select up to ten such groups simultaneously. We highlight intersections between two or more groups by color-coding them in black. Hovering over any individual node reveals its label and highlights its corresponding hypernode in the global topological view, as well as its location in the (to-be-discussed) element-similarity matrix in a similar way.

In the example shown in Figure 7.1a, three of the seven aforementioned *Mus musculus* KEGG pathways have been selected, one colored green, the other blue, and the final one orange. Nodes that form an intersection between any two of the three selected pathways are colored dark grey. The selected node *58810*, which forms an intersection between all three selected pathways, is colored black.



(a) (b)

Figure 7.1: The **(a) set-relative topology** of a selection of three sets from the global topological view, colored in blue, orange, and green in both views. The **(b) global topology view** for seven KEGG mouse (*MMU*) pathways (groups) and their intersections. A notable intersection between these three selected is the entity (vertex) with identifier *58810* which is highlighted in black on the (a).

Set and Vertex Similarity To navigate both sets and elements, we propose the use of two interactive similarity matrices—one depicting *set similarity* and the other representing *vertex similarity* (Figure 7.2c). Unlike identity matrix representations, which may necessitate panning and scrolling to navigate owing to potential asymmetries, a square, symmetric similarity matrix offers a clearer overview of all sets or elements simultaneously. Moreover, matrix representations offer overall greater visual clarity and flexibility over many other approaches for set and element visualization [AMA+16]. This conceptual flexibility includes the **straightforward use of different similarity measures** with which to understand groups and elements relationships [VK16]. Here, for any two sets $A, B \in S$, their similarities are computed using the Jaccard distance $J(A,B) = (|A \cap B|)/(|A \cup B|)$, chosen for its conceptual simplicity and its applicability to both set and topological node similarity. To quantify element similarity, the same conceptual approach based on the Jaccard distance is used: for any two vertices/elements $a, b \in V$, we can define their "sets" as $A = q(a)$ and $B = q(b)$, where $q(x)$, for an element $x \in V$, can take different forms. We make use of a topological set definition: $q(x) = \{v : \forall v \in V, \{x, v\} \in E\}$, i.e., we simply define a node's set as its immediate neighborhood. In both cases, the similarity is encoded using a luminance colormap, i.e., white indicating low and black high similarity. Moreover, a selected set's rows and columns are indicated using colored lines. Additionally, **matrices can be flexibly reordered** based on selected criteria to facilitate the detection of novel or unexpected relationships between entities [MML07]. Here, as a proof of concept, we allow the user to

reorder both set and element similarity matrices by either i) a random ordering for the identification of unexpected relationships, or ii) an attribute-based ordering that organizes entities by their lexicographic order, or iii) an agglomerative, hierarchical clustering-based order to identify clusters of interest [KR90].

In the example shown in Figure 7.2c, the set-similarity matrix represents the character similarities among the first four books of the Harry Potter series [Rav24]. Each set, i.e., book, is represented by one of the four rows and columns of the matrix, ordered lexicographically. Each cell represents the Jaccard distance between two sets. The darker a cell, the more similar the two sets are. Instead of space-consuming conventional labels, colored lines indicate the identity of each row and column. Here, books two and three have been selected from the global topological view (Figure 7.2a). The vertex-similarity matrix (Figure 7.2c) shows the similarity between nodes (characters) as a function of their immediate neighborhood, i.e., the more neighbors two nodes have in common, the more similar they are. Each row and column corresponds to a particular character, and each cell to the similarity between two nodes. Again, similarity is mapped to cell brightness.

Set-Node Connection Graph The previously discussed similarity matrix representations allow for an in-depth exploration of within-set and within-element relationships. However, they do not visualize the set membership of elements or the mapping of sets to elements [PFH+18]. To link the aforementioned similarity views, we visually exploit the bipartite relationship between them and visualize said relationship as a "*Connection Graph*" (Figure 7.2c). For matrices, drawing (curved) edges between their rows (or columns) is conceptually fairly straightforward, as they already form a one-dimensional projection of their elements [MZ11]. More specifically, given a single selected set S, its projection point is connected to all its elements' ($x \in S$) projection points by curved edges. Alternatively, given a single selected element x, its projection point is connected to all sets of which it is a member, i.e., $\{s : \forall s \in S, x \in s\}$. Edge and label colors utilize the same color palette as in the global and set topology views.

In this particular example (Figure 7.2c), the connection graph shows the mapping of Harry Potter books (sets) to the characters of the series (nodes), and vice-versa. For each book's character, curved edges are drawn between the book's and the character's row in their respective similarity matrices. The color of the curved line corresponds to the color of the set, as depicted in both the global and set-relative topological views (Figures 7.2a and 7.2b). If a node, i.e., character, maps to multiple sets (books), multiple such lines will be drawn, connecting the one node to multiple sets.

Node Relative Topology While the global and set-local topological views of the graph provide a good starting point for exploring particular sets and their elements, they are insufficient in understanding a node's connectivity across all sets. To amend this, we employ a vertex-centric representation of topology for a particular selected entity, i.e., an ego network where the selected vertex forms the "*ego*" [EPF+24]. This ego network is generated from the graph's global topology based on a selected ego node

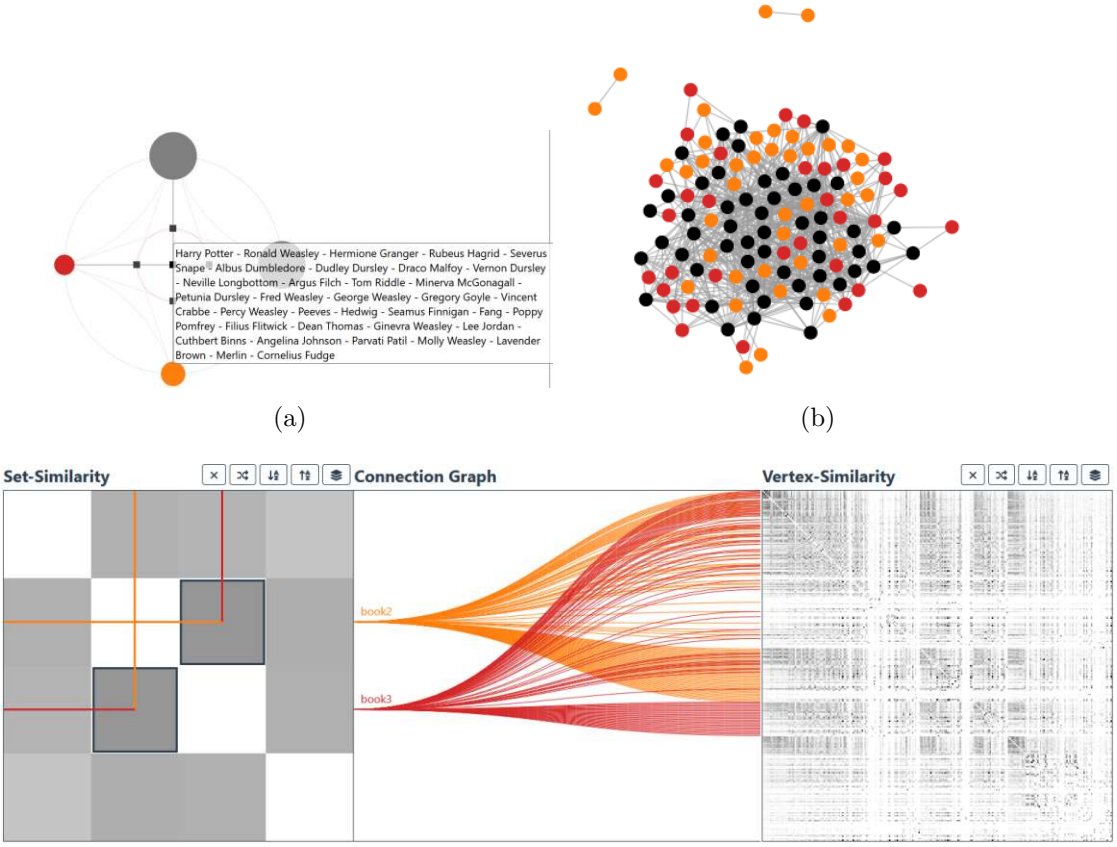


Figure 7.2: Visualizing Character Interactions Across *Harry Potter* Books. **(a) Global Topology View:** Overview of character interactions across the first four books, highlighting recurring characters. **(b) Local Topology View:** Dense inter-character connections between books two and three. **(c) Set and Vertex-Similarity Analysis:** Consistent level of recurring characters and interactions between books.

of interest and a user-specified maximum exploration depth. More specifically, *Penta* employs a breadth-first search algorithm to traverse the graph from the ego node, through its neighbors, its neighbors' neighbors, and so on, up to the specified depth, thereby constructing a tree(-like) data structure that encapsulates the shortest possible paths to all reachable nodes. This tree is visualized as a layered node-link diagram (Figure 7.3c). Its layers, i.e., nodes, are sorted by their inverse weighted edge distance to the ego, i.e., the greater the edge weight between any two nodes, the closer they will be visualized in the tree. This also allows for *Penta* to sort a node's neighbors based on this edge weight, i.e., the closest neighbor appears at the top of its branch, and the furthest at the bottom.

In the example of Figure 7.3, a co-occurrence network of *Star Wars* characters across the first six movies is shown. The depicted ego network displays the node-relative topology of one selected character, namely *R2-D2*, depicted in the top left of Figure 7.3c. The layered node-link diagram shows a character's connections across all sets in the dataset. The more a character interacted with *R2-D2*, i.e., the larger their co-occurrence edge weight, the higher up in the layered node-link diagram it appears, i.e., *Anakin* interacted more with *R2-D2* than *Qui-Gon*, who interacted more than *Luke*. The same ordering system applies to each layer of this layered node-link diagram. The ordering is reinforced by scaling each edge's horizontal length based on edge weight, i.e the more two characters interact, the larger their edge weight, and the closer they appear in the horizontal space.

7.5 Use Cases

In order to demonstrate the potential utility of *Penta*, three small-scale usage scenarios are presented. These three usage scenarios are demonstrated using three different datasets: a KEGG *Mus musculus* example dataset [KG00], a Harry Potter character co-occurrence network [Rav24], and a Star Wars character co-occurrence network [Gab24]. These datasets were selected, as they are publicly available, and, in the case of the Harry Potter and Star Wars data, hopefully familiar to many readers.

Scenario 1: Noticing Notable Nodes In order to demonstrate the utility of both the global and set-local topological view, consider a *Mus musculus* example dataset, in which each node represents a gene, each edge a gene-gene interaction, and groups of genes represent KEGG pathways in mice [KG00]. For seven such pathways, their global topology is visualized as a hypernode-link diagram (Figure 7.1b). Here, three pathways of particular interest, *mmu0010*, *mmu00053* and *mmu00040*, colored in blue, green, and orange, respectively, are selected in the global topological view, their vertices visualized in the corresponding set-local topology view (Figure 7.1a). From the global topology view, we can immediately discern the relative sizes of these three sets, i.e., *mmu0010* (blue) is larger than any other pathway in the dataset, indicated by the area encoding of the nodes. Moreover, we can immediately see how all our datasets intersect and the cardinalities of these intersections. More specifically, a particular intersection between the three pathways of interest is revealed to be a single node, namely *58810*. The importance of this particular node across the selected pathways is also made immediately apparent in the set-relative local topological view (Figure 7.1a), where said node not only forms an intersection between these three pathways but also a bridge between two large clusters of nodes. Among those, one forms an intersection of its own between two selected pathways (colored in grey). The combined use of these two views—hypernode and traditional node-link diagrams—has enabled an intuitive and speedy overview of the pathway's global and local topology and facilitated the identification of a notable node.

Scenario 2: Such Similarity, Much Wow In order to demonstrate the utility of set and node-similarity matrix views, consider a Harry Potter character co-occurrence

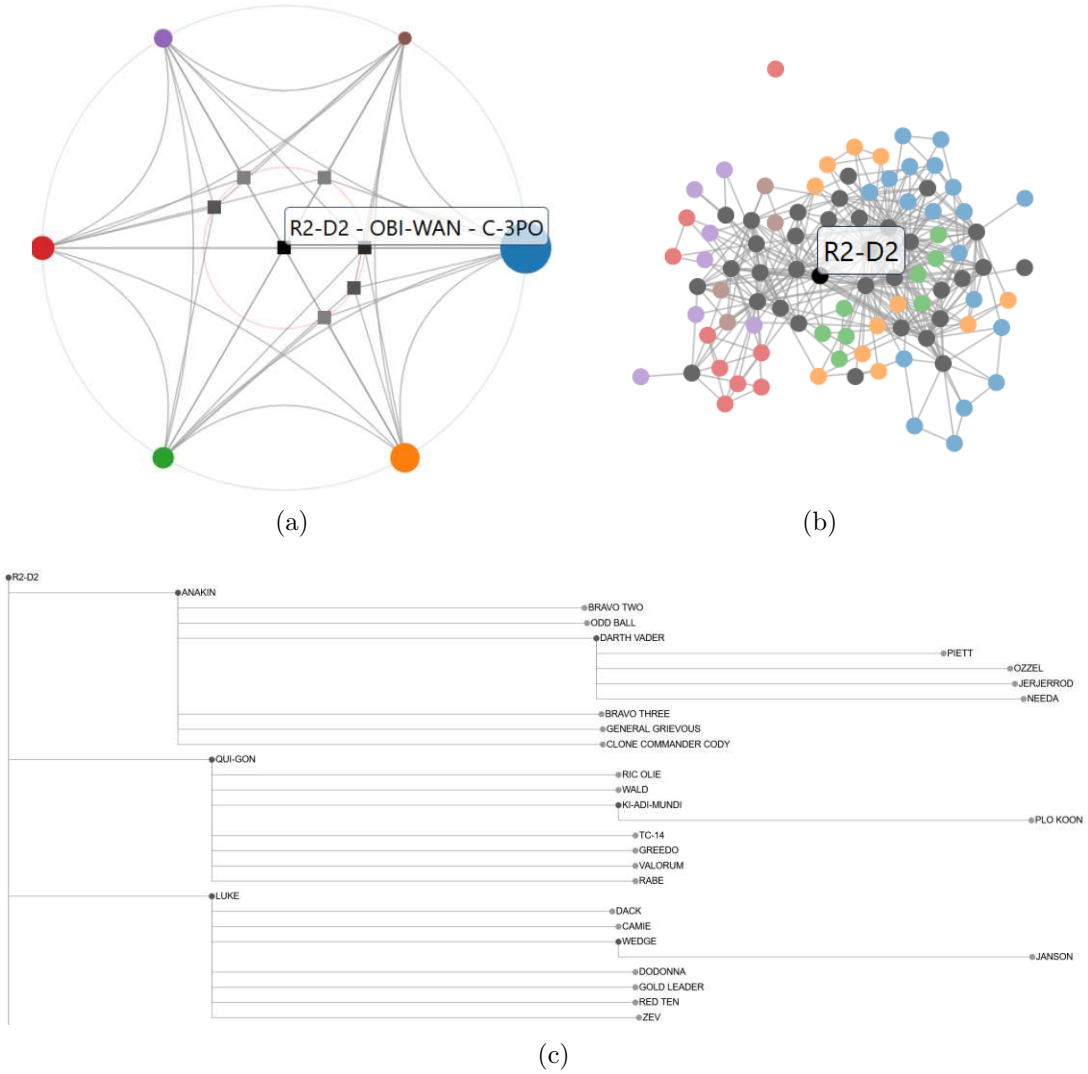


Figure 7.3: Analyzing the *Star Wars* Character Network. **(a) Global Topology View:** Comprehensive visual map of the Star Wars character network, highlighting key characters across all movies. **(b) Local Topology View:** View into the dense interrelationships of characters across the first six movies. **(c) Ego-Network of R2-D2:** Central role and extensive connections of R2-D2 within the Star Wars narrative.

network [Rav24], in which each node represents a character, each weighted edge the number of interactions two characters shared, and each group one of the first four books of the series. In order to determine the similarities between books two and three, colored orange and red respectively, the two groups' hypernodes have been selected from the global topological view (Figure 7.2a) and their nodes and connections added to the local topological view (Figure 7.2b), as discussed previously. This view already sheds

some light on the strong overlap in characters between the two books, given the many (black) nodes that form intersections between the two books. However, viewing the set-similarity matrix (Figure 7.2c), the darker gray color of books two and three's similarity immediately communicates that these two books share the greatest set of characters across the first four published books. Following the connection graph and viewing the vertex similarity matrix, one can immediately note three interesting blocks of characters, namely i) those that strongly co-occur across books two and three, ii) those that primarily co-occur in book two, and iii) those that primarily co-occur in book three Figure 7.2c). The interactive dashboard now allows for individual inspection of character-character similarity scores. The set and vertex similarity matrices allow us to quickly identify subsets of nodes of potential interest for follow-up study.

Scenario 3: Me! Me! Me! Me! Here, to explore the utility of the ego network, consider a co-occurrence network of Star Wars characters across the first six movies, in which each node represents a character, each weighted edge represents the number of interactions two characters shared, and each group represents one of the six movies [Gab24]. Here, given some prior exploration of groups and their connectivity, a user has identified a particular node of interest, namely *R2-D2*, as this particular character forms an intersection across all six groups (Figure 7.3a). As the local topological view (Figure 7.3b) is too dense to make out said character's immediate neighborhood, a user can utilize the vertex-relative ego network view (Figure 7.3c. Here, the selected ego, here *R2-D2*, forms the root node of the layered tree representation in the top left. From a simple cursory look, a user can immediately identify the ego's closest neighbors, here *Anakin*, *Qui-Gon*, and *Luke*, thanks to the vertical ordering of the ego's neighbors by edge weight, i.e., *R2-D2*'s closest neighbors are located at the top of the tree. Moreover, thanks to the horizontal ordering of nodes based on cumulative proximity to the ego, one can additionally notice that, while *Piett* and *Ozzel* are both a hop-distance of three away from *R2-D2*, *Piett* is closer once edge weights are factored in. This node-relative topological view facilitated by the ego network allows for an in-depth look at the selected ego's (immediate) neighborhood in an intuitive and fast manner.

7.6 Summary and Take-away

The prototypical *Penta* dashboard offers a holistic approach to visualizing compound graph data, enabling users to analyze both topological and group-level relationships across multiple datasets. Through our three case studies, *Penta* demonstrates its ability to highlight both global and local graph structures effectively. For example, in the first scenario, *Penta* allows users to quickly identify key intersections and pathways, such as important genes within biological networks. The second scenario showcases *Penta*'s utility in visualizing co-occurrence networks, revealing shared character groups across Harry Potter books. Lastly, the third scenario focuses on individual nodes within a Star Wars network, where users can explore character relationships based on interaction frequency.

Overall, *Penta*'s prototypical implementation already promises to be a valuable tool for researchers looking to interpret compound graph relationships in larger datasets.

7.7 Conclusion and Future Work

In this chapter, we have presented the prototypical implementation of *Penta*, a novel approach to the holistic visualization of compound graph data. We further demonstrated the use and effectiveness of our tool on three use case scenarios, which indicate *Penta*'s utility in making sense of a compound graph's topology, both on a local and global level, as well as group structure. In future work, we hope to i) implement additional similarity measures with which to view set and vertex relationships[VK16], ii) implement additional clustering algorithms with which to sort similarity matrices [KKPEP20], iii) allow for the integration of node and edge attribute data to go beyond purely topological analyses, and iv) improve the scalability of multiple views through, for example, latent variable space scatter plots [PFH+18].

CHAPTER 8

Discussion and Conclusion

In Section 2, we identified several outstanding challenges and opportunities within the context of biological network visualization. We, however, opted to focus on three key questions, which we deemed particularly interesting:

Research Question 1 How to improve the visualization of the kinds of complex networks common in biological network endeavors?

Research Question 2 What kind of uncertainty is prevalent in biological networks, and how should said uncertainty be best visualized?

Research Question 3 How to effectively visualize the kind of compound graph group structure common to biological network visualization?

In the previous five chapters of this dissertation (Sections 3 - 7), we characterized, contextualized, and tackled these three research questions. In this section, we finally aim to provide i) a summary of how, as well as how successfully, these research questions were tackled, ii) an overview of concrete outstanding work with which to better position our contributions in the field of biological network visualization, iii) a discussion on broader future research directions within the context of biological network visualization, as well as, finally, iv) a few concluding remarks with which to tie the contributions of this dissertation together.

8.1 Summary

Here, we aim to provide a small summary of how and how well we tackled the three identified research questions across the corresponding five chapters of this dissertation.

8.1.1 Research Question 1: The Hairball Problem

Conventionally available heuristic (stress and force-directed) graph drawing algorithms for straight-line node-link diagram visualizations, commonly relied upon in biological network visualization efforts (Section 2), do not scale well visually with increasingly large and dense networks, producing less and less readable "hairballs". In the introduction to this dissertation (Section 1), we identified three conceptual solutions to this broad challenge.

Improving Graph Drawing Algorithms First, one possible solution hereto would be the development of novel graph drawing algorithms that, for example, embrace more constrained, and thereby possibly more legible, node layouts, such as schematic ortholinear/octolinear, radial, or layered layouts. Alternatively, such novel layout algorithms could aim to optimize carefully selected and weighted graph readability criteria directly. However, this lies more so in the realm of algorithm development and graph drawing.

Cleaning up the Drawing Second, one could build upon existing graph drawing algorithms by improving the network visualizations they produce, i.e., improving network visualization *after* they have been laid out algorithmically. It is this second approach we embraced when proposing a principled approach to vertex splitting for iterative edge crossing resolution (Chapter 3). Several conceptual approaches have been (graph theoretically) investigated, such as node/edge deletions or partial edge drawings.

We investigate vertex splitting, i.e., the duplication and repositioning of a graph aesthetically "problematic" vertex, followed by the redistribution of its incident edges across the thus created copies, thereby improving certain readability criteria in the presented graph drawing. Vertex splitting already finds some, though limited and simple, use in practice, both in social and biological networks, highlighting its conceptual usefulness. We opted to specifically focus on edge crossing resolution as they are one of the most important quantitative graph aesthetic metrics to optimize, both in small and (to a lesser extent) larger graphs. To resolve these crossings, thereby improving the drawing's overall (graph aesthetic) readability, we proposed a novel algorithmic approach (with local guarantees) to vertex selection, split vertex location selection, and incident edge redistribution. Care must, however, be taken when splitting vertices, as we hypothesize the number of split nodes may itself be a readability criterion: the more of them there are, the more difficult the split graph drawing becomes to read.

Thus, we subsequently studied the impact of vertex splitting on small graphs in a crowdsourced user study. The results hereof indicate that vertex splitting does not hinder user performance or affect user preference, at least for the smaller graphs presented

during our user study. Vertex splitting did, however, increase study participants' task completion times, indicating, to us, that either split graph drawings i) require additional/deeper user training, or ii) indeed make certain tasks, such as path tracing or adjacency determination, more demanding. We note that, for biological networks, the studied smaller and intermediate-sized graph visualizations remain relevant within the context of, for example, knowledge subgraph visualization, e.g., *KEGG* or *Reactome* pathways. This, to us, indicates that vertex splitting, if developed and studied further, could be a valuable approach in improving the readability of such small and intermediate biological networks. However, as many domains struggle with dense and complex graph drawings of poor readability, we argue that vertex splitting is a conceptually useful approach across application domains.

Sidestepping the Problem A third approach to solving hairballs still presents itself: instead of attempting to make sense of an entire network at once, *Focus+Context* techniques could simplify the visualization problem by focusing on smaller, and more manageable, sub-parts. Here, we study ego networks (Chapter 4) as a (within the context of biological networks) novel *Focus* technique.

Ego networks opt to restrict the visualization of a larger network to only those nodes and edges that are (up to a certain search depth) adjacent and incident, respectively, to some selected node of interest. Given the little systematic work done on characterizing ego networks, we first survey the current state-of-the-art of ego network visualization across domains. From this literature review, we identify the four most common approaches, i.e., straight-line node-link diagrams, radial node-link diagrams, layered node-link diagrams, and (centered) adjacency matrices, in order to compare them in a subsequent large-scale, crowd-sourced, mixed methods user study.

Our results indicate that adjacency matrices, despite their conceptual advantages, require substantially more initial effort and training to use effectively, as they systematically underperformed compared to the three studied node-link diagrams. Between the three node-link diagrammatic representations, little quantitative or qualitative differences could be detected. Some comments left by users do hint at the potential superiority of layered node-link diagrams, specifically, owing to the ease with which participants were able to learn and use them. However, given the wide variety of ego network sizes and densities, different approaches may prove more or less useful, depending on the particular application: While adjacency matrices proved themselves worse for the "*Les Misérables*" undirected graph data presented ($|V| = 77$ and $|E| = 254$), a larger and/or denser graph could have rendered the node-link diagrams comparatively ineffective. Given the heterogeneity of biological networks, we note the need for future studies investigating more diverse network types. Nonetheless, given their frequent use in the social sciences, we argue that, if implemented within an interactive system for biological network visualization, ego networks could form the *Focus* component of novel *Focus+Context* techniques.

8.1.2 Research Question 2: Incorporating Uncertainty

Uncertainty is present in almost all data throughout the visualization pipeline, regardless of the domain. Subsequently, it is unsurprising that biological networks also "suffer" from uncertainty in their visualization efforts, be it during data collection, data cleaning, data analysis, or rendering. For network visualizations, in particular, uncertainty can express itself in three key ways: i) probabilistic graph uncertainty, i.e., the probabilistic presence and absence of edges, ii) layout uncertainty, i.e., the uncertainty induced by graph drawing itself, and finally, iii) node and edge attribute uncertainty. It is this last category that we focused on when tackling uncertainty visualization in networks (Chapter 5), owing to their prevalence in and importance to biological network visualization efforts: Many biological and biochemical networks are actually multivariate networks, comprised of experimental data attached to biological entities, as well as knowledge-graph-derived topology describing the relationships between these entities. This experimental data, be it the differences in gene expression values between two clinical conditions or average protein abundances in a sample, brings with it uncertainty and variance often ignored in the final visualization of these multivariate networks.

Here, we aim to study different approaches to visualizing such node attribute uncertainties in commonly employed straight-line node-link diagrams. Given the lack of systematic work done on such uncertainty visualization in networks, we first review, in a large-scale survey of literature, existing approaches to uncertainty visualization outside of the context of network visualization. From this survey, we select three commonly utilized, intuitive, and powerful approaches, upon which we base our implementation of three uncertainty encodings: node enclosure, node fuzziness, and node saturation.

We compare these three chosen encodings to a novel, fourth encoding, i.e., animated node wiggliness, in a larger-scale, crowd-sourced, mixed methods user study. Ultimately, despite its controversial standing in the literature, animation proved to be just as useful as its three competitors. Specifically, we found no (statistically) notable quantitative or qualitative evidence suggesting the inferiority of animated wiggliness. This, in our estimation, is a solid first step in showcasing the possible utility of animated wiggliness within the context of graphs' node attribute uncertainty visualization. For uncertain multivariate biological network visualization efforts, we note the need for follow-up studies in larger and denser networks. Such studies may highlight perceptual issues, i.e., too many moving nodes to keep track of, or limitations of the method, i.e., node occlusions induced by wiggliness given a denser graph. Moreover, we stress the need to investigate more complex (domain-specific) tasks, such as path-tracing or cluster identification in highly uncertain settings. Nonetheless, we argue that wiggliness shows promise for node attribute uncertainty visualization in biological networks and beyond, by i) effectively and intuitively drawing attention to uncertain nodes in the network visualization, and ii) visualizing uncertainty without necessitating the use of visual channels commonly reserved across various biological domains, such as node color or size.

8.1.3 Research Question 3: Visualizing Compound Graphs

Multivariate graphs, i.e., graphs whose nodes and edges carry (meta)data in addition to their topological adjacency and incidence, are common across domains, including biological networks. One common attribute is group structure. These so-called compound graphs are comprised of nodes and edges that share (hierarchical) set relationships in addition to their topological ones. In biological networks, such group structures are highly common and complicate their visualization. Consider, for example, so-called *Single Nucleotide Polymorphisms* (SNPs) which fall within genes, which fall within haplotypes (or vice versa), which fall within chromosomes. Alternatively, consider genes which fall within pathways, which themselves are often grouped in a hierarchical ontology of function. Here, although some systematic work had been done to characterize compound, dynamic, multilevel, and multivariate graphs, these efforts were, at the time of writing, several years old. Moreover, despite their conceptual similarities, these graphs are often studied separately.

Thus, we set out to unify the bibliographies of these past surveys and reviews as well as add to them. Additionally, we propose a novel taxonomy that distinguishes explicitly between i) the visual relationship between graph topology and group structure, as well as i i) the visual encoding chosen to visualize the group structure itself (Chapter 6). From this survey and taxonomic classification of literature, we are able to identify several outstanding challenges.

Inspired by the findings of this survey, we put forward the prototypical dashboard *Penta* (Chapter 7), which aims to tackle (non-hierarchical) compound graph visualization as a set visualization problem. Instead of visualizing graph topology and group-level relationships in a single complex view, *Penta* decomposes the compound graph visualization problem into five separate, linked views, i.e., global set topology, set-local topology, set similarity, entity similarity, and node-relative topology. We argue that this decomposition ensures that each of these five individual views into the graph is as visually clear and uncluttered as possible. We demonstrate the utility of *Penta* in three small-scale use cases, including the visualization of biological *KEGG* pathway data. Here, we showcase the advantage of the global set topological view as it immediately highlights important intersections between sets, i.e., nodes that map to multiple groups. Through a subsequent investigation of the linked set-local topological view, one of these intersections is revealed to also be of topological importance, as it forms a bridge between two large groups of nodes. Ultimately, while still a prototype, we argue that the conceptual approach of *Penta*, if extended and refined, could be highly useful to biological domain experts.

8.2 Outstanding Work

Hands-Off Vertex Splitting In Chapter 3, we lay the foundations for a systematic approach to iterative edge crossing resolution through vertex splitting. However, several challenges were not tackled in said chapter, which ultimately limits the application of vertex splitting in its current state, especially to larger and more complex (biological)

networks. Here, we identify several interesting outstanding research challenges underlying algorithmic vertex splitting in (straight-line) node-link diagrams. First, we investigated only 2-split vertex splitting, i.e., an algorithm that only ever creates two duplicates of a selected vertex. Here, we see an opportunity to investigate **generalized k-split vertex splitting**, i.e., a splitting algorithm that is able to split a selected vertex k times, should such a split better optimize the graph drawing's overall readability criteria. Such a generalized approach might be able to improve the readability of a given graph drawing more efficiently and effectively. Second, we note the lack of **human-readability-based stopping criteria and guidelines** for splitting: *how many vertices can be split, and how many split copies can be present before a graph drawing becomes unintelligible to a human user*? In graph theoretical circles, complete planarization, i.e., the removal of all edge crossings, is often the goal. While graph theoretically interesting, such a graph drawing is not necessarily optimized for actual human use. Thus, we see the need for multiple large-scale and in-depth user studies to develop guidelines and stopping criteria for vertex splitting to be actually useful to users. Third, across the limited work done on vertex splitting, efforts have always concentrated on resolving edge crossings. While edge crossings are indeed a crucial graph aesthetic criterion, we argue that **optimizing multiple criteria** may yield overall more readable graphs. Careful consideration and study would need to be given to the selection of certain graph aesthetics over others, and their relative weights in the optimization. This may require further study into the relative importance of graph aesthetic criteria in general. Fourth, beyond automatic edge-crossing, i.e., the algorithmic selection, duplication, placing, and connecting of problematic vertices, we see value in a **semi-automatic, human-in-the-loop approach to vertex splitting**. Different users may find different vertices more or less problematic, and thus may wish to see certain vertices split or not. Interactive selection of these vertices, coupled with algorithmic placing and connecting of these vertices, may yield a personalized graph drawing, optimized to each user. Finally, given the current interest in **artificial intelligence**, we wonder whether such tools could split vertices optimally at a data level, i.e before the graph has been rendered. Conceptually, this could be more efficient than the currently studied approaches to vertex splitting at the drawing level. Overall, we believe that vertex splitting holds substantial, unexplored value as a general tool with which to improve the readability of network visualizations, agnostic of the layout algorithm chosen. Given the prevalence of complex and dense graphs in biological network visualization efforts, we thus argue that vertex splitting holds particular value here.

Application-Oriented Wiggliness In Chapter 5, we lay the foundations to justify the future investigation of animated node wiggliness for node attribute uncertainty visualization. This is an important first step in studying and illustrating the value of wiggliness for uncertainty visualization efforts in biological networks. However, we see the need for two key follow-up studies to investigate its actual usefulness to the field. First, the effectiveness of animated wiggliness as a visual channel must be studied on larger networks, i.e., the kind of sizes that one would expect in modern biological network visualization applications. It is possible that the added visual fatigue caused by the nodes'

movement may disqualify animated wiggliness for such large networks. Alternatively, it may indicate that animated wiggliness should only be employed as a *Focus+Context* technique for smaller subsets of a larger graph (e.g., when combined with semantic zooming), instead of the graph as a whole. Second, we must gauge how to best implement animated wiggliness "in the wild", i.e., characterize the additional constraints the domain of biological network visualization brings with it. Here, we see value in collaborations with domain experts who may reveal some of these constraints and assist in fine-tuning wiggliness for domain-specific applications.

A Fully-Fledged Penta Dashboard In Chapter 7, we propose the prototype of the *Penta* dashboard for the visualization of compound graphs and demonstrate its potential for biological network visualization using an example of *KEGG* pathway visualization. In summary, *Penta* aims to decompose the compound-graph-visualization problem into five separate, but linked, views. However, the implementation of *Penta* is far from finalized and production-ready. For this dashboard to be truly useful to biological network visualization domain experts, we see the need for an application-oriented follow-up study in collaboration with relevant experts in order to ensure *Penta*'s usefulness to the field. First, in order to make *Penta* useful within the context of experimental data analysis and exploration, such as gene expression analyses or protein abundance studies, we see value in allowing the visual **integration of node and edge-specific (meta)data**. For example, within the context of gene expression studies, experts might aim to encode gene expression differences in a (circular) node's surface area, the direction of the difference in a diverging color scale, and the statistical significance of a difference in a node's shape or opacity. Second, for the tool to be useful "out of the box", it should be **integrated into existing and commonly utilized knowledge graph repositories**, such as *KEGG* or *Reactome*. Conceptually, users would need only to upload their experimental data to the tool and then select the pathways they wish to visually explore, from their preferred repository. Third, we see value in investigating **domain-specific visual idioms and language** when visualizing networks, nodes, and edges. Some **biological (sub)domains may reserve certain visual channels,** such as node color or shape, for particular data or topological properties. More broadly, certain domains may also employ certain graph drawing styles for certain data types, e.g., schematic node-link diagrammatic visualizations for *KEGG* pathways or radial node-link diagrams for gene variation graphs. Finally, we may need to **facilitate particular (topological) analyses** for particular biological (sub)domains, such as edge predictions within the context of metabolomic networks, motif identification in the context of gene regulatory networks, or comparative visualization for gene co-expression analyses.

8.3 Future Research Directions

Uncertainty in Biological Network Visualization As discussed in Chapter 5, uncertainty in networks emerges in three key ways: i) edge uncertainty in so-called probabilistic graphs, ii) layout uncertainty caused by differences in layout algorithms,

and iii) attribute uncertainty in nodes and edges. As also described in this chapter, uncertainty itself comes in different forms, i.e., aleatoric, epistemic, and ontological. While the uncertainty in general biological data has been characterized, we wonder whether biological networks, biological experimental data, and the combination thereof, bring with them still unknown and unaddressed challenges:

1. What biological visual idioms and languages constrain uncertain biological network visualization efforts?
2. What particular network analysis tasks must be supported by certain uncertain biological network visualizations?
3. How should the simultaneous visualization of multiple (different types of) uncertainties in biological networks be handled?

Here, we argue that future research should focus on surveying and characterizing the different types of uncertainty and their importance in biological network visualization, in order to subsequently tackle their visualization more completely and meaningfully.

Beyond Simple Compound Graphs As discussed in Chapter 6, the majority of compound graph visualization tools and approaches deal with either some form of disjoint group structure or specifically non-disjoint but non-hierarchical group structures. This means that there exist few techniques for the visualization of overlapping and hierarchical data specifically. Given the breadth of biological networks, from protein-protein interaction networks through gene variation graphs to phylogenetic trees, it is perhaps unsurprising that such overlapping and hierarchical structures can also be found in biological network visualization endeavors. Thus, we see ample opportunity to develop novel approaches and domain-specific tools that aim to fill this gap. Such approaches should allow for the meaningful visual analysis and exploration of the graph's group structure across the various levels of its hierarchy. Additionally, intersections (at various levels of the hierarchical group structure) should be visually communicated to the user effectively. Finally, care would need to be taken not to impede the visual exploration and analysis of the compound's graph's topology itself. Such an approach would be interesting not only to biological network visualization efforts, but general compound graph visualization endeavors as well. Within the context of, for example, (dynamic) social networks, individual people, i.e., nodes, may fall within (hierarchical) overlapping group structures. Here, *Penta* could facilitate the effective visualization of such data both on a group and topological level.

Automated Schematic Representations Within the context of visual pathway analyses, domain experts frequently rely on the (hand-drawn) pathway visualization provided by their knowledge graph repository of choice, such as *KEGG* or *Reactome*. These visualizations adhere to particular visualization conventions, rendering nodes and edges in particular ways. Indeed, domain experts often become so familiar with the subgraphs and their visual language that (sub)graph and visualization become inextricably linked. Introducing novel approaches to pathway visualization is, hence, often met with

skepticism and resistance. Here, we see an opportunity to develop automated, schematic network visualization techniques that adhere to the visual language(s) of these repositories. This could allow for the reproduction of existing pathway visualization, thereby providing domain experts with a familiar visual language, but with additional interactivity and analysis tools necessitated by modern biological network analyses. Additionally, this could allow for the visualization of pathways not yet featured in these repositories in a manner familiar to domain experts.

Statistically Augmented Network Visualization As noted in Chapter 2, the majority of biological network visualizations and available tools were designed for the purposes of exploratory network analysis and hypothesis generation. Within this context, we note that few interactive visual systems featured analysis techniques or descriptive metrics to assist or guide users. Even comparatively simple centrality and graph density measures were surprisingly uncommon. Here, we wonder whether (especially in the context of more complex analysis efforts such as graph comparisons or edge predictions) the inclusion of more sophisticated analysis techniques could allow domain experts to better explore their data and generate novel hypotheses. Depending on the biological (sub)domain, different techniques could more effectively and efficiently guide them to areas of topological interest, which may subsequently be of substantive interest as well. For example, topologically identified or data-informed clusters could be ranked based on their densities, allowing users to quickly assess which clusters are the most/least connected and thus where to start their visual exploration. Nodes could be ranked and highlighted based on a selection of centrality measures, thereby guiding users to nodes of possible interest, which could then be further explored using an appropriate ego network visualization. In general, such descriptive statistics of nodes, edges, and clusters could provide users with topologically meaningful approaches to filtering and sifting through (especially) large biological network visualizations.

8.4 Conclusion

In this dissertation, we identified several outstanding challenges in the field of biological network visualization (Section 2) and focused on three particular research questions that we deemed particularly relevant and interesting. While we may have tackled these three research questions in the context of abstract network visualization, we argue that our findings are still highly relevant to the biological domain. For example, within the context of metabolic pathway visualization, domain experts might benefit from understanding novel approaches for the visualization of pathway group structure (Chapters 6 and 7). Depending on the particular visual task at hand, a domain expert might tackle their exploration of such pathway data not using conventional (interchangeable and juxtaposed) multiples, but using, for example, matrices to more quickly understand which entities map to multiple pathways. Alternatively, if retaining an overview of pathway topologies is relevant, a domain expert might use (line, region, or hybrid) overlays to simultaneously explore pathway topologies and intersections. Within the context of gene regulatory

networks, visual analysts might benefit from simplifying the network's visualization to only those biological entities relevant to some select gene using an k-hop ego network (Chapter 4). This might enable researchers to quickly explore the regulation of just a particular gene of interest, without needing to consider the added complexity added by the larger network surrounding it. Within the context of gene expression data analyses, domain experts might benefit from visualizing the deviation of gene expression levels in their biological network visualizations (Chapter 5) alongside expression levels themselves, whether this be done using animated wiggliness or more conventional approaches. This could allow researchers to make more informed decisions about what genes to select for follow-up studies, i.e., using a combination of not only a gene's topological and statistical information but its uncertainty as well. Finally, within the context of multi-omics networks, we see (a more mature version of) iterative vertex splitting (Chapter 3) as an ultimately useful approach to improving the readability of such dense and complex graph drawings. For example, a node involved in many reactions across multiple omics levels might, owing to its large number of incident edges, induce many edge crossings in the drawing. Splitting such a node (multiple times) might notably simplify each omics level's visual representation, and thereby improve a domain expert's ability to visually explore and analyze the network.

We also argue that the more fundamental nature of the research presented in this dissertation makes it useful beyond just biological network visualization, as the challenges presented by complex networks rear their heads in a great many domains. For example, an approach such as readability-improving vertex splitting (Chapter 3) might, in the context of social network visualization, assist in breaking apart highly connected hub nodes, whose many incident edges (and subsequently many induced edge crossings) can make graph topology hard to read. Such nodes could be split multiple times, perhaps based on their group memberships, thereby making the graph drawing more visually understandable, both locally and globally. Alternatively, a *Focus+Context* approach, such as ego network visualization (Chapter 4), might, after being highlighted by our work, find more common applications outside of primarily the (dynamic) social network visualization. For example, within the context of metropolitan transportation networks, layered node-link diagrammatic ego network representations might allow passengers to more easily identify which connections are within their immediate reach. Finally, algorithmically identifying clusters of nodes in a network is common across a variety of domains. These clusters form a group structure that must be visualized alongside the graph's topology (Section 6). Here, we see the opportunity for a more mature version of *Penta* (Section 7) to be of possible use in allowing users across domains to effectively visually explore the node clusters of their network.

All in all, in this dissertation, we first characterized the current state of biological network visualization and identified several outstanding challenges. Inspired by these (uniquely biological) problems, we tackled three key challenges, i.e., the visualization of large and dense biological networks, the visualization of uncertainty in biological networks, and the visualization of group structures in biological networks. To tackle the

three challenges, we study i) improving the visualization of commonly utilized straight-line node-link diagrams through iterative vertex splitting, ii) visualizing node-relative topology as an underexplored *Focus+Context* technique, iii) the visualization of node attribute uncertainties using animation as well as common alternatives, and, finally, iv) the visualization of compound graphs in order to subsequently propose a prototypical dashboard for the visualization thereof. While much work remains to be done, we believe the work presented here has laid some important groundwork, both in the context of abstract and biological network visualization.

Overview of Generative AI Tools Used

No generative artificial intelligence (AI) tools were utilized in the conceptualization, analysis, or writing of any of the here presented content.

List of Figures

List of Tables

Bibliography

[AABS+14] Ali K. Al-Awami, Johanna Beyer, Hendrik Strobelt, Narayanan Kasthuri, Jeff W. Lichtman, Hanspeter Pfister, and Markus Hadwiger. NeuroLines: A Subway Map Metaphor for Visualizing Nanoscale Neuronal Connectivity. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2369–2378, December 2014.

[AAWS09] Bill Andreopoulos, Aijun An, Xiaogang Wang, and Michael Schroeder. A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, 10(3):297–314, May 2009.

[AB08] Gediminas Adomavicius and Jesse Bockstedt. C-TREND: Temporal Cluster Graphs for Identifying and Visualizing Trends in Multiattribute Transactional Data. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):721–735, June 2008.

[ABF+07] Adel Ahmed, Vladimir Batagelj, Xiaoyan Fu, Seok-hee Hong, Damian Merrick, and Andrej Mrvar. Visualisation and analysis of the internet movie database. In *2007 6th International Asia-Pacific Symposium on Visualization*, pages 17–24, February 2007.

[ABHR+13] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 483–492, New York, NY, USA, April 2013. Association for Computing Machinery.

[ABZD13] Ala Abuthawabeh, Fabian Beck, Dirk Zeckzer, and Stephan Diehl. Finding structures in multi-type code couplings with node-link and matrix visualizations. In *2013 First IEEE Working Conference on Software Visualization (VISSOFT)*, pages 1–10, September 2013.

[ACZ+21] Michael Aichem, Tobias Czauderna, Yan Zhu, Jinxin Zhao, Matthias Klapperstück, Karsten Klein, Jian Li, and Falk Schreiber. Visual exploration of large metabolic models. *Bioinformatics*, 37(23):4460–4468, December 2021.

[ADLD+20] Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen Kobourov, and Mingwei Li. Graph Drawing via Gradient Descent, $(GD)^2$. In David Auber and Pavel Valtr, editors, *Graph Drawing and Network Visualization*, Lecture Notes in Computer Science, pages 3–17, Cham, 2020. Springer International Publishing.

[ADM+19] Lorenzo Angori, Walter Didimo, Fabrizio Montecchiani, Daniele Pagliuca, and Alessandra Tappini. ChordLink: A New Hybrid Visualization Model. In *Graph Drawing and Network Visualization: 27th International Symposium, GD 2019, Prague, Czech Republic, September 17–20, 2019, Proceedings*, pages 276–290, Berlin, Heidelberg, September 2019. Springer-Verlag.

[ADM+22] Lorenzo Angori, Walter Didimo, Fabrizio Montecchiani, Daniele Pagliuca, and Alessandra Tappini. Hybrid Graph Visualizations With ChordLink: Algorithms, Experiments, and Applications. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1288–1300, February 2022.

[AES05] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 111–117, Minneapolis, MN, USA, 2005. IEEE.

[AKK+10] Mario Albrecht, Andreas Kerren, Karsten Klein, Oliver Kohlbacher, Petra Mutzel, Wolfgang Paul, Falk Schreiber, and Michael Wybrow. On Open Problems in Biological Network Visualization. In David Eppstein and Emden R. Gansner, editors, *Graph Drawing*, pages 256–267, Berlin, Heidelberg, 2010. Springer.

[AKY05a] James Abello, Stephen G. Kobourov, and Roman Yusufov. Visualizing Large Graphs with Compound-Fisheye Views and Treemaps. In János Pach, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 431–441, Berlin, Heidelberg, 2005. Springer.

[AKY+05b] Kazuharu Arakawa, Nobuaki Kono, Yohei Yamada, Hirotada Mori, and Masaru Tomita. KEGG-Based Pathway Visualization Tool for Complex Omics Data. *In Silico Biology*, 5(4):419–423, January 2005.

[Alb05] Réka Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118(21):4947–4957, November 2005.

[ALMAN18] Gregorio Alanis-Lobato, Pablo Mier, and Miguel Andrade-Navarro. The latent geometry of the human protein interaction network. *Bioinformatics*, 34(16):2826–2834, August 2018.

[AMA07a] Daniel Archambault, Tamara Munzner, and David Auber. Grouse: Feature-Based, Steerable Graph Hierarchy Exploration. In *EuroVis*, volume 2007, pages 67–74, 2007.

[AMA07b] Daniel Archambault, Tamara Munzner, and David Auber. TopoLayout: Multilevel Graph Layout by Topological Features. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):305–317, March 2007.

[AMA08] Daniel Archambault, Tamara Munzner, and David Auber. GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):900–913, July 2008.

[AMA11] Daniel Archambault, Tamara Munzner, and David Auber. Tugging Graphs Faster: Efficiently Modifying Path-Preserving Hierarchies for Browsing Paths. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):276–289, March 2011. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[AMA+16] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. The State-of-the-Art of Set Visualization. *Computer Graphics Forum*, 35(1):234–260, 2016.

[AP13] Daniel Archambault and Helen C. Purchase. Mental Map Preservation Helps User Orientation in Dynamic Graphs. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing*, pages 475–486, Berlin, Heidelberg, 2013. Springer.

[APP11] Daniel Archambault, Helen Purchase, and Bruno Pinaud. Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, April 2011.

[APS14] Jae-wook Ahn, Catherine Plaisant, and Ben Shneiderman. A Task Taxonomy for Network Evolution Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):365–376, March 2014.

[ARRC11] Basak Alper, Nathalie Riche, Gonzalo Ramos, and Mary Czerwinski. Design Study of LineSets, a Novel Set Visualization Technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, December 2011.

[ASA+23] Moataz Abdelaal, Nathan D. Schiele, Katrin Angerbauer, Kuno Kurzhals, Michael Sedlmair, and Daniel Weiskopf. Comparative Evaluation of Bipartite, Node-Link, and Matrix-Based Network Representations. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):896–906, January 2023.

[AvHK06] James Abello, Frank van Ham, and Neeraj Krishnan. ASK-GraphView: A Large Scale Graph Visualization System. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, September 2006.

[AWM10] H. Akiba, C. Wang, and K.-L. Ma. AniViz: A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications*, 30(5):61–71, 2010.

[BA99] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999.

[Bau72] David F. Bauer. Constructing Confidence Sets Using Rank Statistics. *Journal of the American Statistical Association*, 67(339):687–690, September 1972.

[BB03] Albert-László Barabási and Eric Bonabeau. Scale-Free Networks. *Scientific American*, 288(5):60–69, 2003.

[BB05] Michael Baur and Ulrik Brandes. Crossing Reduction in Circular Layouts. In Juraj Hromkovič, Manfred Nagl, and Bernhard Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science*, pages 332–343, Berlin, Heidelberg, 2005. Springer.

[BBDW14] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. The State of the Art in Visualizing Dynamic Graphs. In *EuroVis - STARs*. The Eurographics Association, 2014.

[BBDW17] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

[BBHR+16] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum*, 35(3):693–716, June 2016.

[BBIF12] Nadia Boukhelifa, Anastasia Bezerianos, Tobias Isenberg, and Jean-Daniel Fekete. Evaluating Sketchiness as a Visual Variable for the Depiction of Qualitative Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2769–2778, December 2012.

[BBL12] Ilya Boyandin, Enrico Bertini, and Denis Lalanne. A Qualitative Study on the Exploration of Temporal Changes in Flow Maps with Animation and Small-Multiples. *Computer Graphics Forum*, 31(3pt2):1005–1014, 2012.

[BBS14] Christian Bachmaier, Ulrik Brandes, and Falk Schreiber. Biological Networks. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, pages 621–651. CRC Press, 2014.

[BC19] Anna D. Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1017, March 2019.

[BCD+16] Carla Binucci, Markus Chimani, Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. Placing Arrows in Directed Graph Drawings. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing and Network Visualization*, Lecture Notes in Computer Science, pages 44–51, Cham, 2016. Springer International Publishing.

[BD05] M. Balzer and O. Deussen. Exploring Relations within Software Systems Using Treemap Enhanced Hierarchical Graphs. In *3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pages 1–6, September 2005.

[BDF+14] Michael Behrisch, James Davey, Fabian Fischer, Olivier Thonnard, Tobias Schreck, Daniel Keim, and Jörn Kohlhammer. Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom. *Computer Graphics Forum*, 33(3):411–420, 2014.

[Ber05] Pavel Berkhin. A Survey on PageRank Computing. *Internet Mathematics*, 2(1):73–120, January 2005.

[BETT98] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, USA, 1st edition, 1998.

[BFBW00] Steve Blenkinsop, Pete Fisher, Lucy Bastin, and Jo Wood. Evaluating the Perception of Uncertainty in Alternative Visualization Strategies. *Cartographica*, 37(1):1–14, March 2000.

[BFG+21] Michael A Bekos, Henry Förster, Christian Geckeler, Lukas Holländer, Michael Kaufmann, Amadäus M Spallek, and Jan Splett. A Heuristic Approach Towards Drawings of Graphs With High Crossing Resolution. *The Computer Journal*, 64(1):7–26, January 2021.

[BGL11] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, January 2011.

[BGP+13] Vincenzo Bonnici, Rosalba Giugno, Alfredo Pulvirenti, Dennis Shasha, and Alfredo Ferro. A subgraph isomorphism algorithm and its application to biochemical data. *BMC Bioinformatics*, 14(S7):S13, April 2013.

[BH03] Gary D. Bader and Christopher WV Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(1):2, January 2003.

[BHJ+14] Georges-Pierre Bonneau, Hans-Christian Hege, Chris R. Johnson, Manuel M. Oliveira, Kristin Potter, Penny Rheingans, and Thomas Schultz. Overview and State-of-the-Art of Uncertainty Visualization. In Charles D. Hansen, Min Chen, Christopher R. Johnson, Arie E. Kaufman, and Hans Hagen, editors, *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, Mathematics and Visualization, pages 3–27. Springer, London, 2014.

[BHP+24] N. Brich, T.a. Harbig, M. Witte Paz, K. Nieselt, and M. Krone. ProtEGOnist: Visual Analysis of Interactions in Small World Networks Using Ego-graphs. *Computer Graphics Forum*, 43(3):e15078, 2024.

[BHR17] Juhee Bae, Tove Helldin, and Maria Riveiro. Understanding Indirect Causal Relationships in Node-Link Graphs. *Computer Graphics Forum*, 36(3):411–421, 2017.

[BHRD+15] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J-D. Fekete, and T. Grabowski. Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks. *Computer Graphics Forum*, 34(3):31–40, 2015.

[BHW+21] Michael Burch, Weidong Huang, Mathew Wakefield, Helen C. Purchase, Daniel Weiskopf, and Jie Hua. The State of the Art in Empirical User Evaluation of Graph Visualizations. *IEEE Access*, 9:4173–4198, 2021.

[BKC+12] Rita Borgo, Johannes Kehrer, David H. S. Chung, Eamonn Maguire, Robert S. Laramee, Helwig Hauser, Matthew Ward, and Min Chen. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. *Eurographics 2013 - State of the Art Reports*, page 25 pages, 2012.

[BM13] Matthew Brehmer and Tamara Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013.

[BMGK08] Aaron Barsky, Tamara Munzner, Jennifer Gardy, and Robert Kincaid. Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1253–1260, November 2008.

[BMK96] Jim Blythe, Cathleen McGrath, and David Krackhardt. The effect of graph layout on inference from social network data. In Franz J. Brandenburg, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 40–51, Berlin, Heidelberg, 1996. Springer.

[BMMS91] A. Buja, J.A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proceeding Visualization '91*, pages 156–163, October 1991.

[BMSW19] Aurora S Blucher, Shannon K McWeeney, Lincoln Stein, and Guanming Wu. Visualization of drug target interactions in the contexts of pathways and networks with ReactomeFIViz. *F1000Res.*, 8:908, June 2019.

[BNH+19] Wolfgang Beyer, Adam M Novak, Glenn Hickey, Jeffrey Chan, Vanessa Tan, Benedict Paten, and Daniel R Zerbino. Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics*, 35(24):5318–5320, 2019.

[BNML19] Alex Bigelow, Carolina Nobre, Miriah Meyer, and Alexander Lex. Origraph: Interactive Network Wrangling. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 81–92, October 2019.

[BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011.

[Bol01] Béla Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 2 edition, 2001.

[Bon36] C. Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[Bon72] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2(1):113–120, January 1972.

[Bon87] Phillip Bonacich. Power and Centrality: A Family of Measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[Bon07] Phillip Bonacich. Some unique properties of eigenvector centrality. *Social Networks*, 29(4):555–564, October 2007.

[BPF14a] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. GraphDiaries: Animated Transitions andTemporal Navigation for Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, May 2014.

[BPF14b] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages

877–886, New York, NY, USA, April 2014. Association for Computing Machinery.

[BPL13] Benjamin Bach, Emmanuel Pietriga, and Ilaria Liccardi. Visualizing Populated Ontologies with OntoTrix:. *International Journal on Semantic Web and Information Systems*, 9(4):17–40, October 2013.

[BRH+17] Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, and Tim Dwyer. Towards Unambiguous Edge Bundling: Investigating Confluent Drawings for Network Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):541–550, January 2017.

[Bro04] Ross Brown. Animated visual vibrations as an uncertainty visualisation technique. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '04, pages 84–89, New York, NY, USA, June 2004. Association for Computing Machinery.

[BS08] Michael Baur and Thomas Schank. *Dynamic Graph Drawing in Visone*. 2008.

[BSC13] Christopher W. Berardi, Erin T. Solovey, and Mary L. Cummings. Investigating the efficacy of network visualizations for intelligence tasks. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 278–283, June 2013.

[BSH13] Gwenael Bothorel, Mathieu Serrurier, and Christophe Hurter. Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Baoxin Li, Fatih Porikli, Victor Zordan, James Klosowski, Sabine Coquillart, Xun Luo, Min Chen, and David Gotz, editors, *Advances in Visual Computing*, Lecture Notes in Computer Science, pages 396–405, Berlin, Heidelberg, 2013. Springer.

[BSL+14] Joachim Böttger, Alexander Schäfer, Gabriele Lohmann, Arno Villringer, and Daniel S. Margulies. Three-Dimensional Mean-Shift Edge Bundling for the Visualization of Functional Connectivity in the Brain. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):471–480, March 2014.

[BSW13] Michael Burch, Benjamin Schmidt, and Daniel Weiskopf. A Matrix-Based Visualization for Exploring Dynamic Compound Digraphs. In *2013 17th International Conference on Information Visualisation*, pages 66–73, July 2013.

[BTM+19] J. Byška, T. Trautner, S.m. Marques, J. Damborský, B. Kozlíková, and M. Waldner. Analysis of Long Molecular Dynamics Simulations Using

Interactive Focus+Context Visualization. *Computer Graphics Forum*, 38(3):441–453, 2019.

[BVK+15] Eric Bonnet, Eric Viara, Inna Kuperstein, Laurence Calzone, David P A Cohen, Emmanuel Barillot, and Andrei Zinovyev. NaviCell Web Service for network-based data visualization. *Nucleic Acids Res.*, 43(W1):W560–5, July 2015.

[BVPtHR09] Ralph Brecheisen, Anna Vilanova, Bram Platel, and Bart ter Haar Romeny. Parameter Sensitivity Visualization for DTI Fiber Tracking. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1441–1448, November 2009.

[BW11] Juhee Bae and Benjamin Watson. Developing and Evaluating Quilts for the Depiction of Large Layered Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2268–2275, December 2011.

[BWC01] Lyn Bartram, Colin Ware, and Tom Calvert. Moving Icons, Detection and Distraction. *Interact*, July 2001.

[BWK+16] Anwesha Bohler, Guanming Wu, Martina Kutmon, Leontius Adhika Pradhana, Susan L Coort, Kristina Hanspers, Robin Haw, Alexander R Pico, and Chris T Evelo. Reactome from a WikiPathways perspective. *PLoS Comput. Biol.*, 12(5):e1004941, May 2016.

[BYCH+12] Christopher W Bartlett, Soo Yeon Cheong, Liping Hou, Jesse Paquette, Pek Yee Lum, Günter Jäger, Florian Battke, Corinna Vehlow, Julian Heinrich, Kay Nieselt, Ryo Sakai, Jan Aerts, and William C Ray. An eQTL biological data visualization challenge and approaches from the visualization community. *BMC Bioinformatics*, 13(S8):S8, December 2012.

[Car99] Mackinlay Card. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, January 1999.

[CBDM17] Chunlei Chang, Benjamin Bach, Tim Dwyer, and Kim Marriott. Evaluating Perceptually Complementary Views for Network Exploration Tasks. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1397–1407, New York, NY, USA, May 2017. Association for Computing Machinery.

[CCC16] Jennifer Chang, Hyejin Cho, and Hui-Hsien Chou. Mango: combining and analyzing heterogeneous biological networks. *BioData Min.*, 9(1):25, August 2016.

[CCP07] Christopher Collins, Sheelagh Carpendale, and Gerald Penn. Visualization of Uncertainty in Lattices to Support Decision-Making. In

IEEE-VGTC Symposium on Visualization. The Eurographics Association, 2007.

[CDA+14] S. Chaturvedi, C. Dunne, Z. Ashktorab, R. Zachariah, and B. Shneiderman. Group-in-a-Box Meta-Layouts for Topological Clusters and Attribute-Based Groups: Space-Efficient Visualizations of Network Communities and Their Ties. *Computer Graphics Forum*, 33(8):52–68, 2014.

[CDBM21] Kun-Ting Chen, Tim Dwyer, Benjamin Bach, and Kim Marriott. It's a Wrap: Toroidal Wrapping of Network Visualisations Supports Cluster Understanding Tasks. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–12, New York, NY, USA, May 2021. Association for Computing Machinery.

[CDMB20] Kun-Ting Chen, Tim Dwyer, Kim Marriott, and Benjamin Bach. DoughNets: Visualising Networks Using Torus Wrapping. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–11, New York, NY, USA, April 2020. Association for Computing Machinery.

[CFR15] Colin W Combe, Lutz Fischer, and Juri Rappsilber. xiNET: cross-link network maps with residue resolution. *Mol. Cell. Proteomics*, 14(4):1137–1147, April 2015.

[CFSV04] L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, October 2004.

[CG14] Michael Correll and Michael Gleicher. Error Bars Considered Harmful: Exploring Alternate Encodings for Mean and Error. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2142–2151, December 2014.

[CGH+24] Melanie Conroy, Christina Gillmann, Francis Harvey, Tamara Mchedlidze, Sara Irina Fabrikant, Florian Windhager, Gerik Scheuermann, Timothy R. Tangherlini, Christopher N. Warren, Scott B. Weingart, Malte Rehbein, Katy Börner, Kimmo Elo, Stefan Jänicke, Andreas Kerren, Martin Nöllenburg, Tim Dwyer, Oyvind Eide, Stephen Kobourov, and Gregor Betz. Uncertainty in humanities network visualization. *Frontiers in Communication*, 8, January 2024.

[CGLN19] Yuzhou Chen, Yulia Gel, R., Vyacheslav Lyubchich, and Kusha Nezafati. Snowboot: Bootstrap Methods for Network Inference. *The R Journal*, 10(2):95, 2019.

[CH18] Rajkumar Chakraborty and Yasha Hasija. miDerma: An Integrated Database and Tool for Analysis of miRNAs associated with Dermatological Disorders. In *2018 International Conference on Bioinformatics and Systems Biology (BSB)*, pages 170–173, Allahabad, India, October 2018. IEEE.

[Chi97] Michelene T.H. Chi. Quantifying Qualitative Analyses of Verbal Data: A Practical Guide. *Journal of the Learning Sciences*, 6(3):271–315, July 1997.

[CHI10] Han-Yu Chuang, Matan Hofree, and Trey Ideker. A Decade of Systems Biology. *Annual Review of Cell and Developmental Biology*, 26(1):721–744, November 2010.

[CHL13] Jonathan Cox, Donald House, and Michael Lindell. VISUALIZING UNCERTAINTY IN PREDICTED HURRICANE TRACKS. *International Journal for Uncertainty Quantification*, 3(2), 2013.

[CHWO13] Pádraig Cunningham, Martin Harrigan, Guangyu Wu, and Derek O'callaghan. Characterizing ego-networks using motifs. *Network Science*, 1(2):170–190, August 2013.

[CK21] Genís Calderer and Marieke L. Kuijjer. Community Detection in Large-Scale Bipartite Biological Networks. *Frontiers in Genetics*, 12, April 2021.

[CMH12] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 74–77, New York, NY, USA, May 2012. Association for Computing Machinery.

[CMH18] Michael Correll, Dominik Moritz, and Jeffrey Heer. Value-Suppressing Uncertainty Palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–11, New York, NY, USA, April 2018. Association for Computing Machinery.

[CMJ+20] A. Chatzimparmpas, R. M. Martins, I. Jusufi, K. Kucher, F. Rossi, and A. Kerren. The State of the Art in Enhancing Trust in Machine Learning Models with the Use of Visualizations. *Computer Graphics Forum*, 39(3):713–756, 2020.

[CMS11] Julia Chuzhoy, Yury Makarychev, and Anastasios Sidiropoulos. On Graph Crossing Number and Edge Planarization. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1050–1069. Society for Industrial and Applied Mathematics, January 2011.

[CPAM18] Antonio Cruz, Joel P. Arrais, and Penousal Machado. Interactive Network Visualization of Gene Expression Time-Series Data. In *2018 22nd International Conference Information Visualisation (IV)*, pages 574–580, Fisciano, Italy, July 2018. IEEE.

[CPS11] Nathaniel Cesario, Alex Pang, and Lisa Singh. Visualizing node attribute uncertainty in graphs. In *Visualization and Data Analysis 2011*, volume 7868, pages 169–181. SPIE, January 2011.

[CR00] A. Cedilnik and P. Rheingans. Procedural annotation of uncertain information. In *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, pages 77–84, October 2000.

[CY06] Jingchun Chen and Bo Yuan. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics (Oxford, England)*, 22(18):2283–2290, September 2006.

[CZSX20] Le Chang, Guangyan Zhou, Othman Soufan, and Jianguo Xia. miRNet 2.0: network-based visual analytics for miRNA functional analysis and systems biology. *Nucleic Acids Res.*, 48(W1):W244–W251, July 2020.

[DCG+12] J. Dubois, L. Cottret, A. Ghozlane, D. Auber, F. Bringaud, P. Thebault, F. Jourdan, and R. Bourqui. Systrip: A Visual Environment for the Investigation of Time-series Data in the Context of Metabolic Networks. In *2012 16th International Conference on Information Visualisation*, pages 204–213, Montpellier, France, July 2012. IEEE.

[DCP+10] Emek Demir, Michael P Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'Eustachio, Carl Schaefer, Joanne Luciano, Frank Schacherer, Irma Martinez-Flores, Zhenjun Hu, Veronica Jimenez-Jacinto, Geeta Joshi-Tope, Kumaran Kandasamy, Alejandra C Lopez-Fuentes, Huaiyu Mi, Elgar Pichler, Igor Rodchenkov, Andrea Splendiani, Sasha Tkachev, Jeremy Zucker, Gopal Gopinath, Harsha Rajasimha, Ranjani Ramakrishnan, Imran Shah, Mustafa Syed, Nadia Anwar, Özgün Babur, Michael Blinov, Erik Brauner, Dan Corwin, Sylva Donaldson, Frank Gibbons, Robert Goldberg, Peter Hornbeck, Augustin Luna, Peter Murray-Rust, Eric Neumann, Oliver Ruebenacker, Matthias Samwald, Martijn Van Iersel, Sarala Wimalaratne, Keith Allen, Burk Braun, Michelle Whirl-Carrillo, Kei-Hoi Cheung, Kam Dahlquist, Andrew Finney, Marc Gillespie, Elizabeth Glass, Li Gong, Robin Haw, Michael Honig, Olivier Hubaut, David Kane, Shiva Krupa, Martina Kutmon, Julie Leonard, Debbie Marks, David Merberg, Victoria Petri, Alex Pico, Dean Ravenscroft, Liya Ren, Nigam Shah, Margot Sunshine, Rebecca Tang, Ryan Whaley, Stan Letovksy, Kenneth H Buetow, Andrey Rzhetsky, Vincent Schachter, Bruno S Sobral, Ugur Dogrusoz, Shannon McWeeney, Mirit Aladjem, Ewan Birney, Julio Collado-Vides, Susumu

Goto, Michael Hucka, Nicolas Le Novère, Natalia Maltsev, Akhilesh Pandey, Paul Thomas, Edgar Wingender, Peter D Karp, Chris Sander, and Gary D Bader. The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, 28(9):935–942, September 2010.

[DDPA15] Manlio De Domenico, Mason A. Porter, and Alex Arenas. MuxViz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, 3(2):159–176, June 2015.

[DEKB+14] Kasper Dinkla, Mohammed El-Kebir, Cristina-Iulia Bucur, Marco Siderius, Martine J. Smit, Michel A. Westenberg, and Gunnar W. Klau. eXamine: Exploring annotated modules in networks. *BMC Bioinformatics*, 15(1):201, July 2014.

[DGC+17] J S Dussaut, C A Gallo, F Cravero, M J Martínez, J A Carballido, and I Ponzoni. GeRNet: a gene regulatory network tool. *Biosystems.*, 162:1–11, December 2017.

[DGDMT21] Emilio Di Giacomo, Walter Didimo, Fabrizio Montecchiani, and Alessandra Tappini. A User Study on Hybrid Graph Visualizations. In Helen C. Purchase and Ignaz Rutter, editors, *Graph Drawing and Network Visualization*, pages 21–38, Cham, 2021. Springer International Publishing.

[DH96] Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, October 1996.

[DHRMM13] Tim Dwyer, Nathalie Henry Riche, Kim Marriott, and Christopher Mears. Edge Compression Techniques for Visualization of Dense Directed Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2596–2605, December 2013.

[DKMT18] Walter Didimo, Evgenios M. Kornaropoulos, Fabrizio Montecchiani, and Ioannis G. Tollis. A Visualization Framework and User Studies for Overloaded Orthogonal Drawings. *Computer Graphics Forum*, 37(1):288–300, 2018.

[DLM19] Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A Survey on Graph Drawing Beyond Planarity. *ACM Computing Surveys*, 52(1):4:1–4:37, February 2019.

[DLR11] Walter Didimo, Giuseppe Liotta, and Salvatore A. Romeo. Topology-Driven Force-Directed Algorithms. In Ulrik Brandes and Sabine Cornelsen, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 165–176, Berlin, Heidelberg, 2011. Springer.

[DMF15] Tuan Nhon Dang, Paul Murray, and Angus Graeme Forbes. PathwayMatrix: visualizing binary relationships between proteins in biological pathways. *BMC Proceedings*, 9(S6):S3, December 2015.

[DMF17] Tommy Dang, Paul Murray, and Angus Forbes. BioLinker: Bottom-up exploration of protein interaction networks. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 265–269, April 2017.

[DMM+14] Tim Dwyer, Christopher Mears, Kerri Morgan, Todd Niven, Kim Marriott, and Mark Wallace. Improved Optimal and Approximate Power Graph Compression for Clearer Visualisation of Dense Graphs. In *2014 IEEE Pacific Visualization Symposium*, pages 105–112, March 2014. ISSN: 2165-8773.

[DNW13] Peter Droste, Katharina Nöh, and Wolfgang Wiechert. Omix – A Visualization Tool for Metabolic Networks with Highest Usability and Customizability in Focus. *Chemie Ingenieur Technik*, 85(6):849–862, 2013.

[DP20] Evanthia Dimara and Charles Perin. What is Interaction for Data Visualization? *IEEE Transactions on Visualization and Computer Graphics*, 26(1):119–129, January 2020.

[DPG+19] Christina Durón, Yuan Pan, David H. Gutmann, Johanna Hardin, and Ami Radunskaya. Variability of Betweenness Centrality and Its Effect on Identifying Essential Genes. *Bulletin of Mathematical Biology*, 81(9):3655–3673, September 2019.

[DRO+02] Eric H. Davidson, Jonathan P. Rast, Paola Oliveri, Andrew Ransick, Cristina Calestani, Chiou-Hwa Yuh, Takuya Minokawa, Gabriele Amore, Veronica Hinman, César Arenas-Mena, Ochan Otim, C. Titus Brown, Carolina B. Livi, Pei Yun Lee, Roger Revilla, Alistair G. Rust, Zheng jun Pan, Maria J. Schilstra, Peter J. C. Clarke, Maria I. Arnone, Lee Rowen, R. Andrew Cameron, David R. McClay, Leroy Hood, and Hamid Bolouri. A Genomic Regulatory Network for Development. *Science*, 295(5560):1669–1678, March 2002.

[DS13] Cody Dunne and Ben Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3247–3256, New York, NY, USA, April 2013. Association for Computing Machinery.

[dSMZ08] MÁrcio Rosa da Silva, Hongwu Ma, and An-Ping Zeng. Centrality, Network Capacity, and Modularity as Parameters to Analyze the Core-Periphery Structure in Metabolic Networks. *Proceedings of the IEEE*, 96(8):1411–1420, August 2008.

[Duc17] César Ducruet. Multilayer dynamics of complex spatial networks: The case of global maritime flows (1977–2008). *Journal of Transport Geography*, 60:47–58, April 2017.

[DvKSW12] Kasper Dinkla, Marc J. van Kreveld, Bettina Speckmann, and Michel A. Westenberg. Kelp Diagrams: Point Set Membership Visualization. *Computer Graphics Forum*, 31(3pt1):875–884, 2012.

[DWS+14] Donny T. Daniel, Egon Wuchner, Konstantin Sokolov, Michael Stal, and Peter Liggesmeyer. Polyptychon: A Hierarchically-Constrained Classified Dependencies Visualization. In *2014 Second IEEE Working Conference on Software Visualization*, pages 83–86, September 2014.

[Ead84] Peter Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[EBJ06] Timothy M. D. Ebbels, Bernard F. Buxton, and David T. Jones. springScape: visualisation of microarray and contextual bioinformatic data using spring embedding and an 'information landscape'. *Bioinformatics*, 22(14):e99–e107, July 2006.

[EBK+25] Henry Ehlers, Nicolas Brich, Michael Krone, Martin Nöllenburg, Jiacheng Yu, Hiroaki Natsukawa, Xiaoru Yuan, and Hsiang-Yun Wu. An introduction to and survey of biological network visualization. *Computers & Graphics*, 126:104115, February 2025.

[EdMN96] P. Eades and C. F. X. de Mendonça N. Vertex splitting and tension-free layout. In Franz J. Brandenburg, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 202–211, Berlin, Heidelberg, 1996. Springer.

[EF10] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, May 2010.

[EFK01] Markus Eiglsperger, Sándor P. Fekete, and Gunnar W. Klau. Orthogonal Graph Drawing. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs*, volume 2025, pages 121–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[EHAE16] Hafez Ezaiza, Shah Rukh Humayoun, Ragaad AlTarawneh, and Achim Ebert. PerSoN-Vis: Visualizing Personal Social Networks (Ego Networks). In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 1222–1228, New York, NY, USA, May 2016. Association for Computing Machinery.

[EHK+04] Cesim Erten, Philip J. Harding, Stephen G. Kobourov, Kevin Wampler, and Gary Yee. GraphAEL: Graph Animations with Evolving Layouts. In Giuseppe Liotta, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 98–110, Berlin, Heidelberg, 2004. Springer.

[EHKP14] Alon Efrat, Yifan Hu, Stephen G. Kobourov, and Sergey Pupyrev. MapSets: Visualizing Embedded and Clustered Graphs. In Christian Duncan and Antonios Symvonis, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 452–463, Berlin, Heidelberg, 2014. Springer.

[EHP+11] Ozan Ersoy, Christophe Hurter, Fernando Paulovich, Gabriel Cantareiro, and Alex Telea. Skeleton-Based Edge Bundling for Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, December 2011.

[EKHW21] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. An Aligned Rank Transform Procedure for Multifactor Contrast Tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, pages 754–768, New York, NY, USA, October 2021. Association for Computing Machinery.

[EKR25] Henry Ehlers, Mario Kerndler, and Renata Raidou. Penta: Towards Visualizing Compound Graphs as Set-Typed Data. pages 913–921, March 2025.

[EMW86] Peter D. Eades, B. D. McKay, and Nicholas Charles Wormald. On an edge crossing problem. *Proceedings of the Ninth Australian Computer Science Conference*, pages 327–334, 1986.

[EMWR24] Henry Ehlers, Diana Marin, Hsiang-Yun Wu, and Renata Raidou. Visualizing Group Structure in Compound Graphs: The Current State, Lessons Learned, and Outstanding Opportunities. In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 1 - GRAPP, HUCAPP and IVAPP: IVAPP, pages 697–708, Rome, June 2024.

[ENS+20] Jordan M. Eizenga, Adam M. Novak, Jonas A. Sibbesen, Simon Heumos, Ali Ghaffaari, Glenn Hickey, Xian Chang, Josiah D. Seaman, Robin Rounthwaite, Jana Ebler, Mikko Rautiainen, Shilpa Garg, Benedict Paten, Tobias Marschall, Jouni Sirén, and Erik Garrison. Pangenome Graphs. *Annual Review of Genomics and Human Genetics*, 21(1):139–162, 2020.

[EPdB+25a] Henry Ehlers, Daniel Pahr, Sara di Bartolomeo, Velitchko Filipov, Hsiang-Yun Wu, and Renata G. Raidou. Wiggle! Wiggle! Wiggle! Visualizing uncertainty in node attributes in straight-line node-link

diagrams using animated wiggliness. *Computers & Graphics*, 131:104290, October 2025.

[EPdB+25b] Henry Ehlers, Daniel Pahr, Sara di Bartolomeo, Christina Stoiber, and Velitchko Filipov. *BattleGraphs: Forge, Fortify, and Fight in the Network Arena*. The Eurographics Association, 2025.

[EPF+24] Henry Ehlers, Daniel Pahr, Velitchko Filipov, Hsiang-Yun Wu, and Renata G. Raidou. Me! Me! Me! Me! A study and comparison of ego network representations. *Computers & Graphics*, 125:104123, December 2024.

[ER11] Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *WIREs Computational Molecular Science*, 1(1):68–79, 2011.

[ERG02] P. Eklund, N. Roberts, and S. Green. OntoRama: Browsing RDF ontologies using a hyperbolic-style browser. In *First International Symposium on Cyber Worlds, 2002. Proceedings.*, pages 405–411, November 2002.

[ESDS16] Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346-347:180–197, June 2016.

[ESG97] Charles R. Ehlschlaeger, Ashton M. Shortridge, and Michael F. Goodchild. Visualizing spatial data uncertainty using animation. *Computers & Geosciences*, 23(4):387–395, May 1997.

[EVRW23] Henry Ehlers, Anaïs Villedieu, Renata G. Raidou, and Hsiang-Yun Wu. Improving readability of static, straight-line graph drawings: A first look at edge crossing resolution through iterative vertex splitting. *Computers & Graphics*, 116:448–463, November 2023.

[FAM+11] Paolo Federico, Wolfgang Aigner, Silvia Miksch, Florian Windhager, and Lukas Zenk. A visual analytics approach to dynamic social networks. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 1–8, New York, NY, USA, September 2011. Association for Computing Machinery.

[FAM23] Velitchko Filipov, Alessio Arleo, and Silvia Miksch. Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies. *Computer Graphics Forum*, n/a(n/a), April 2023.

[FBP21] Álvaro Martínez Fernández, Lars Ailo Bongo, and Edvard Pedersen. GeneNet VR: Interactive visualization of large-scale biological networks using a standalone headset. *arXiv:2109.02937 [cs]*, September 2021. arXiv: 2109.02937.

[FCM+17] Takanori Fujiwara, Jia-Kai Chou, Andrew M. McCullough, Charan Ranganath, and Kwan-Liu Ma. A visual analytics system for brain functional connectivity comparison across individuals, groups, and time points. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 250–259, April 2017.

[FHQ11] Michael Farrugia, Neil Hurley, and Aaron Quigley. Exploring Temporal Ego Networks Using Small Multiples and Tree-ring Layouts. 2011.

[Fig15] Ana Figueiras. Towards the Understanding of Interaction in Information Visualization. In *2015 19th International Conference on Information Visualisation*, pages 140–147, July 2015.

[Fis05] D. Fisher. Using egocentric networks to understand communication. *IEEE Internet Computing*, 9(5):20–28, September 2005.

[FK99] Alan Frieze and Ravi Kannan. Quick Approximation to Matrices and Applications. *Combinatorica*, 19(2):175–220, February 1999.

[FKN+05] David Forrester, Stephen G. Kobourov, Armand Navabi, Kevin Wampler, and Gary V. Yee. Graphael: A System for Generalized Force-Directed Layouts. In János Pach, editor, *Graph Drawing*, pages 454–464, Berlin, Heidelberg, 2005. Springer.

[FM12] Nathaniel Fout and Kwan-Liu Ma. Fuzzy Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2335–2344, December 2012.

[FMJ+08] Akira Funahashi, Yukiko Matsuoka, Akiya Jouraku, Mineo Morohashi, Norihiro Kikuchi, and Hiroaki Kitano. CellDesigner 3.5: A Versatile Modeling Tool for Biochemical Networks. *Proceedings of the IEEE*, 96(8):1254–1265, August 2008.

[FMKT03] Akira Funahashi, Mineo Morohashi, Hiroaki Kitano, and Naoki Tanimura. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1(5):159–162, November 2003.

[FMW+21] Kun Fu, Tingyun Mao, Yang Wang, Daoyu Lin, Yuanben Zhang, and Xian Sun. DyEgoVis: Visual Exploration of Dynamic Ego-Network Evolution. *Applied Sciences*, 11(5):2399, January 2021.

[FQ11] Michael Farrugia and Aaron Quigley. Effective Temporal Graph Layout: A Comparative Study of Animation versus Static Display Methods. *Information Visualization*, 10(1):47–64, January 2011.

[FR91] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[Fre77] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[FSA+16] Yannan Fan, Keith Siklenka, Simran K Arora, Paula Ribeiro, Sarah Kimmins, and Jianguo Xia. miRNet - dissecting miRNA-target interactions and functional associations through network-based visual analysis. *Nucleic Acids Res.*, 44(W1):W135–41, July 2016.

[FT04] Y. Frishman and Ayellet Tal. Dynamic Drawing of Clustered Graphs. In *IEEE Symposium on Information Visualization*, pages 191–198, October 2004.

[FX18] Yannan Fan and Jianguo Xia. MiRNet-functional analysis and visual exploration of miRNA-target interactions in a network context. *Methods Mol. Biol.*, 1819:215–233, 2018.

[Gab24] Evelina Gabasova. evelinag/StarWars-social-network, February 2024.

[GBFM16] Theresia Gschwandtnei, Markus Bögl, Paolo Federico, and Silvia Miksch. Visual Encodings of Temporal Uncertainty: A Comparative User Study. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):539–548, January 2016.

[GCW+16] Brian R. Granger, Yi-Chien Chang, Yan Wang, Charles DeLisi, Daniel Segrè, and Zhenjun Hu. Visualization of Metabolic Interaction Networks in Microbial Communities Using VisANT 5.0. *PLOS Computational Biology*, 12(4):e1004875, April 2016.

[GDL+24] Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Alessandra Tappini. Comparative Study and Evaluation of Hybrid Visualizations of Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3503–3515, July 2024.

[GES+22] Yang Guo, Fatemeh Esfahani, Xiaojian Shao, Venkatesh Srinivasan, Alex Thomo, Li Xing, and Xuekui Zhang. Integrative COVID-19 biological network inference with probabilistic core decomposition. *Briefings in Bioinformatics*, 23(1):bbab455, January 2022.

[GF09] Simone Garlandini and Sara Irina Fabrikant. Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization. In Kathleen Stewart Hornsby, Christophe Claramunt, Michel Denis, and Gérard Ligozat, editors, *Spatial Information Theory*, pages 195–211, Berlin, Heidelberg, 2009. Springer.

[GFC04] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *IEEE Symposium on Information Visualization*, pages 17–24, October 2004.

[GFC05] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. On the Readability of Graphs Using Node-Link and Matrix-Based Representations: A Controlled Experiment and Statistical Analysis. *Information Visualization*, 4(2):114–135, June 2005.

[GHK10] Emden R. Gansner, Yifan Hu, and Stephen Kobourov. GMap: Visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208, March 2010.

[GHL15] Hua Guo, Jeff Huang, and David H. Laidlaw. Representing Uncertainty in Graph Edges: An Evaluation of Paired Visual Variables. *IEEE Transactions on Visualization and Computer Graphics*, 21(10):1173–1186, October 2015.

[GHN+21] Andrea Guarracino, Simon Heumos, Sven Nahnsen, Pjotr Prins, and Erik Garrison. ODGI: understanding pangenome graphs. Technical report, Genomics Research Centre, Human Technopole, Milan, Italy, 2021.

[GJ83] M. R. Garey and D. S. Johnson. Crossing Number is NP-Complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, September 1983.

[GKN04] Emden R. Gansner, Yehuda Koren, and Stephen North. Graph drawing by stress majorization. In *Proceedings of the 12th international conference on Graph Drawing*, GD'04, pages 239–250, Berlin, Heidelberg, September 2004. Springer-Verlag.

[GKO18] Meiyuzi Gao, Philip Kortum, and Frederick Oswald. Psychometric Evaluation of the USE (Usefulness, Satisfaction, and Ease of use) Questionnaire for Reliability and Validity. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62(1):1414–1418, September 2018.

[GLA+23] Kathryn Gray, Mingwei Li, Reyan Ahmed, Md. Khaledur Rahman, Ariful Azad, Stephen Kobourov, and Katy Börner. A Scalable Method for Readable Tree Layouts. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–15, 2023.

[GMVW24] Florian Grötschla, Joël Mathys, Robert Veres, and Roger Wattenhofer. CoRe-GD: A Hierarchical Framework for Scalable Graph Visualization with GNNs, February 2024.

[GOB+10] Nils Gehlenborg, Seán I. O'Donoghue, Nitin S. Baliga, Alexander Goesmann, Matthew A. Hibbs, Hiroaki Kitano, Oliver Kohlbacher, Heiko Neuweger, Reinhard Schneider, Dan Tenenbaum, and Anne-Claude

Gavin. Visualization of omics data for systems biology. *Nature Methods*, 7(3):S56–S68, March 2010.

[GOHS15] Miriam Greis, Thorsten Ohler, Niels Henze, and Albrecht Schmidt. Investigating Representation Alternatives for Communicating Uncertainty to Non-experts. In Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, pages 256–263, Cham, 2015. Springer International Publishing.

[GR04] G. Grigoryan and P. Rheingans. Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):564–573, September 2004.

[GS06] Henning Griethe and Heidrun Schumann. The Visualization of Uncertain Data: Methods and Problems. In *Simulation und Visualisierung*, 2006.

[GSN+18] Erik Garrison, Jouni Sirén, Adam M. Novak, Glenn Hickey, Jordan M. Eizenga, Eric T. Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F. Lin, Benedict Paten, and Richard Durbin. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, 36(9):875–879, 2018.

[GST+14] Mahdieh Ghasemi, Hossein Seidkhani, Faezeh Tamimi, Maseud Rahgozar, and Ali Masoudi-Nejad. Centrality Measures in Biological Networks. *Current Bioinformatics*, 9(4):426–441, September 2014.

[GT94] Olivier Goldschmidt and Alexan Takvorian. An efficient graph planarization two-phase heuristic. *Networks*, 24(2):69–73, 1994.

[HAP+19] Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N. Mendoza, Anne Richelle, Almut Heinken, Hulda S. Haraldsdóttir, Jacek Wachowiak, Sarah M. Keating, Vanja Vlasov, Stefania Magnusdóttir, Chiam Yu Ng, German Preciat, Alise Žagare, Siu H. J. Chan, Maike K. Aurich, Catherine M. Clancy, Jennifer Modamio, John T. Sauls, Alberto Noronha, Aarash Bordbar, Benjamin Cousins, Diana C. El Assal, Luis V. Valcarcel, Iñigo Apaolaza, Susan Ghaderi, Masoud Ahookhosh, Marouen Ben Guebila, Andrejs Kostromins, Nicolas Sompairac, Hoai M. Le, Ding Ma, Yuekai Sun, Lin Wang, James T. Yurkovich, Miguel A. P. Oliveira, Phan T. Vuong, Lemmer P. El Assal, Inna Kuperstein, Andrei Zinovyev, H. Scott Hinton, William A. Bryant, Francisco J. Aragón Artacho, Francisco J. Planes, Egils Stalidzans, Alejandro Maass, Santosh Vempala, Michael Hucka, Michael A. Saunders, Costas D. Maranas, Nathan E. Lewis, Thomas Sauter, Bernhard Ø Palsson, Ines Thiele, and Ronan M. T. Fleming. Creation and analysis of biochemical constraint-based

models using the COBRA Toolbox v.3.0. *Nature Protocols*, 14(3):639–702, March 2019.

[HB03] Mark Harrower and Cynthia A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, June 2003.

[HBF08] Nathalie Henry, Anastasia Bezerianos, and Jean-Daniel Fekete. Improving the Readability of Clustered Social Networks using Node Duplication. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1317–1324, November 2008.

[HBF+16] Daniel H. Huson, Sina Beier, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans-Joachim Ruscheweyh, and Rewati Tappu. MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data. *PLOS Computational Biology*, 12(6):e1004957, 2016.

[HBVV12] Pedro Hermosilla, Ralph Brecheisen, Pere-Pau Vázquez, and Anna Vilanova. Uncertainty Visualization of Brain Fibers, 2012.

[HBW14] Marcel Hlawatsch, Michael Burch, and Daniel Weiskopf. Visual Adjacency Lists for Dynamic Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 20(11):1590–1603, November 2014.

[HCG17] Sijia Huang, Kumardeep Chaudhary, and Lana X. Garmire. More Is Better: Recent Progress in Multi-Omics Data Integration Methods. *Frontiers in Genetics*, 8, 2017.

[HCH+21] Jingming Hu, Tuan Tran Chu, Seok-Hee Hong, Jialu Chen, Amyra Meidiana, Marnijati Torkel, Peter Eades, and Kwan-Liu Ma. BC tree-based spectral sampling for big complex network visualization. *Applied Network Science*, 6(1):60, December 2021.

[HdDTMM+18] Rafael Hernández-de Diego, Sonia Tarazona, Carlos Martínez-Mira, Leandro Balzano-Nogueira, Pedro Furió-Tarí, Georgios J Pappas, Jr, and Ana Conesa. PaintOmics 3: a web resource for the pathway analysis and visualization of multi-omics data. *Nucleic Acids Research*, 46(W1):W503–W509, July 2018.

[Hea96] C.G. Healey. Choosing effective colours for data visualization. In *Proceedings of Seventh Annual IEEE Visualization '96*, pages 263–270, October 1996.

[HEAE16] Shah Rukh Humayoun, Hafez Ezaiza, Ragaad AlTarawneh, and Achim Ebert. Social-Circles Exploration through Interactive Multi-Layered Chord Layout. In *Proceedings of the International Working Conference*

on Advanced Visual Interfaces, AVI '16, pages 314–315, New York, NY, USA, June 2016. Association for Computing Machinery.

[HEHL13] Weidong Huang, Peter Eades, Seok-Hee Hong, and Chun-Cheng Lin. Improving multiple aesthetics produces better graph drawings. *Journal of Visual Languages & Computing*, 24(4):262–272, August 2013.

[Hen03] Tomislav Hengl. Visualisation of uncertainty using the HSI colour model: computations with colours. 2003.

[HF07] Nathalie Henry and Jean-Daniel Fekete. MatLink: Enhanced Matrix Visualization for Analyzing Social Networks. In Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa, editors, *Human-Computer Interaction – INTERACT 2007*, Lecture Notes in Computer Science, pages 288–302, Berlin, Heidelberg, 2007. Springer.

[HFM07] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, November 2007.

[HFS+03] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, March 2003.

[HHE06] Weidong Huang, Seok-Hee Hong, and Peter Eades. Layout Effects on Sociogram Perception. In Patrick Healy and Nikola S. Nikolov, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 262–273, Berlin, Heidelberg, 2006. Springer.

[HHLM99] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl):C47–52, December 1999.

[HHW+09] Zhenjun Hu, Jui-Hung Hung, Yan Wang, Yi-Chien Chang, Chia-Ling Huang, Matt Huyck, and Charles DeLisi. VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. *Nucleic Acids Research*, 37(suppl_2):W115–W121, July 2009.

[HJ18] Anja Hartmann and Anna Maria Jozefowicz. VANTED: A tool for integrative visualization and analysis of -omics data. *Methods Mol. Biol.*, 1696:261–278, 2018.

[Hli06] Petr Hliněný. Crossing number is hard for cubic graphs. *Journal of Combinatorial Theory, Series B*, 96(4):455–471, July 2006.

[HM96] David Howard and Alan M. MacEachren. Interface Design for Geographic Visualization: Tools for Representing Reliability. *Cartography and Geographic Information Systems*, 23(2):59–77, January 1996.

[HM21] Joel Hancock and Jörg Menche. Structure and Function in Complex Biological Networks. In Olaf Wolkenhauer, editor, *Systems Medicine*, pages 8–16. Academic Press, Oxford, 2021.

[HMB+18] James Hadfield, Colin Megill, Sidney M Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A Neher. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123, 2018.

[HMPI+16] Bernie Hogan, Joshua R. Melville, Gregory Lee Phillips II, Patrick Janulis, Noshir Contractor, Brian S. Mustanski, and Michelle Birkett. Evaluating the Paper-to-Screen Translation of Participant-Aided Sociograms with High-Risk Participants. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5360–5371, New York, NY, USA, May 2016. Association for Computing Machinery.

[HMT+17] Chen He, Luana Micallef, Zia-ur-Rehman Tanoli, Samuel Kaski, Tero Aittokallio, and Giulio Jacucci. MediSyn: uncertainty-aware visualization of multiple biomedical datasets to support drug treatment selection. *BMC Bioinformatics*, 18(10):393, September 2017.

[HN14] Patrick Healy and Nikola S. Nikolov. Hierarchical Drawing Algorithms. In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*. CRC Press, 2014.

[HQC+19] Jessica Hullman, Xiaoli Qiao, Michael Correll, Alex Kale, and Matthew Kay. In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):903–913, January 2019.

[HR07] Jeffrey Heer and George Robertson. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, November 2007.

[HS88] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Peter A. Hancock and Najmedin Meshkati, editors, *Advances in Psychology*, volume 52 of *Human Mental Workload*, pages 139–183. North-Holland, January 1988.

[HS15] Anja Hartmann and Falk Schreiber. Integrative Analysis of Metabolic Models – from Structure to Dynamics. *Frontiers in Bioengineering and Biotechnology*, 2, 2015.

[HSL17] Yehudit Hasin, Marcus Seldin, and Aldons Lusis. Multi-omics approaches to disease. *Genome Biology*, 18(1):83, May 2017.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[Hu06] Yifan Hu. Efficient, High-Quality Force-Directed Graph Drawing. *Mathematica Journal*, 10(1):37–71, 2006.

[Hul20] Jessica Hullman. Why Authors Don't Visualize Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):130–139, January 2020.

[HVCM17] Jamie Herring, Matthew S. VanDyke, R. Glenn Cummins, and Forrest Melton. Communicating Local Climate Risks Online Through an Interactive Data Visualization. *Environmental Communication*, 11(1):90–105, January 2017.

[HWM+15] Jenni Hultman, Mark P. Waldrop, Rachel Mackelprang, Maude M. David, Jack McFarland, Steven J. Blazewicz, Jennifer Harden, Merritt R. Turetsky, A. David McGuire, Manesh B. Shah, Nathan C. VerBerkmoes, Lang Ho Lee, Kostas Mavrommatis, and Janet K. Jansson. Multi-omics of permafrost, active layer and thermokarst bog soil microbiomes. *Nature*, 521(7551):208–212, May 2015.

[HZL+16] Qinglai He, Min Zhu, Binbin Lu, Hanqing Liu, and Qiaomu Shen. MENA: Visual Analysis of Multivariate Egocentric Network Evolution. In *2016 International Conference on Virtual Reality and Visualization (ICVRV)*, pages 488–496, September 2016.

[Jai10] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, June 2010.

[JBIH05] Maliackal Poulo Joy, Amy Brock, Donald E. Ingber, and Sui Huang. High-Betweenness Proteins in the Yeast Protein Interaction Network. *Journal of Biomedicine and Biotechnology*, 2005(2):96–103, 2005.

[JBN12] Günter Jäger, Florian Battke, and Kay Nieselt. Reveal—visual eQTL analytics. *Bioinformatics*, 28(18):i542–i548, 2012.

[JdKW21] Bart M. P. Jansen, Jari J. H. de Kroon, and Michał Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1757–1769, New York, NY, USA, June 2021. Association for Computing Machinery.

[JED+20] Amit Jena, Ulrich Engelke, Tim Dwyer, Venkatesh Raiamanickam, and Cecile Paris. Uncertainty Visualisation: An Interactive Visual Survey. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–205, June 2020.

[JGH11] Yuntao Jia, Michael Garland, and John C. Hart. Social Network Clustering and Visualization using Hierarchical Edge Bundles. *Computer Graphics Forum*, 30(8):2314–2327, 2011.

[JJHS+22] Lucas Joos, Sabrina Jaeger-Honz, Falk Schreiber, Daniel A. Keim, and Karsten Klein. Visual Comparison of Networks in VR. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3651–3661, November 2022.

[JKS06] Björn H. Junker, Dirk Koschützki, and Falk Schreiber. Exploration of biological network centralities with CentiBiN. *BMC Bioinformatics*, 7(1):219, April 2006.

[JKZ13] Ilir Jusufi, Andreas Kerren, and Björn Zimmer. Multivariate Network Exploration with JauntyNets. In *2013 17th International Conference on Information Visualisation*, pages 19–27, July 2013.

[JLS13] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A Near-Optimal Planarization Algorithm. In *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1802–1811. Society for Industrial and Applied Mathematics, December 2013.

[JLW+20] Yuexu Jiang, Yanchun Liang, Duolin Wang, Dong Xu, and Trupti Joshi. A dynamic programing approach to integrate gene expression data and network information for pathway model generation. *Bioinformatics*, 36(1):169–176, January 2020.

[JM97] Michael Jünger and Petra Mutzel. 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25, 1997.

[JOK95] Bin Jiang, Ormeling Ormeling, and Wolfang Kainz. Visualization Support for Fuzzy Spatial Analysis. In *Auto Carto 12, Charlotte, North Carolina*, pages 291–300, 1995.

[JSS+19] Libo Jiang, Hexin Shi, Mengmeng Sang, Chenfei Zheng, Yige Cao, Xuli Zhu, Xiaokang Zhuo, Tangren Cheng, Qixiang Zhang, Rongling Wu, and Lidan Sun. A Computational Model for Inferring QTL Control Networks Underlying Developmental Covariation. *Frontiers in Plant Science*, 10:1557, 2019.

[JSWD09] Ming Jia, Sivakumar Swaminathan, Eve Syrkin Wurtele, and Julie Dickerson. MetNetGE: Visualizing biological networks in hierarchical views and 3D tiered layouts. In *2009 IEEE International Conference on Bioinformatics and Biomedicine Workshop*, pages 287–294, November 2009.

[JSYG+16] Mahdi Jalili, Ali Salehzadeh-Yazdi, Shailendra Gupta, Olaf Wolkenhauer, Marjan Yaghmaie, Osbaldo Resendis-Antonio, and Kamran Alimoghaddam. Evolution of Centrality Measurements for the Detection of Essential Proteins in Biological Networks. *Frontiers in Physiology*, 7, 2016.

[JYC+10] Radu Jianu, Kebing Yu, Lulu Cao, Vinh Nguyen, Arthur R. Salomon, and David H. Laidlaw. Visual Integration of Quantitative Proteomic Data, Pathways, and Protein Interactions. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):609–620, July 2010.

[KAB+14] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, September 2014.

[Kan19] Minoru Kanehisa. Toward understanding the origin and evolution of cellular organisms. *Protein Science: A Publication of the Protein Society*, 28(11):1947–1951, November 2019.

[KBC+19] Peter D. Karp, Richard Billington, Ron Caspi, Carol A. Fulcher, Mario Latendresse, Anamika Kothari, Ingrid M. Keseler, Markus Krummenacker, Peter E. Midford, Quang Ong, Wai Kit Ong, Suzanne M. Paley, and Pallavi Subhraveti. The BioCyc collection of microbial genomes and metabolic pathways. *Briefings in Bioinformatics*, 20(4):1085–1093, July 2019.

[KBJM12] Martin Krzywinski, Inanc Birol, Steven JM Jones, and Marco A Marra. Hive plots—rational approach to visualizing networks. *Briefings in Bioinformatics*, 13(5):627–644, September 2012.

[KBT+10] Vivek Kaimal, Eric E. Bardes, Scott C. Tabar, Anil G. Jegga, and Bruce J. Aronow. ToppCluster: a multiple gene list feature analyzer for comparative enrichment clustering and network-based dissection of biological systems. *Nucleic Acids Research*, 38(suppl_2):W96–W102, July 2010.

[KCK17] Kyungyoon Kim, John V. Carlis, and Daniel F. Keefe. Comparison techniques utilized in spatial 3D and 4D data visualizations: A survey and future directions. *Computers & Graphics*, 67:138–147, October 2017.

[KDJ+21] Aasim Kamal, Parashar Dhakal, Ahmad Y. Javaid, Vijay K. Devabhaktuni, Devinder Kaur, Jack Zaientz, and Robert Marinier. Recent advances and challenges in uncertainty visualization: a survey. *Journal of Visualization*, 24(5):861–890, October 2021.

[KDSM20] Michal Krassowski, Vivek Das, Sangram K. Sahu, and Biswapriya B. Misra. State of the Field in Multi-Omics Research: From Computational Needs to Data Mining and Sharing. *Frontiers in Genetics*, 11, December 2020.

[KDT14] Ilias Kalamaras, Anastasios Drosou, and Dimitrios Tzovaras. Multi-Objective Optimization for Multimodal Visualization. *IEEE Transactions on Multimedia*, 16(5):1460–1472, August 2014.

[KEC06] René Keller, Claudia M. Eckert, and P. John Clarkson. Matrices or Node-Link Diagrams: Which Visual Representation is Better for Visualising Connectivity Models? *Information Visualization*, 5(1):62–76, March 2006.

[Kei02] D.A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, March 2002.

[KFS+22] Minoru Kanehisa, Miho Furumichi, Yoko Sato, Masayuki Kawashima, and Mari Ishiguro-Watanabe. KEGG for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Research*, page gkac963, October 2022.

[KFW08] Kari Kelton, Kenneth R. Fleischmann, and William A. Wallace. Trust in digital information. *Journal of the American Society for Information Science and Technology*, 59(3):363–374, 2008.

[KG00] M. Kanehisa and S. Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, January 2000.

[KK89] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, April 1989.

[KKC15] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. A Task Taxonomy for Temporal Graph Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 21(10):1160–1172, October 2015.

[KKK17] Sungjin Kwon, Hyosil Kim, and Hyun Seok Kim. Identification of Pharmacologically Tractable Protein Complexes in Cancer Using the R-Based Network Clustering and Visualization Program MCODER. *BioMed Research International*, 2017:1016305, June 2017.

[KKPEP20] Mikaela Koutrouli, Evangelos Karatzas, David Paez-Espino, and Georgios A. Pavlopoulos. A Guide to Conquer the Biological Network Era Using Graph Theory. *Frontiers in Bioengineering and Biotechnology*, 8:34, 2020.

[KKPP21] Mikaela Koutrouli, Evangelos Karatzas, Katerina Papanikolopoulou, and Georgios A. Pavlopoulos. NORMA: The Network Makeup Artist — A Web Tool for Network Annotation Visualization. *Genomics, Proteomics & Bioinformatics*, June 2021.

[KKSZ11] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[KKZ12] Andreas Kerren, Harald Köstinger, and Björn Zimmer. VINCENT - Visualization of Network Centralities. In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications (VISIGRAPP 2012) - IVAPP*, pages 703–712, 2012.

[KLKP15] Min-Seok Kwon, Sungyoung Lee, Yongkang Kim, and Taesung Park. VizEpis : A visualization and mapping tool for interpreting epistasis. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1363–1366, Washington, DC, USA, November 2015. IEEE.

[KM17] Aziz Khan and Anthony Mathelier. Intervene: a tool for intersection and visualization of multiple gene or genomic region sets, March 2017.

[KMMP08] Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna E. Paluch. An (O^2n) Algorithm for Minimum Cycle Basis of Graphs. *Algorithmica*, 52(3):333–349, November 2008.

[KN17] Bisma S Khan and Muaz A Niazi. Network Community Detection: A Review and Visual Survey. Technical report, Department of Computer Science, COMSATS Institute of Information Technology, Islamabad, Pakistan, 2017.

[Kni08] Joe Michael Kniss. Managing uncertainty in visualization and analysis of medical data. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 832–835, May 2008.

[KNKH19] Alex Kale, Francis Nguyen, Matthew Kay, and Jessica Hullman. Hypothetical Outcome Plots Help Untrained Observers Judge Trends in Ambiguous Data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):892–902, January 2019.

[Knu93] Knuth, Donald Ervin. *The Stanford GraphBase: a platform for combinatorial computing*, volume 1. AcM Press New York, 1993.

[Kob12] Stephen G. Kobourov. Spring Embedders and Force Directed Graph Drawing Algorithms, January 2012.

[Kob14] Stephen Kobourov. Force-Directed Drawing Algorithms. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*. CRC Press, 2014.

[Kof35] K. Koffka. *Principles of Gestalt psychology*. Principles of Gestalt psychology. Harcourt, Brace, Oxford, England, 1935. Pages: 720.

[KPS14] Stephen G. Kobourov, Sergey Pupyrev, and Bahador Saket. Are Crossings Important for Drawing Large Graphs? In Christian Duncan and Antonios Symvonis, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 234–245, Berlin, Heidelberg, 2014. Springer.

[KR90] L. Kaufman and P. Rousseeuw. Agglomerative Nesting (Program AGNES). In *Finding Groups in Data*, Wiley Series in Probability and Statistics, pages 199–252. John Wiley & Sons, Ltd, 1990.

[KRM+17] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. Graph Layouts by t-SNE. *Computer Graphics Forum*, 36(3):283–294, June 2017.

[KS08a] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews. Molecular Cell Biology*, 9(10):770–780, October 2008.

[KS08b] Dirk Koschützki and Falk Schreiber. Centrality Analysis Methods for Biological Networks and Their Application to Gene Regulatory Networks. *Gene Regulation and Systems Biology*, 2:GRSB.S702, January 2008.

[KSB+09] Martin Krzywinski, Jacqueline Schein, İnanç Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J. Jones, and Marco A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, September 2009.

[KSP23] Bharat Kale, Maoyuan Sun, and Michael E. Papka. The State of the Art in Visualizing Dynamic Multivariate Networks. *Computer Graphics Forum*, 42(3):471–490, 2023.

[KSV+16] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. DELTACON: Principled Massive-Graph Similarity Function with Attribution. In *ACM Transactions on Knowledge Discovery from Data*, volume 10, pages 1–43, February 2016.

[KTT13] Tien-Chueh Kuo, Tze-Feng Tian, and Yufeng Jane Tseng. 3Omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. *BMC Syst. Biol.*, 7(1):64, July 2013.

[KVIB+15] Martina Kutmon, Martijn P. Van Iersel, Anwesha Bohler, Thomas Kelder, Nuno Nunes, Alexander R. Pico, and Chris T. Evelo. PathVisio 3: An Extendable Pathway Analysis Toolbox. *PLOS Computational Biology*, 11(2):e1004085, February 2015.

[KWKJ19] T J M Kuijpers, J E J Wolters, J C S Kleinjans, and D G J Jennen. DynOVis: a web tool to study dynamic perturbations for capturing dose-over-time effects in biological networks. *BMC Bioinformatics*, 20(1):417, August 2019.

[LB21] Ivica Letunic and Peer Bork. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. *Nucleic Acids Research*, 49(W1):W293–W296, July 2021.

[LBI+11] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Seven Guiding Scenarios for Information Visualization Evaluation. [Research Report] 2011-992-04 hal- 00723057, French Institute for Research in Computer Science and Automation (INRIA), 2011.

[LCBD11] K. Lowell, B. Christy, K. Benke, and G. Day. Modelling fundamentals and the quantification and spatial presentation of uncertainty. *Journal of Spatial Science*, 56(2):185–201, December 2011.

[LDB11] A. Lambert, J. Dubois, and R. Bourqui. Pathway Preserving Representation of Metabolic Networks. *Computer Graphics Forum*, 30(3):1021–1030, 2011.

[LDW18] Qun Liu, Zhishan Dong, and En Wang. Cut Based Method for Comparing Complex Networks. *Scientific Reports*, 8(1):5134, March 2018.

[LG14] Alexander Lex and Nils Gehlenborg. Sets and intersections. *Nature Methods*, 11(8):779–779, August 2014.

[LGD+17] Dongyu Liu, Fangzhou Guo, Bowen Deng, Huamin Qu, and Yingcai Wu. egoComp: A node-link-based technique for visual comparison of ego-networks. *Information Visualization*, 16(3):179–189, July 2017.

[LGH+17] Yafeng Lu, Rolando Garcia, Brett Hansen, Michael Gleicher, and Ross Maciejewski. The State-of-the-Art in Predictive Visual Analytics. *Computer Graphics Forum*, 36(3):539–562, 2017.

[LGS+14] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, December 2014.

[LH08] Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, December 2008.

[LHG+20] Qiang Lu, Jing Huang, Yifan Ge, Dajiu Wen, Bin Chen, and Ye Yu. EgoVis: A Visual Analysis System for Social Networks Based on Egocentric Research. *International Journal of Cooperative Information Systems*, 29(01n02):1930003, March 2020.

[LHLW19] Chun-Cheng Lin, Weidong Huang, Wan-Yu Liu, and Sheng-Feng Wu. A novel centrality-based method for visual analytics of small-world networks. *Journal of Visualization*, 22(5):973–990, October 2019.

[LHS+15] Qingsong Liu, Yifan Hu, Lei Shi, Xinzhu Mu, Yutao Zhang, and Jie Tang. EgoNetCloud: Event-based egocentric dynamic network visualization. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 65–72, October 2015.

[Lii10] Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(2):70–91, 2010.

[LJT21] Jing Liu, Ling Jing, and Xilin Tu. Weighted gene co-expression network analysis identifies specific modules and hub genes related to coronary artery disease. *Sci. Rep.*, 11(1):6711, March 2021.

[LLPY07] Claes Lundström, Patric Ljung, Anders Persson, and Anders Ynnerman. Uncertainty Visualization in Medical Volume Rendering Using Probabilistic Animation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1648–1655, November 2007.

[LLT+17] Min Li, Dongyan Li, Yu Tang, Fangxiang Wu, and Jianxin Wang. CytoCluster: A Cytoscape Plugin for Cluster Analysis and Visualization of Biological Networks. *International Journal of Molecular Sciences*, 18(9):1880, September 2017.

[LMH+10] Gabriele Lohmann, Daniel S. Margulies, Annette Horstmann, Burkhard Pleger, Joeran Lepsien, Dirk Goldhahn, Haiko Schloegl, Michael Stumvoll, Arno Villringer, and Robert Turner. Eigenvector Centrality Mapping for Analyzing Connectivity Patterns in fMRI Data of the Human Brain. *PLOS ONE*, 5(4):e10232, April 2010.

[LOP+18] Alessandra Livigni, Laura O'Hara, Marta E Polak, Tim Angus, Derek W Wright, Lee B Smith, and Tom C Freeman. A graphical and computational modeling platform for biological pathways. *Nat. Protoc.*, 13(4):705–722, April 2018.

[LPK+13] Alexander Lex, Christian Partl, Denis Kalkofen, Marc Streit, Samuel Gratzl, Anne Mai Wassermann, Dieter Schmalstieg, and Hanspeter Pfister. Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2536–2545, December 2013.

[LPP+06a] Bongshin Lee, C.S. Parr, C. Plaisant, B.B. Bederson, V.D. Veksler, W.D. Gray, and C. Kotfila. TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, November 2006.

[LPP+06b] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, BELIV '06, pages 1–5, New York, NY, USA, May 2006. Association for Computing Machinery.

[LPP+19] Claudio D. G. Linhares, Jean R. Ponciano, Jose Gustavo S. Paiva, Bruno A. N. Travençolo, and Luis E. C. Rocha. Visualisation of Structure and Processes on Temporal Networks. In Petter Holme and Jari Saramäki, editors, *Temporal Network Theory*, pages 83–105. Springer International Publishing, Cham, 2019.

[LRK18] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph Classification using Structural Attention. In *Proceedings of the 24th ACM SIGKDD*

International Conference on Knowledge Discovery & Data Mining, pages 1666–1674, London United Kingdom, July 2018. ACM.

[LSDK18] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph Summarization Methods and Applications: A Survey. *ACM Computing Surveys*, 51(3):62:1–62:34, June 2018.

[LSKS10] Alexander Lex, Marc Streit, Ernst Kruijff, and Dieter Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, March 2010.

[LSM+17] Quan Li, Qiaomu Shen, Yao Ming, Peng Xu, Yun Wang, Xiaojuan Ma, and Huamin Qu. A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in MMORPGs. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 31–40, April 2017. ISSN: 2165-8773.

[LT20] Jean-Baptiste Lamy and Rosy Tsopra. RainBio: Proportional Visualization of Large Sets in Biology. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3285–3298, November 2020.

[LVJV03] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, February 2003.

[LWB18] Po-Ming Law, Yanhong Wu, and Rahul C. Basole. Segue: Overviewing Evolution Patterns of Egocentric Networks by Interactive Construction of Spatial Layouts. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 72–83, October 2018.

[LZ13] JingRon Li and WenJun Zhang. Identification of crucial metabolites/reactions in tumor signaling networks. *Network Biology*, 3(4):13, 2013.

[LZH+17] B. Lu, M. Zhu, Q. He, M. Li, and R. Jia. TMNVis: Visual analysis of evolution in temporal multivariate network at multiple granularities. *Journal of Visual Languages & Computing*, 43:30–41, December 2017.

[MAR+21] Marvin Martens, Ammar Ammar, Anders Riutta, Andra Waagmeester, Denise N Slenter, Kristina Hanspers, Ryan A. Miller, Daniela Digles, Elisson N Lopes, Friederike Ehrhart, Lauren J Dupuis, Laurent A Winckers, Susan L Coort, Egon L Willighagen, Chris T Evelo, Alexander R Pico, and Martina Kutmon. WikiPathways: connecting communities. *Nucleic Acids Research*, 49(D1):D613–D621, January 2021.

[MAU15] Rodrigo Mesa-Arango and Satish V. Ukkusuri. Demand clustering in freight logistics networks. *Transportation Research Part E: Logistics and Transportation Review*, 81:36–51, September 2015.

[MBAR13] Atieh Mirshahvalad, Olivier H. Beauchesne, Éric Archambault, and Martin Rosvall. Resampling Effects on Significance Analysis of Network Clustering and Ranking. *PLOS ONE*, 8(1):e53943, January 2013.

[MBSB10] Aurélien Mazurie, Danail Bonchev, Benno Schwikowski, and Gregory A. Buck. Evolution of metabolic network organization. *BMC Systems Biology*, 4(1):59, May 2010.

[MCC+20] Ji Ma, Jinjin Chen, Liye Chen, Linjiang Jin, and Xujia Qin. Dynamic Visualization of Uncertainties in Medical Feature of Interest. *IEEE Access*, 8:119170–119183, 2020.

[MCER25] M. Musleh, D. Ceneda, H. Ehlers, and R. G. Raidou. ConAn: Measuring and Evaluating User Confidence in Visual Data Analysis Under Uncertainty. *Computer Graphics Forum*, 44(1):e15272, 2025.

[MCK13] Danielle P. Mersch, Allessandro Crespi, and Laurent Keller. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 340(6136):1090–1093, 2013.

[MELS95] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout Adjustment and the Mental Map. *Journal of Visual Languages & Computing*, 6(2):183–210, June 1995.

[MEM+25] Kevin Mildau, Henry Ehlers, Mara Meisenburg, Elena Del Pup, Robert A. Koetsier, Laura Rosina Torres Ortega, Niek F. de Jonge, Kumar Saurabh Singh, Dora Ferreira, Kgalaletso Othibeng, Fidele Tugizimana, Florian Huber, and Justin J. J. van der Hooft. Effective data visualization strategies in untargeted metabolomics. *Natural Product Reports*, 2025.

[MEO+24] Kevin Mildau, Henry Ehlers, Ian Oesterle, Manuel Pristner, Benedikt Warth, Maria Doppler, Christoph Bueschl, Jürgen Zanghellini, and Justin J. J. van der Hooft. Tailored Mass Spectral Data Exploration Using the SpecXplore Interactive Dashboard. *Analytical Chemistry*, 96(15):5798–5806, April 2024.

[MFA18] Md Abdul Motaleb Faysal and Shaikh Arifuzzaman. A Comparative Analysis of Large-scale Network Visualization Tools. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4837–4843, December 2018.

[MGM+19] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, and B. Pinaud. The State of the Art in Multilayer Network Visualization. *Computer Graphics Forum*, 38(6):125–149, 2019.

[MGO+21] Fintan McGee, Mohammad Ghoniem, Benoît Otjacques, Benjamin Renoust, Daniel Archambault, Andreas Kerren, Bruno Pinaud, Guy

Melançon, Margit Pohl, and Tatiana Von Landesberger. *Visual Analysis of Multilayer Networks*. Synthesis Lectures on Visualization. Springer International Publishing, Cham, 2021.

[MGT+03] Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 453–462, New York, NY, USA, July 2003. Association for Computing Machinery.

[MHBG16] Grant McKenzie, Mary Hegarty, Trevor Barrett, and Michael Goodchild. Assessing the effectiveness of different visualizations for judgments of positional uncertainty. *International Journal of Geographical Information Science*, 30(2):221–239, February 2016.

[MHSW19] Eva Mayr, Nicole Hynek, Saminu Salisu, and Florian Windhager. Trust in Information Visualization. *EuroVis Workshop on Trustworthy Visualization (TrustVis)*, page 5 pages, 2019.

[Mic17] Gerhard Michal. Roche - Biochemical Pathways, 2017.

[MKF+15] Chihua Ma, Robert V. Kenyon, Angus G. Forbes, Tanya Berger-Wolf, Bernard J. Slater, and Daniel A. Llano. Visualizing Dynamic Brain Networks Using an Animated Dual-Representation. In *EuroVis Short Papers*, pages 73–77, 2015.

[MLM+17] Zakhar Sergeevich Mustafin, Sergey Alexandrovich Lashin, Yury Georgievich Matushkin, Konstantin Vladimirovich Gunbin, and Dmitry Arkadievich Afonnikov. Orthoscape: a cytoscape application for grouping and visualization KEGG based gene networks by taxonomy and homology principles. *BMC Bioinformatics*, 18(1):1–9, January 2017.

[MMB05] James Moody, Daniel McFarland, and Skye Bender-deMoll. Dynamic Network Visualization. *American Journal of Sociology*, 110(4):1206–1241, January 2005.

[MMF17] Paul Murray, Fintan McGee, and Angus G. Forbes. A taxonomy of visualization tasks for the analysis of biological pathway data. *BMC Bioinformatics*, 18(2):21, February 2017.

[MML07] Christopher Mueller, Benjamin Martin, and Andrew Lumsdaine. A comparison of vertex ordering algorithms for large graph visualization. In *2007 6th International Asia-Pacific Symposium on Visualization*, pages 141–148, February 2007.

[MMTM11] Robert Miller, Vadim Mozhayskiy, Llias Tagkopoulos, and Kwan-Liu Ma. EVEVis: A multi-scale visualization system for dense evolutionary data. In *2011 IEEE Symposium on Biological Data Visualization (BioVis).*, pages 143–150, October 2011.

[MOB+12] John P. McIntire, O. Isaac Osesina, Cecilia Bartley, M. Eduard Tudoreanu, Paul R. Havig, and Eric E. Geiselman. Visualizing weighted networks: a performance comparison of adjacency matrices versus node-link diagrams. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*, pages 83891G–83891G–7, Baltimore, Maryland, May 2012.

[Mor12] Kenneth Moreland. A survey of visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):367–378, 2012.

[MP08] Tijana Milenković and Nataša Pržulj. Uncovering Biological Network Function via Graphlet Degree Signatures. *Cancer Informatics*, 6:CIN.S680, January 2008.

[MRH+05] Alan M. MacEachren, Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elisabeth Hetzler. Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know. *Cartography and Geographic Information Science*, 32(3):139–160, January 2005.

[MRO+12] Alan M. MacEachren, Robert E. Roth, James O'Brien, Bonan Li, Derek Swingley, and Mark Gahegan. Visual Semiotics & Uncertainty Visualization: An Empirical Study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2496–2505, December 2012.

[MRS+13] Wouter Meulemans, Nathalie Henry Riche, Bettina Speckmann, Basak Alper, and Tim Dwyer. KelpFusion: A Hybrid Set Visualization Technique. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1846–1858, November 2013.

[MS15] Wouter Meulemans and André Schulz. A Tale of Two Communities: Assessing Homophily in Node-Link Diagrams. In Emilio Di Giacomo and Anna Lubiw, editors, *Graph Drawing and Network Visualization*, Lecture Notes in Computer Science, pages 489–501, Cham, 2015. Springer International Publishing.

[MSOI+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science (New York, N.Y.)*, 298(5594):824–827, October 2002.

[MSX+02] Candido Mendonça, Karl Schaffer, Erico Xavier, Jorge Stolfi, Luerbio Faria, and Celina Figueiredo. The Splitting Number and Skewness of Cn × Cm. *Ars Comb.*, 63, April 2002.

[Mun14] Tamara Munzner. *Visualization Analysis and Design*. A K Peters/CRC Press, New York, October 2014.

[MYH+19] Liu Miao, Rui-Xing Yin, Feng Huang, Shuo Yang, Wu-Xian Chen, and Jin-Zhen Wu. Integrated analysis of gene expression changes associated with coronary artery disease. *Lipids Health Dis.*, 18(1):92, April 2019.

[MZ03] Hong-Wu Ma and An-Ping Zeng. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics*, 19(11):1423–1430, July 2003.

[MZ11] Kazuo Misue and Qi Zhou. Drawing Semi-bipartite Graphs in Anchor+Matrix Style. In *2011 15th International Conference on Information Visualisation*, pages 26–31, July 2011.

[NBKM20] Sunil Nagpal, Krishanu Das Baksi, Bhusan K. Kuntal, and Sharmila S. Mande. NetConfer: a web application for comparative analysis of multiple biological networks. *BMC Biology*, 18(1):53, May 2020.

[NDG+17] Alberto Noronha, Anna Dröfn Daníelsdóttir, Piotr Gawron, Freyr Jóhannsson, Soffía Jónsdóttir, Sindri Jarlsson, Jón Pétur Gunnarsson, Sigurður Brynjólfsson, Reinhard Schneider, Ines Thiele, and Ronan M T Fleming. ReconMap: an interactive visualization of human metabolism. *Bioinformatics*, 33(4):605–607, February 2017.

[NDP+21] Hiroaki Natsukawa, Ethan R. Deyle, Gerald M. Pao, Koji Koyamada, and George Sugihara. A Visual Analytics Approach for Ecosystem Dynamics based on Empirical Dynamic Modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):506–516, February 2021.

[New10] M. E. J. Newman. Measures and metrics: An introduction to some standard measures and metrics for quantifying network structure, many of which were introduced first in the study of social networks, although they are now in wide use in many other areas. In Mark Newman, editor, *Networks: An Introduction*, page 0. Oxford University Press, March 2010.

[NHEM17] Quan Hoang Nguyen, Seok-Hee Hong, Peter Eades, and Amyra Meidiana. Proxy Graph: Visual Quality Metrics of Big Graph Sampling. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1600–1611, June 2017.

[Nis84] Y. Nishizuka. The role of protein kinase C in cell surface signal transduction and tumour promotion. *Nature*, 308(5961):693–698, April 1984.

[NIST12] Rina Nakazawa, Takayuki Itoh, Jun Sese, and Aika Terada. Integrated Visualization of Gene Network and Ontology Applying a Hierarchical Graph Visualization Technique. In *2012 16th International Conference on Information Visualisation*, pages 81–86, July 2012.

[NKHC08] Joshua New, Wesley Kendall, Jian Huang, and Elissa Chesler. Dynamic Visualization of Coexpression in Systems Genetics Data. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1081–1095, September 2008.

[NMH+18] Christian F. A. Negre, Uriel N. Morzan, Heidi P. Hendrickson, Rhitankar Pal, George P. Lisi, J. Patrick Loria, Ivan Rivalta, Junming Ho, and Victor S. Batista. Eigenvector centrality for characterization of protein allosteric pathways. *Proceedings of the National Academy of Sciences*, 115(52):E12201–E12208, December 2018.

[NMSL19] C. Nobre, M. Meyer, M. Streit, and A. Lex. The State of the Art in Visualizing Multivariate Networks. *Computer Graphics Forum*, 38(3):807–832, 2019.

[noa14] Prolific, 2014.

[NOM+21] Sune S. Nielsen, Marek Ostaszewski, Fintan McGee, David Hoksza, and Simone Zorzan. Machine Learning to Support the Presentation of Complex Pathway Graphs. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(3):1130–1141, June 2021.

[NST+22] Martin Nöllenburg, Manuel Sorge, Soeren Terziadis, Anaïs Villedieu, Hsiang-Yun Wu, and Jules Wulms. Planarizing Graphs and their Drawings by Vertex Splitting, 2022.

[NVR14] Alberto Noronha, Paulo Vilaça, and Miguel Rocha. An integrated network visualization framework towards metabolic engineering applications. *BMC Bioinformatics*, 15(1):420, December 2014.

[NWHL20] Carolina Nobre, Dylan Wootton, Lane Harrison, and Alexander Lex. Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, April 2020. Association for Computing Machinery.

[NYP12] Tamás Nepusz, Haiyuan Yu, and Alberto Paccanaro. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9(5):471–472, March 2012.

[OGG+10] Seán I. O'Donoghue, Anne-Claude Gavin, Nils Gehlenborg, David S. Goodsell, Jean-Karim Hériché, Cydney B. Nielsen, Chris North, Arthur J. Olson, James B. Procter, David W. Shattuck, Thomas Walter, and Bang Wong. Visualizing biological data—now and in the future. *Nature Methods*, 7(3):S2–S4, March 2010.

[OJ15a] Mershack Okoe and Radu Jianu. Ecological Validity in Quantitative User Studies – a Case Study in Graph Evaluation. In *Proc. IEEE VIS*, 2015.

[OJ15b] Mershack Okoe and Radu Jianu. GraphUnit: Evaluating Interactive Graph Visualizations Using Crowdsourcing. *Computer Graphics Forum*, 34(3):451–460, 2015.

[OJK18] Mershack Okoe, Radu Jianu, and Stephen Kobourov. Revisited Experimental Comparison of Node-Link and Matrix Representations. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization*, Lecture Notes in Computer Science, pages 287–302, Cham, 2018. Springer International Publishing.

[OJK19] Mershack Okoe, Radu Jianu, and Stephen Kobourov. Node-Link or Adjacency Matrices: Old Question, New Insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(10):2940–2952, October 2019.

[OKB17] Mark Ortmann, Mirza Klimenta, and Ulrik Brandes. A Sparse Stress Model. *Journal of Graph Algorithms and Applications*, 21(5):791–821, 2017.

[OKSK16] Yosuke Onoue, Nobuyuki Kukimoto, Naohisa Sakamoto, and Koji Koyamada. Minimizing the Number of Edges via Edge Concentration in Dense Layered Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1652–1661, June 2016.

[OSA+21] Furkan Ozden, Metin Can Siper, Necmi Acarsoy, Tugrulcan Elmas, Bryan Marty, Xinjian Qi, and A. Ercument Cicek. DORMAN: Database of Reconstructed MetAbolic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(4):1474–1480, July 2021.

[OT81] T. Ozawa and H. Takahashi. A graph-planarization algorithm and its application to random graphs. In N. Saito and T. Nishizeki, editors, *Graph Theory and Algorithms*, pages 95–107, Berlin, Heidelberg, 1981. Springer.

[OVER08] Arzucan Ozgur, Thuy Vu, Gunes Erkan, and Dragomir R. Radev. Identifying gene-disease associations using centrality on a literature

mined gene-interaction network. *Bioinformatics*, 24(13):i277–i285, July 2008.

[PBH+21] Suzanne Paley, Richard Billington, James Herson, Markus Krummenacker, and Peter D. Karp. Pathway Tools Visualization of Organism-Scale Metabolic Networks. *Metabolites*, 11(2):64, January 2021.

[PCA02] Helen C. Purchase, David Carrington, and Jo-Anne Allder. Empirical Evaluation of Aesthetics-based Graph Layout. *Empirical Software Engineering*, 7(3):233–255, September 2002.

[PCJ96] Helen C. Purchase, Robert F. Cohen, and Murray James. Validating graph drawing aesthetics. In Franz J. Brandenburg, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 435–446, Berlin, Heidelberg, 1996. Springer.

[PD17] Miriam Perkins and Karen Daniels. Visualizing Dynamic Gene Interactions to Reverse Engineer Gene Regulatory Networks Using Topological Data Analysis. In *2017 21st International Conference Information Visualisation (IV)*, pages 384–389, London, July 2017. IEEE.

[PDA10] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLOS ONE*, 5(3):e9490, October 2010.

[PDBE+25] D. Pahr, S. Di Bartolomeo, H. Ehlers, V. A. Filipov, C. Stoiber, W. Aigner, H.-Y. Wu, and R. G. Raidou. NODKANT: Exploring Constructive Network Physicalization. *Computer Graphics Forum*, n/a(n/a):e70140, 2025.

[PEF25] Daniel Pahr, Henry Ehlers, and Velitchko Filipov. HoloGraphs: An Interactive Physicalization for Dynamic Graphs, January 2025. arXiv:2502.00044 [cs].

[PF15] Francesco Paduano and Angus Graeme Forbes. Extended LineSets: a visualization technique for the interactive inspection of biological pathways. *BMC Proceedings*, 9(6):S4, August 2015.

[PFH+18] Nicola Pezzotti, Jean-Daniel Fekete, Thomas Höllt, Boudewijn P.F. Lelieveldt, Elmar Eisemann, and Anna Vilanova. Multiscale Visualization and Exploration of Large Bipartite Graphs. *Computer Graphics Forum*, 37(3):549–560, 2018.

[PHE+18] Robert Pienta, Fred Hohman, Alex Endert, Acar Tamersoy, Kevin Roundy, Chris Gates, Shamkant Navathe, and Duen Horng Chau. VIGOR: Interactive Visual Exploration of Graph Query Results. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):215–225, January 2018.

[PHR+15] Lace M. Padilla, Grace Hansen, Ian T. Ruginski, Heidi S. Kramer, William B. Thompson, and Sarah H. Creem-Regehr. The influence of different graphical displays on nonexpert decision making under uncertainty. *Journal of Experimental Psychology: Applied*, 21(1):37–46, 2015.

[PK21] Suzanne Paley and Peter D. Karp. The BioCyc Metabolic Network Explorer. *BMC Bioinformatics*, 22(1):208, December 2021.

[PKH20] Lace Padilla, Matthew Kay, and Jessica Hullman. Uncertainty Visualization, April 2020.

[PLS+13] Christian Partl, Alexander Lex, Marc Streit, Denis Kalkofen, Karl Kashofer, and Dieter Schmalstieg. enRoute: dynamic path extraction from biological pathway maps for exploring heterogeneous experimental datasets. *BMC Bioinformatics*, 14(19):S3, November 2013.

[PM20a] Sabyasachi Patra and Anjali Mohapatra. Review of tools and algorithms for network motif discovery in biological networks. *IET Systems Biology*, 14(4):171–189, 2020.

[PM20b] Bernhard Preim and Monique Meuschke. A survey of medical animations. *Computers & Graphics*, 90:145–168, August 2020.

[Pot13] Kristin Potter. Uncertainty Visualization and Data References, 2013.

[PRJ12] Kristin Potter, Paul Rosen, and Chris R. Johnson. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches. In Andrew M. Dienstfrey and Ronald F. Boisvert, editors, *Uncertainty Quantification in Scientific Computing*, pages 226–249, Berlin, Heidelberg, 2012. Springer.

[PRKM+14] José P Pinto, Ravi Kiran Reddy Kalathur, Rui S R Machado, Joana M Xavier, José Bragança, and Matthias E Futschik. StemCellNet: an interactive platform for network-oriented investigations in stem cell biology. *Nucleic Acids Res.*, 42(Web Server issue):W154–60, July 2014.

[PS12] Adam Perer and Jimeng Sun. MatrixFlow: Temporal Network Visual Analytics to Track Symptom Evolution during Disease Progression. *AMIA Annual Symposium Proceedings*, 2012:716–725, November 2012.

[PSGG+19] Blanca Pijuan-Sala, Jonathan A Griffiths, Carolina Guibentif, Tom W Hiscock, Wajid Jawaid, Fernando J Calero-Nieto, Carla Mulas, Ximena Ibarra-Soria, Richard CV Tyser, Debbie Lee Lian Ho, and others. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019.

[PSM+11] Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, December 2011.

[PTLZ18] Di Peng, Wei Tian, Binbin Lu, and Min Zhu. DMNEVis: A Novel Visual Approach to Explore Evolution of Dynamic Multivariate Network. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4304–4311, Miyazaki, Japan, October 2018. IEEE.

[Pur97] Helen Purchase. Which aesthetic has the greatest effect on human understanding? In Giuseppe DiBattista, editor, *Graph Drawing*, Lecture Notes in Computer Science, pages 248–261, Berlin, Heidelberg, 1997. Springer.

[Pur98] HELEN C. Purchase. Performance of Layout Algorithms: Comprehension, not Computation. *Journal of Visual Languages & Computing*, 9(6):647–657, December 1998.

[Pur00] H.C Purchase. Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers*, 13(2):147–162, December 2000.

[Pur02] HELEN C. Purchase. Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, October 2002.

[PVW06] A. Johannes Pretorius and Jarke J. Van Wijk. Visual Analysis of Multivariate State Transition Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):685–692, September 2006.

[PWL97] Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, November 1997.

[PWS08] Georgios A Pavlopoulos, Anna-Lynn Wegener, and Reinhard Schneider. A survey of visualization tools for biological network analysis. *BioData Min.*, 1(1):12, November 2008.

[PZS+20] Jiansu Pu, Jingwen Zhang, Hui Shao, Tingting Zhang, and Yunbo Rao. egoDetect: Visual Detection and Exploration of Anomaly in Social Communication Network. *Sensors*, 20(20):5895, January 2020.

[Que15] Laurent E. Quevillon. Social, spatial, and temporal organization in a complex insect society. *Scientific Reports*, 5, 2015.

[Rav24] Nikhil Ravi. nikhil-ravi/harry-potter-interactions, March 2024.

[RCM+20] Guido Reina, Hank Childs, Krešimir Matković, Katja Bühler, Manuela Waldner, David Pugmire, Barbora Kozlíková, Timo Ropinski, Patric Ljung, Takayuki Itoh, Eduard Gröller, and Michael Krone. The moving target of visualization software for an increasingly complex world. *Computers & Graphics*, 87:12–29, 2020.

[RD10] Nathalie Henry Riche and Tim Dwyer. Untangling Euler Diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, November 2010.

[RDD16] Oscar Robinson, David Dylus, and Christophe Dessimoz. Phylo.io : Interactive Viewing and Comparison of Large Phylogenetic Trees on the Web. *Molecular Biology and Evolution*, 33(8):2163–2166, August 2016.

[RFF+08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, November 2008.

[RHR+10] Markus Rohrschneider, Christian Heine, André Reichenbach, Andreas Kerren, and Gerik Scheuermann. A Novel Grid-Based Visualization Approach for Metabolic Networks with Advanced Focus&Context View. In David Eppstein and Emden R. Gansner, editors, *Graph Drawing*, Lecture Notes in Computer Science, pages 268–279, Berlin, Heidelberg, 2010. Springer.

[Rif12] Scott A. Rifkin. *Quantitative Trait Loci (QTL): Methods and Protocols*, volume 871 of *Methods in Molecular Biology*. Humana, New York, NY, USA, 1 edition, 2012.

[RJH+12] Hendrik Rohn, Astrid Junker, Anja Hartmann, Eva Grafahrend-Belau, Hendrik Treutler, Matthias Klapperstück, Tobias Czauderna, Christian Klukas, and Falk Schreiber. VANTED v2: a framework for systems biology applications. *BMC Systems Biology*, 6(1):139, November 2012.

[RLBS03] P J Rhodes, R S Laramee, R D Bergeron, and T M Sparr. Uncertainty Visualization Methods in Isosurface Rendering. 2003.

[RM14] Sébastien Rufiange and Guy Melançon. AniMatrix: A Matrix-Based Visualization of Software Evolution. In *2014 Second IEEE Working Conference on Software Visualization*, pages 137–146, September 2014.

[RM16] Giorgio Roffo and Simone Melzi. Feature Selection via Eigenvector Centrality. *Contributo in atti di convegno*, 4(1):13, 2016.

[RMF12] Sébastien Rufiange, Michael J. McGuffin, and Christopher P. Fuhrman. TreeMatrix: A Hybrid Visualization of Compound Graphs. *Computer Graphics Forum*, 31(1):89–101, 2012.

[RMM15] B. Renoust, G. Melançon, and T. Munzner. Detangler: Visual Analytics for Multiplex Networks. *Computer Graphics Forum*, 34(3):321–330, 2015.

[RMO+19] Donghao Ren, Laura R. Marusich, John O'Donovan, Jonathan Z. Bakdash, James A. Schaffer, Daniel N. Cassenti, Sue E. Kase, Heather E. Roy, Wan-yi (Sabrina) Lin, and Tobias Höllerer. Understanding node-link and matrix visualizations of networks: A large-scale online experiment. *Network Science*, 7(2):242–264, June 2019.

[RN15] Ryan A. Rossi and K. Ahmed Nesreen. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. 2015.

[RPM20] Muhammad Rizwan Riaz, Gail M Preston, and Aziz Mithani. MAPPS: A web-based tool for metabolic pathway prediction and network Analysis in the postgenomic era. *ACS Synth. Biol.*, 9(5):1069–1082, May 2020.

[RRT99] Peter J. Rousseeuw, Ida Ruts, and John W. Tukey. The Bagplot: A Bivariate Boxplot. *The American Statistician*, 53(4):382–387, November 1999.

[RSA+16] Peter Rodgers, Gem Stapleton, Bilal Alsallakh, Luana Michallef, Rob Baker, and Simon Thompson. A task-based evaluation of combined set and network visualization. *Information Sciences*, 367-368:58–79, November 2016.

[RSC15] Peter Rodgers, Gem Stapleton, and Peter Chapman. Visualizing Sets with Linear Diagrams. *ACM Transactions on Computer-Human Interaction*, 22(6):27:1–27:39, September 2015.

[SA17] Asuda Sharma and Hesham H. Ali. Analysis of clustering algorithms in biological networks. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2303–2305, November 2017.

[SAHB11] Hamid Sanatnama, Afshin Amini, Alireza Bitaraf Haghighi, and Farshad Brahimi. Positioning a New Vertex that Minimize the Number of New Crossings. *Journal of Applied Sciences*, 11(12):2260–2264, June 2011.

[SAK+21] J. Sorger, A. Arleo, P. Kán, W. Knecht, and M. Waldner. Egocentric Network Exploration for Immersive Analytics. *Computer Graphics Forum*, 40(7):241–252, 2021.

[Say04] Craig Sayers. Node-centric RDF Graph Visualization. 2004.

[SBG00] T.C. Sprenger, R. Brunella, and M.H. Gross. H-BLOB: a hierarchical visual clustering method using implicit surfaces. In *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, pages 61–68, October 2000.

[SBL93] John Stasko, Albert Badre, and Clayton Lewis. Do algorithm animations assist learning? an empirical study and analysis. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 61–66, New York, NY, USA, May 1993. Association for Computing Machinery.

[SBo99] Tom AB Snijders, Stephen P Borgatti, and others. Non-parametric standard errors and tests for network statistics. *Connections*, 22(2):161–170, 1999. Publisher: Citeseer.

[SBV+21] Sherif Sakr, Angela Bonifati, Hannes Voigt, Alexandru Iosup, Khaled Ammar, Renzo Angles, Walid Aref, Marcelo Arenas, Maciej Besta, Peter A. Boncz, Khuzaima Daudjee, Emanuele Della Valle, Stefania Dumbrava, Olaf Hartig, Bernhard Haslhofer, Tim Hegeman, Jan Hidders, Katja Hose, Adriana Iamnitchi, Vasiliki Kalavri, Hugo Kapp, Wim Martens, M. Tamer Özsu, Eric Peukert, Stefan Plantikow, Mohamed Ragab, Matei R. Ripeanu, Semih Salihoglu, Christian Schulz, Petra Selmer, Juan F. Sequeda, Joshua Shinavier, Gábor Szárnyas, Riccardo Tommasini, Antonino Tumeo, Alexandru Uta, Ana Lucia Varbanescu, Hsiang-Yun Wu, Nikolay Yakovets, Da Yan, and Eiko Yoneki. The future is big graphs: a community view on graph processing systems. *Communications of the ACM*, 64(9):62–71, August 2021.

[Sch11] Hans-Jorg Schulz. Treevis.net: A Tree Visualization Reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, November 2011.

[SD14] Antonio Scala and G. D'Agostino. *Networks of Networks: The Last Frontier of Complexity*. Springer Complexity. Springer, January 2014.

[SERG14] Sucheta Soundarajan, Tina Eliassi-Rad, and Brian Gallagher. A Guide to Selecting a Network Similarity Method. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 1037–1045. Society for Industrial and Applied Mathematics, April 2014.

[SGL+19] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, Lars J Jensen, and Christian von Mering. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, 47(D1):D607–D613, January 2019.

[SGLLB18] Mahito Sugiyama, M Elisabetta Ghisu, Felipe Llinares-López, and Karsten Borgwardt. graphkernels: R and Python packages for graph comparison. *Bioinformatics*, 34(3):530–532, February 2018.

[Shn96] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, Boulder, CO, USA, 1996. IEEE Comput. Soc. Press.

[SHW+20] Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. Persistent Homology Guided Force-Directed Graph Layouts. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):697–707, January 2020.

[SI17] Christine Schikora and Daniel Isemann. InfluViz — A Visualization Tool for Exploring and Analyzing Creative Influence Between Artists and their Works. In *2017 21st International Conference Information Visualisation (IV)*, pages 336–343, July 2017.

[SJ08] Ralf Steuer and Björn H. Junker. Computational Models of Metabolism: Stability and Regulation in Metabolic Networks. In Stuart A. Rice, editor, *Advances in Chemical Physics*, pages 105–251. John Wiley & Sons, Inc., Hoboken, NJ, USA, November 2008.

[SL14] Ruojing W. Scholz and Yongmei Lu. Uncertainty in Geographic Data on Bivariate Maps: An Examination of Visualization Preference and Decision Making. *ISPRS International Journal of Geo-Information*, 3(4):1180–1197, December 2014.

[SLN05] Purvi Saraiya, P. Lee, and C. North. Visualization of graphs with associated timeseries data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 225–232, October 2005.

[SM03] Victor Spirin and Leonid A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12123–12128, October 2003.

[SM07] Zeqian Shen and Kwan-Liu Ma. Path visualization for adjacency matrices. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC conference on Visualization*, EUROVIS'07, pages 83–90, Goslar, DEU, May 2007. Eurographics Association.

[SMDS14] Ramik Sadana, Timothy Major, Alistair Dove, and John Stasko. OnSet: A Visualization Technique for Large-scale Binary Set Data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1993–2002, December 2014.

[SMO+03] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13(11):2498–2504, November 2003.

[SN11] Stephan Symons and Kay Nieselt. MGV: a generic graph viewer for comparative omics data. *Bioinformatics*, 27(16):2248–2255, August 2011.

[SNG+17] Christoph Schulz, Arlind Nocaj, Jochen Goertler, Oliver Deussen, Ulrik Brandes, and Daniel Weiskopf. Probabilistic Graph Layout for Uncertain Network Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):531–540, January 2017.

[SOGB10] A Shabbeer, C Ozcaglar, M Gonzalez, and K P Bennett. Optimal Embedding of Heterogeneous Graph Data with Edge Crossing Constraints. November 2010.

[SOR+11] Michael E Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, February 2011.

[SR05] Wolfgang Schnotz and Thorsten Rasch. Enabling, facilitating, and inhibiting effects of animations in multimedia learning: Why reduction of cognitive load can have negative results on learning. *Educational Technology Research and Development*, 53(3):47–58, September 2005.

[SS18] Johannes Schwank and Sebastian Schöffel. Visualizing Uncertainty in Node-Link Diagrams - a User Study. In Tareq Ahram and Christianne Falcão, editors, *Advances in Usability and User Experience*, pages 495–507, Cham, 2018. Springer International Publishing.

[SSBE08] Lucy Skrabanek, Harpreet K. Saini, Gary D. Bader, and Anton J. Enright. Computational prediction of protein-protein interactions. *Molecular Biotechnology*, 38(1):1–17, January 2008.

[SSK+16] Dominik Sacha, Hansi Senaratne, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. The Role of Uncertainty, Awareness, and Trust in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):240–249, January 2016.

[SSS+14] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, December 2014.

[SSSE16] Johannes Schwank, Sebastian Schöffel, Jan Stärz, and Achim Ebert. Visualizing Uncertainty of Edge Attributes in Node-Link Diagrams. In *2016 20th International Conference Information Visualisation (IV)*, pages 45–50, July 2016.

[SSSEA20] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1064–1074, January 2020.

[SSvH14] Christoph Daniel Schulze, Miro Spönemann, and Reinhard von Hanxleden. Drawing layered graphs with port constraints. *Journal of Visual Languages & Computing*, 25(2):89–106, April 2014.

[STT81] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, February 1981.

[SVK+20] Indhupriya Subramanian, Srikant Verma, Shiva Kumar, Abhay Jere, and Krishanpal Anamika. Multi-omics Data Integration, Interpretation, and Its Application. *Bioinformatics and Biology Insights*, 14:117793221989905, January 2020.

[SWKP18] Stephen D. Shank, Steven Weaver, and Sergei L. Kosakovsky Pond. phylotree.js - a JavaScript library for application development and interactive data visualization in phylogenetics. *BMC Bioinformatics*, 19(1):276, July 2018.

[SWW+15] Lei Shi, Chen Wang, Zhen Wen, Huamin Qu, Chuang Lin, and Qi Liao. 1.5D Egocentric Dynamic Network Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(5):624–637, May 2015.

[SZB+09] Jibonananda Sanyal, Song Zhang, Gargi Bhattacharya, Phil Amburn, and Robert Moorhead. A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1209–1218, November 2009.

[SZD+10] Jibonananda Sanyal, Song Zhang, Jamie Dyer, Andrew Mercer, Philip Amburn, and Robert Moorhead. Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, November 2010.

[SZPM10] Arnaud Sallaberry, Faraz Zaidi, C Pich, and Guy Melançon. Interactive Visualization and Navigation of Web Search Results Revealing Community Structures and Bridges. In *Proceedings of Graphics Interface*, pages 105–112, 2010.

[TA08] Alexandru Telea and David Auber. Code Flows: Visualizing Structural Evolution of Source Code. *Computer Graphics Forum*, 27(3):831–838, 2008.

[Tam14] Roberto Tamassia, editor. *Handbook of graph drawing and visualization*. Discrete mathematics and its applications. CRC Press, Taylor & Francis Group, Boca Raton, 2014.

[TBNV10] Jan Terje Bjørke, Stein Nilsen, and Margaret Varga. Visualization of network structure by the application of hypernodes. *International Journal of Approximate Reasoning*, 51(3):275–293, February 2010.

[TCG+22] Jakob Troidl, Corrado Cali, Eduard Gröller, Hanspeter Pfister, Markus Hadwiger, and Johanna Beyer. Barrio: Customizable Spatial Neighborhood Analysis and Comparison for Nanoscale Brain Structures. *Computer Graphics Forum*, 41(3):183–194, 2022.

[TCG24] Matteo Tiezzi, Gabriele Ciravegna, and Marco Gori. Graph Neural Networks for Graph Drawing. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4668–4681, April 2024.

[TDDM+16] Oren Tzfadia, Tim Diels, Sam De Meyer, Klaas Vandepoele, Asaph Aharoni, and Yves Van de Peer. CoExpNetViz: Comparative Co-Expression Networks Construction and Visualization Tool. *Frontiers in Plant Science*, 6, 2016.

[TEP+17] Theodosios Theodosiou, Georgios Efstathiou, Nikolas Papanikolaou, Nikos C Kyrpides, Pantelis G Bagos, Ioannis Iliopoulos, and Georgios A Pavlopoulos. NAP: The Network Analysis Profiler, a web tool for easier topological analysis and comparison of medium-scale biological networks. *BMC Res. Notes*, 10(1):278, July 2017.

[TITP19] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing methods for comparing networks. *Scientific Reports*, 9(1):17557, December 2019.

[TJS86] K. Thulasiraman, R. Jayakumar, and M. Swamy. On maximal planarization of nonplanar graphs. *IEEE Transactions on Circuits and Systems*, 33(8):843–844, August 1986.

[TLLS20] J. Thompson, Z. Liu, W. Li, and J. Stasko. Understanding the Design Space and Authoring Paradigms for Animated Data Graphics. *Computer Graphics Forum*, 39(3):207–218, 2020.

[TLW+15] Yu Tang, Min Li, Jianxin Wang, Yi Pan, and Fang-Xiang Wu. CytoNCA: A cytoscape plugin for centrality analysis and evaluation of protein interaction networks. *Biosystems*, 127:67–72, January 2015.

[TMB02] BARBARA Tversky, JULIE BAUER Morrison, and MIREILLE Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, October 2002.

[TMB+10] Isaak Y. Tecle, Naama Menda, Robert M. Buels, Esther van der Knaap, and Lukas A. Mueller. solQTL: a tool for QTL analysis, visualization and linking to genomes at SGN database. *BMC Bioinformatics*, 11(1):525, 2010.

[TRZ18] Yin Tong, Beibei Ru, and Jiangwen Zhang. miRNACancerMAP: an integrative web server inferring miRNA regulation network for cancer. *Bioinformatics*, 34(18):3211–3213, September 2018.

[TTvE14] Susanne Tak, Alexander Toet, and Jan van Erp. The Perception of Visual UncertaintyRepresentation by Non-Experts. *IEEE Transactions on Visualization and Computer Graphics*, 20(6):935–943, June 2014.

[VBAW15] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. Visualizing the Evolution of Communities in Dynamic Graphs. *Computer Graphics Forum*, 34(1):277–288, 2015.

[VBW15] Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. The State of the Art in Visualizing Group Structures in Graphs. *Eurographics Conference on Visualization (EuroVis) - STARs*, page 20 pages, 2015.

[VDS+07] Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Geeta Joshi-Tope, Gopal Gopinath, David Croft, Bernard De Bono, Marc Gillespie, Bijay Jassal, Suzanna Lewis, Lisa Matthews, Guanming Wu, Ewan Birney, and Lincoln Stein. Reactome: a knowledge base of biologic pathways and processes. *Genome Biology*, 8(3):R39, 2007.

[VGN14] Leif Väremo, Francesco Gatto, and Jens Nielsen. Kiwi: a tool for integration and visualization of network topology and gene-set analysis. *BMC Bioinformatics*, 15(1):408, December 2014.

[VHK+12] Corinna Vehlow, Jan Hasenauer, Andrei Kramer, Julian Heinrich, Nicole Radde, Frank Allgöwer, and Daniel Weiskopf. Uncertainty-aware visual analysis of biochemical reaction networks. In *2012 IEEE Symposium on Biological Data Visualization (BioVis)*, pages 91–98, October 2012.

[VHTW13] Corinna Vehlow, Jan Hasenauer, Fabian J. Theis, and Daniel Weiskopf. Visualizing edge-edge relations in graphs. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208, February 2013.

[VHW08] F. Van Ham and M. Wattenberg. Centrality Based Visualization of Small World Graphs. *Computer Graphics Forum*, 27(3):975–982, 2008.

[Vie16] Damien M. de Vienne. Lifemap: Exploring the Entire Tree of Life. *PLOS Biology*, 14(12):e2001624, 2016.

[VK16] M.K Vijaymeena and K Kavitha. A Survey on Similarity Measures in Text Mining. *Machine Learning and Applications: An International Journal*, 3(1):19–28, March 2016.

[VKKG17] Zana Vosough, Dietrich Kammer, Mandy Keck, and Rainer Groh. Visualizing uncertainty in flow diagrams: a case study in product costing. In *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, VINCI '17, pages 1–8, New York, NY, USA, August 2017. Association for Computing Machinery.

[VKKG19] Zana Vosough, Dietrich Kammer, Mandy Keck, and Rainer Groh. Visualization approaches for understanding uncertainty in flow diagrams. *Journal of Computer Languages*, 52:44–54, June 2019.

[vLKS+11] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.j. van Wijk, J.-D. Fekete, and D.w. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[VM21] Loan Vulliard and Jörg Menche. Complex Networks in Health and Disease. *Systems Medicine*, pages 26–33, 2021.

[VRW14] Corinna Vehlow, Thomas Reinhardt, and Daniel Weiskopf. Visualizing Fuzzy Overlapping Communities in Networks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2486–2495, 2014.

[War21] Colin Ware. Chapter Three - Lightness, Brightness, Contrast, and Constancy. In Colin Ware, editor, *Information Visualization (Fourth Edition)*, Interactive Technologies, pages 69–94. Morgan Kaufmann, January 2021.

[WB05] Colin Ware and Robert Bobrow. Supporting Visual Queries on Medium-Sized Node–Link Diagrams. *Information Visualization*, 4(1):49–58, March 2005.

[WBQH12] E. Wong, B. Baur, S. Quader, and C.-H. Huang. Biological network motif detection: principles and practice. *Briefings in Bioinformatics*, 13(2):202–215, March 2012.

[WBR15] Christian Wiwie, Jan Baumbach, and Richard Röttger. Comparing the performance of biomedical clustering methods. *Nature Methods*, 12(11):1033–1038, November 2015.

[WDD+14] Guanming Wu, Eric Dawson, Adrian Duong, Robin Haw, and Lincoln Stein. ReactomeFIViz: a Cytoscape app for pathway and network-based data analysis. *F1000Research*, 3:146, September 2014.

[WDNM17] Lin Wang, Satyakam Dash, Chiam Yu Ng, and Costas D. Maranas. A review of computational tools for design and reconstruction of metabolic pathways. *Synthetic and Systems Biotechnology*, 2(4):243–252, December 2017.

[WEF+14] Michael Wybrow, Niklas Elmqvist, Jean-Daniel Fekete, Tatiana von Landesberger, Jarke J. van Wijk, and Björn Zimmer. Interaction in the Visualization of Multivariate Networks. In Andreas Kerren, Helen C. Purchase, and Matthew O. Ward, editors, *Multivariate Network Visualization: Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions*, pages 97–125. Springer International Publishing, Cham, 2014.

[Wei22] Daniel Weiskopf. Uncertainty Visualization: Concepts, Methods, and Applications in Biological Data Visualization. *Frontiers in Bioinformatics*, 2, February 2022.

[WEK02] Roland Wiese, Markus Eiglsperger, and Michael Kaufmann. yFiles: Visualization and Automatic Layout of Graphs. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing*, volume 2265, pages 453–454. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[WFGH11] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 143–146, New York, NY, USA, May 2011. Association for Computing Machinery.

[Wil92] Frank Wilcoxon. Individual Comparisons by Ranking Methods. In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics: Methodology and Distribution*, Springer Series in Statistics, pages 196–202. Springer, New York, NY, 1992.

[WJW+20] Yong Wang, Zhihua Jin, Qianwen Wang, Weiwei Cui, Tengfei Ma, and Huamin Qu. DeepDrawing: A Deep Learning Approach to Graph Drawing. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):676–686, January 2020.

[WLL+16] Xiting Wang, Shixia Liu, Junlin Liu, Jianfei Chen, Jun Zhu, and Baining Guo. TopicPanorama: A Full Picture of Relevant Topics. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2508–2521, December 2016.

[WNSV19] Hsiang-Yun Wu, Martin Nöllenburg, Filipa L. Sousa, and Ivan Viola. Metabopolis: scalable network layout for biological pathway diagrams in urban map style. *BMC Bioinformatics*, 20(1):187, April 2019.

[WNV19] Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Graph Models for Biological Pathway Visualization: State of the Art and Future Challenges. In *The 1st Workshop on Multilayer Nets: Challenges in Multilayer Network Visualization and Analysis*, Vancouver, Canada, October 2019.

[WNV22] Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Multi-Level Area Balancing of Clustered Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(7):2682–2696, July 2022.

[WPZ+16] Yanhong Wu, Naveen Pitipornvivat, Jian Zhao, Sixiao Yang, Guowei Huang, and Huamin Qu. egoSlider: Visual Analysis of Egocentric Network Evolution. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):260–269, January 2016.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[WSA+16] Yong Wang, Qiaomu Shen, Daniel Archambault, Zhiguang Zhou, Min Zhu, Sixiao Yang, and Huamin Qu. AmbiguityVis: Visualization of Ambiguity in Graph Layouts. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):359–368, January 2016.

[WSZH15] Ryan R. Wick, Mark B. Schultz, Justin Zobel, and Kathryn E. Holt. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, 31(20):3350–3352, 2015.

[WYHS21] Xiaoqi Wang, Kevin Yen, Yifan Hu, and Han-Wei Shen. DeepGD: A Deep Learning Framework for Graph Drawing Using GNN. *IEEE Computer Graphics and Applications*, 41(5):32–44, September 2021.

[WZC+15] Jianxin Wang, Jiancheng Zhong, Gang Chen, Min Li, Fang-Xiang Wu, and Yi Pan. ClusterViz: A Cytoscape APP for cluster analysis of biological network. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 12(4):815–822, July 2015.

[WZM+16] Xu-Meng Wang, Tian-Ye Zhang, Yu-Xin Ma, Jing Xia, and Wei Chen. A Survey of Visual Analytic Pipelines. *Journal of Computer Science and Technology*, 31(4):787–804, July 2016.

[XGH15] Jianguo Xia, Erin E. Gill, and Robert E. W. Hancock. NetworkAnalyst for statistical, visual and network-based meta-analysis of gene expression data. *Nature Protocols*, 10(6):823–844, June 2015.

[XMB05] Xu Liu, Ming Luo, and Ben Shneiderman. Visualization of Sets. 2005.

[XW10] Jianguo Xia and David S. Wishart. MetPA: a web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics*, 26(18):2342–2344, September 2010.

[YAD+18] Vahan Yoghourdjian, Daniel Archambault, Stephan Diehl, Tim Dwyer, Karsten Klein, Helen C. Purchase, and Hsiang-Yun Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Visual Informatics*, 2(4):264–282, December 2018.

[YC17] Kai Yan and Weiwei Cui. Visualizing the uncertainty induced by graph layout algorithms. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 200–209, April 2017.

[YDK+18] Vahan Yoghourdjian, Tim Dwyer, Karsten Klein, Kim Marriott, and Michael Wybrow. Graph Thumbnails: Identifying and Comparing Multiple Graphs at a Glance. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3081–3095, December 2018.

[YHH+08] Jian Yang, Chengcheng Hu, Han Hu, Rongdong Yu, Zhen Xia, Xiuzi Ye, and Jun Zhu. QTLNetwork: mapping and visualizing genetic architecture of complex traits in experimental populations. *Bioinformatics*, 24(5):721–723, 2008.

[YKSJ07] Ji Soo Yi, Youn ah Kang, John Stasko, and J.A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, November 2007.

[YSD+17] Xinsong Yang, Lei Shi, Madelaine Daianu, Hanghang Tong, Qingsong Liu, and Paul Thompson. Blockwise Human Brain Network Visual Comparison Using NodeTrix Representation. *IEEE transactions on visualization and computer graphics*, 23(1):181–190, January 2017.

[yWo22] yWorks GmbH. yEd - Graph Editor, 2022.

[ZA11] Brian Zier and Inoue Atsushi. Fuzzy Relational Visualization for Decision Support. Technical report, 2011.

[ZAH22] Dongping Zhang, Eytan Adar, and Jessica Hullman. Visualizing Uncertainty in Probabilistic Graphs with Network Hypothetical Outcome Plots (NetHOPs). *IEEE Transactions on Visualization and Computer Graphics*, 28(1):443–453, January 2022.

[ZCR+21] Youjia Zhou, Nithin Chalapathi, Archit Rathore, Yaodong Zhao, and Bei Wang. Mapper Interactive: A Scalable, Extendable, and Interactive Toolbox for the Visual Exploration of High-Dimensional Data. *arXiv:2011.03209 [cs]*, April 2021.

[ZGC+16] Jian Zhao, Michael Glueck, Fanny Chevalier, Yanhong Wu, and Azam Khan. Egocentric Analysis of Dynamic Networks with EgoLines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5003–5014, New York, NY, USA, May 2016. Association for Computing Machinery.

[ZJK12] Björn Zimmer, Ilir Jusufi, and Andreas Kerren. Analyzing Multiple Network Centralities with ViNCent. 2012.

[ZJVS19] Armita Zarnegar, Herbert F. Jelinek, Peter Vamplew, and Andrew Stranieri. Integrating Biological Heuristics and Gene Expression Data for Gene Regulatory Network Inference. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW 2019, New York, NY, USA, 2019. Association for Computing Machinery.

[ZKER06] Zeqian Shen, Kwan-Liu Ma, and T. Eliassi-Rad. Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1427–1439, November 2006.

[ZLC+20] Mark Zander, Mathew G. Lewsey, Natalie M. Clark, Lingling Yin, Anna Bartlett, J. Paola Saldierna Guzmán, Elizabeth Hann, Amber E. Langford, Bruce Jow, Aaron Wise, Joseph R. Nery, Huaming Chen, Ziv Bar-Joseph, Justin W. Walley, Roberto Solano, and Joseph R. Ecker. Integrated multi-omics framework of the plant response to jasmonic acid. *Nature Plants*, 6(3):290–302, March 2020.

[ZLZ+20] Jiashuai Zhang, Wenkai Li, Min Zeng, Xiangmao Meng, Lukasz Kurgan, Fang-Xiang Wu, and Min Li. NetEPD: A network-based essential protein discovery platform. *Tsinghua Science and Technology*, 25(4):542–552, August 2020.

[ZMOP08] Elena Zotenko, Julian Mestre, Dianne P. O'Leary, and Teresa M. Przytycka. Why Do Hubs in the Yeast Protein Interaction Network Tend To Be Essential: Reexamining the Connection between the Network Topology and Essentiality. *PLOS Computational Biology*, 4(8):e1000140, August 2008.

[ZQL+18] Huijie Zhang, Dezhan Qu, Quanle Liu, Qi Shang, Yafang Hou, and Han-Wei Shen. Uncertainty visualization for variable associations analysis. *The Visual Computer*, 34(4):531–549, April 2018.

[ZSG+15] Yongnan Zhu, Liang Sun, Alexander Garbarino, Carl Schmidt, Jinglong Fang, and Jian Chen. PathRings: a web-based tool for exploration of ortholog and expression data in biological pathways. *BMC Bioinformatics*, 16(1):165, May 2015.

[ZSZ+17] Mingqian Zhao, Yijia Su, Jian Zhao, Shaoyu Chen, and Huamin Qu. Mobile situated analytics of ego-centric network data. In *SIGGRAPH Asia 2017 Symposium on Visualization*, SA '17, pages 1–8, New York, NY, USA, November 2017. Association for Computing Machinery.

[ZWY22] Panpan Zhang, Tiandong Wang, and Jun Yan. PageRank centrality and algorithms for weighted, directed networks. *Physica A: Statistical Mechanics and its Applications*, 586:126438, January 2022.

[ZX18] Guangyan Zhou and Jianguo Xia. OmicsNet: a web-based tool for creation and visual analysis of biological networks in 3D space. *Nucleic Acids Research*, 46(W1):W514–W522, July 2018.

[ZX19] Guangyan Zhou and Jianguo Xia. Using OmicsNet for network integration and 3D visualization. *Curr. Protoc. Bioinformatics*, 65(1):e69, March 2019.

[ZXQ15] Hong Zhou, Panpan Xu, and Huamin Qu. Visualization of bipartite relations between graphs and sets. *Journal of Visualization*, 18(2):159–172, May 2015.

[ZYZ10] Wen-jun Zheng, Jian Yang, and Jun Zhu. QTLNetworkR: an interactive R package for QTL visualization. *Journal of Zhejiang University SCIENCE B*, 11(7):512–515, July 2010.