Computers & Graphics 126 (2025) 104115

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Survey Paper

# An introduction to and survey of biological network visualization ☆

Henry Ehlers [a,*], Nicolas Brich [b], Michael Krone [b,f], Martin Nöllenburg [a], Jiacheng Yu [c], Hiroaki Natsukawa [d], Xiaoru Yuan [c,g], Hsiang-Yun Wu [e,a]

[a] TU Wien, Austria
[b] University of Tübingen, Germany
[c] National Key Laboratory of General Artificial Intelligence, and School of Intelligence Science and Technology, Peking University, China
[d] Osaka Seikei University, Japan
[e] St. Pölten University of Applied Sciences, Austria
[f] Stuttgart University of Applied Sciences, Germany
[g] Health Data Visualization and Visual Analytics Research Center, National Institute of Health Data Science, Peking University, China

ARTICLE INFO

*Keywords:*
Network visualization
Biological networks
State-of-the-art-report
Visualization pipeline
Sensemaking loop
Visual analytics
Network analysis

ABSTRACT

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 aims 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 paper'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 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.

## 1. 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 [1]. 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 [2]. 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 [3].

☆ This article was recommended for publication by Stefan Bruckner.

* Corresponding author.
*E-mail address:* henry.ehlers@tuwien.ac.at (H. Ehlers).

https://doi.org/10.1016/j.cag.2024.104115
Received 4 July 2024; Received in revised form 21 October 2024; Accepted 28 October 2024
Available online 23 November 2024

These networks are often best understood as an abstract representation of the community's current state of knowledge about a system of interest [4]. 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 [5].

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 [6] (Section 8.1.2) or the development of (by today's standard) small gene regulatory network maps of 40 genes [7] (Section 8.1.1). 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, enable a holistic understanding of these biochemical networks [8]. 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 8.1.4), sometimes described as a "network of networks", i.e. the interlacing of multiple inhomogeneous networks into a single larger network [9]. 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 [10].

Earlier biochemical networks, owing to their simpler nature and smaller size, lent themselves more readily to "automated" quantitative analysis. However, owing to the dimensionality 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 [11].

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 [2], 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 representations are intuitive to read and straightforward to implement [12], but scale poorly with increasing numbers of entities, relationships, and layers [13]; only to produce what is often referred to as a "hairball" [2,14]. On the other hand, a more abstract approach such as Yoghourdjian et al.'s [15] *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 straight-forward 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 [8]. However, sensible representations of high-dimensional data are only one aspect of making such networks understandable. The second, and often underappreciated [16], component to assist in making such visualizations readable is the use of effective interaction techniques [11]. 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 [17]. 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 [3], (ii) an overview of the requirements of, and layouts useful to, biological network visualization [18], (iii) a state-of-the-art report on multivariate graph visualization and analysis tasks associated therewith [19], (iv) surveys on the visualization of group structure in graphs [20,21] (v) the development of a general taxonomy to describe the tasks performed in biological pathway visualization [22], (vi) a discussion of taxonomies to categorize methods of interaction [16], (vii) a break-down of graph theory to assist domain experts in understanding graph data structures and algorithms [2,23], (viii) the identification of a number of popular network visualization tools to compare their applicability to high-dimensional data [24], and most recently (ix) Filipov et al.'s [25] 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. [26]. 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), through the topics common in network analysis (Section 4), to various visualization approaches (Section 5)
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 5.4) and straight-line node-link diagrams (Section 5.1, despite the existence of several powerful alternatives (Section 5),
   (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 4) that go beyond basic descriptive measures with which to rank nodes based on topological importance,

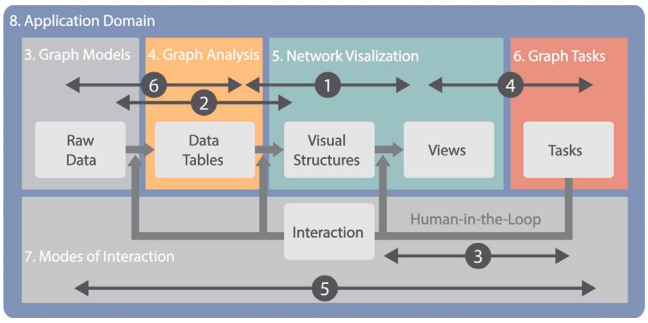



**Fig. 1.** The biological network visualization pipelines (extended from Card et al. [33]) together with the classes of our taxonomy. This pipeline consists of six elements, namely *raw data*, which are processed to form *data tables*, which in turn are processed to form *visual structures*, which produce actual visual *views*. All these individual elements are connected by *Task*-driven *Interaction* techniques. To support the target domain, our report groups the 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 to visualization, (iv) *Graph Tasks*, which determine the tasks necessary for the analysis at had, as well as (v) how to best design modes of *Interaction*, with which to achieve the analysis' goals, and, finally, (vi) *Applications*, which motivate and inform every aspect of the pipeline. While all these classes are discussed in their corresponding sections, only four of these form the basis of our taxonomy (Fig. 2), namely those highlighted in color, i.e. *Graph Analysis*, *Network Visualization*, *Graph Tasks*, and the various *Applications*. The six identified challenges are highlighted in dark gray circles: 1. Network Layouts, 2. Uncertainty, 3. Quantitative Analyses, 4. Graph Comparison, 5. Provenance and Trust, and finally 6. Dynamic Network Visualization. Each challenge's arrows indicate which part of the visualization pipeline they affect.

(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 8). To our knowledge, the last such compilations are now over ten years old [3]. Hence, an up-to-date list should be useful to both the domain and visualization communities.

## 2. A taxonomy for biological network visualization

The analysis methods and visualization of biological networks are non-trivial, as (i) each (sub-)domain brings with it unique goals, tasks, and challenges [27], (ii) the process of knowledge generation is seldom linear owing to users' simultaneous interactive generation and verification of hypotheses [28], and (iii) individual technical steps, from the selected graph layout algorithms to the analysis techniques employed, can influence the effectiveness of the (interactive) visualization [29]. 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 [30–32]. We instead adapt the comparatively simpler *Information Visualization Pipeline* presented by Card et al. [33], for our own taxonomy (Fig. 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 (Fig. 1) [28]. 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 [32]. We adopt the pipeline from Card et al. [33] as the basis and introduce six classes for a better arrangement of analysis strategy in the domain. The six classes include (i) *Graph Models* which describes 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 interaction, (iii) *Network Visualization* which transforms either the data tables or raw data into a structure more amenable to visualization, subsequently presented in one or more views, (iv) *Analysis Tasks* which determine the tasks necessary for the analysis at had, as well as what modes of *Interaction* are provided to achieve those analytical tasks, and (v) *Applications* which motivate and inform every aspect of the pipeline.

We discuss each of these taxonomic classes 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 1 depicts an overview of publication resources we investigated using API-driven queries (where possible) Appendix. To constrain the focus of this report to visualization examples actually relevant to Card et al.'s [33] visualization pipeline, we only adopt full papers that are also relevant in this context. More specifically, we curated a final set of 83 papers (Fig. 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, finally, (iv) this trimmed set of publications was then categorized based on our developed information-visualization-pipeline-motivated taxonomy (Fig. 1).

**Table 1**
Table representing the literature search and sources.

| 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 |

## 3. Graph models in biological networks

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 [10,26]. 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 [34,35]. 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

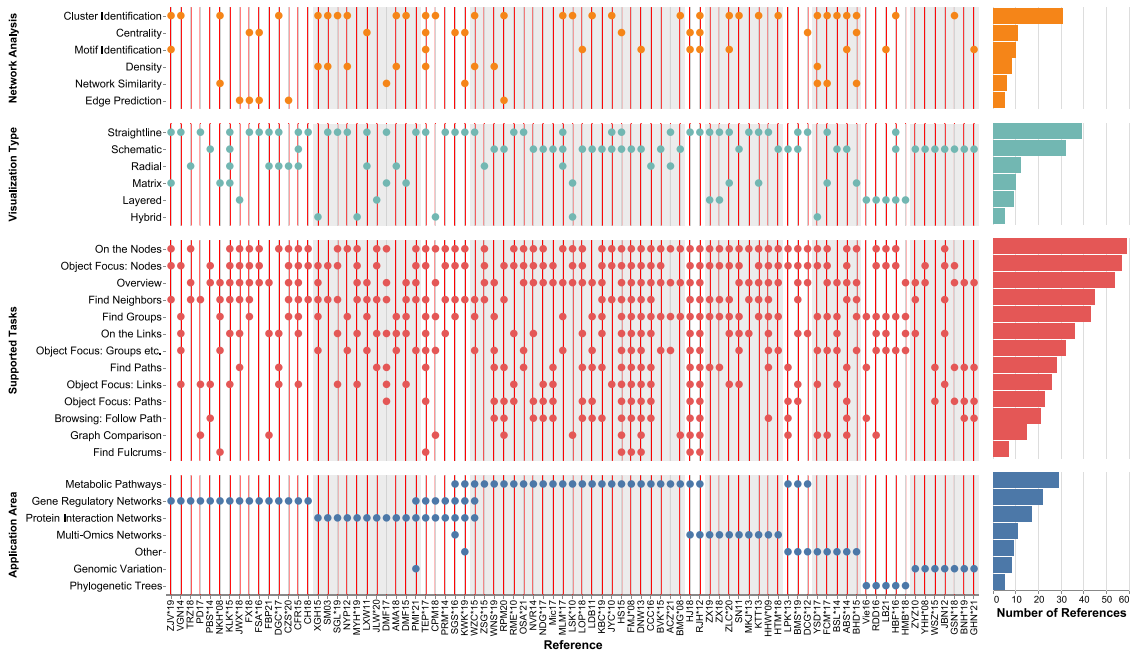



**Fig. 2.** Classification of the collected and filtered 83 papers along the $x$-axis into the four *taxonomy classes*, i.e. (i) *graph analysis techniques*, (ii) *network visualization approaches*, (iii) *graph analysis tasks*, and (iv) *application domains*, which form the y-axis' four facets. Each 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 sub-categories' 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 sub-classification are shown as a bar chart on the right.

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.

### 3.1. Simple graph

A complex relationship is often formulated as a *graph* to be manipulated mathematically [36]. The underlying graph data structure facilitates access to the data [37] 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 $V = \{v_1, v_2, \dots, v_n\}$ representing individual entities. Their mutual connectivity is represented by the edges $E = \{e_1, e_2, \dots, e_m\} \subseteq V \times V$.

This is a common definition that allows us to describe the fundamental relationships between entities for analysis and visualization purposes [38–40]. A simple graph such as this can be shown as a linked list or an adjacency matrix [2], or some hybrid format. They are often interchangeably used for different task purposes.

### 3.2. 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 that has been used in the early development of biological pathway analysis and visualization [36,41]. Examples can be found later in Figs. 11 and 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 Fig. 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.

### 3.3. 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 [42], 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 Fig. 3(b). Examples can be found later in Figs. 15 and 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.

### 3.4. Hypergraph

In principle, a hypergraph is a more intuitive and direct representation of biological pathways, and several biological network notations, such as SBML [43], BioPax [44], and KGML [41] support it. A hyperedge in a hypergraph can refer to a single biochemical reaction, in

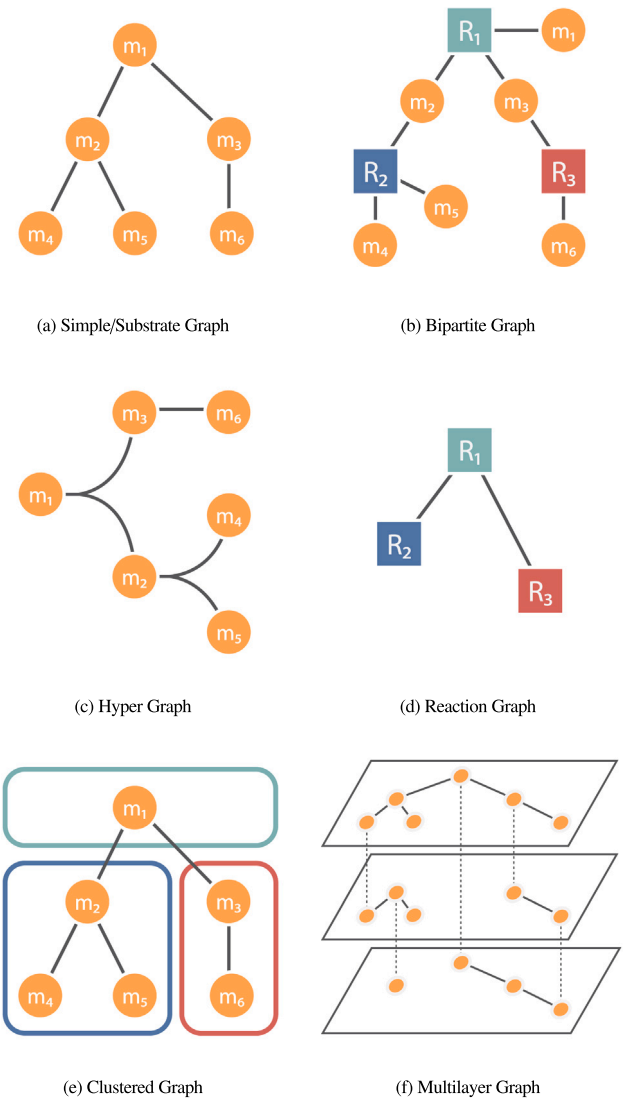



**Fig. 3.** Graph models that are commonly incorporated in biological as well as general network analysis, i.e. (a) *simple* or *substrate graphs*, (b) *bipartite graphs*, (c) *hypergraphs*, (d) *reaction graphs*, (e) *clustered graphs*, and (f) *multilayer graphs*. Across all graphs, with the exception of (f), $V = \{m_1, m_2, m_3, m_4, m_5, m_6\}$ which can be placed in one of three groups $\{R_1, R_2, R_3\}$.

which multiple metabolites are involved (Fig. 3(c)). 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 [36].

**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.

### 3.5. 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 Fig. 3(d)). Reaction graphs have been used in computational models in early research efforts [45].

**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 [46].

### 3.6. 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 [42]. An example can be found later in Fig. 14, while the color boxes show the grouping information.

**Definition:** A clustered graph (Fig. 3(e)) is a simple graph $G$ with additional grouping information. Each $v \in V$ in a clustered graph belongs to one or more clusters $c \in C = \{c_1, c_2, \ldots, c_k\}$.

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 $V$ [47,48]. 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 [49].

### 3.7. Multilayer graphs

We can transform a cluster graph with complex grouping information to 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 concept are also used in current research [22,50]. As summarized by Kivela et al. [50], the above terms can be re-framed 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 [50]. Layers in multilayer (Fig. 3(f)) networks are used to describe the corresponding relationships, where each of which records the property of the corresponding relationships. In this STAR, we follow the formal definition by McGee et al. [51].

**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 $L = \{l_1, l_2, l_3, \dots, l_p\}$, 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 [50]. In practice, we can use multilayer graphs as a unified graph structure because the graphs described in Section 3.1–3.6 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 Figs. 16 and 17.

### 3.8. Graph data structure in practice

Classical biological analytics tools cover the subsets of aforementioned graph data structures. Cytoscape [52] 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 [53]. The COBRA Toolbox [54] integrates MATLAB for quantitative prediction of cellular biochemical networks. The toolkit incorporates CellDesigner [55], which is a graphical editor designed for gene-regulatory and biochemical networks. Reactome [56] is an open-source and peer-reviewed knowledge-base of biomolecular pathways. The visualization tool ReactomeFIViz [57] implements several functions for network-based data analysis and the graphs are extended from Cytoscape [52]. BioCyc [58] is pathway and genome databases, that integrate Pathway Tools [59] which facilitates genome data management, systems biology, and omics data analysis. WikiPathways [60] is a community-based biological pathways database, which integrates PathVisio [61], 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.

## 4. Graph 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 preselect, 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*.

### 4.1. 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 [2]. 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. [23] as

$$density = \frac{|E|}{|E|_{max}}. \tag{1}$$

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 [2]. To make this concept more tangible, consider Fig. 4(a) 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 [62], (ii) the actual metric upon which the vertex clustering is based [63,64], or (iii) a means of comparing and evaluating the identified clusters' structures [65]. 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 metric to compare clusters and even entire graphs. For example, both Koutrouli et al.'s *NORMA* tool [66] and Theodosiou et al. [67], 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 [68], or allow users to filter clusters from the visualization whose density is below some user-set threshold filtering of identified clusters [69,70]. 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. [71] 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* [72] 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.

### 4.2. 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 [73], or to reduce the dimensionality of the problem [74]. This ranking can based on some selected structural features of the network, by utilizing one of the many available measures of centrality [23]. 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 [75], genes and motifs important within genetic regulatory networks [76], or proteins crucial for the network's overall robustness to perturbation in metabolic engineering [77]; each of which requires a different type of centrality. Consider, as an example, Fig. 4(b), in which, given some

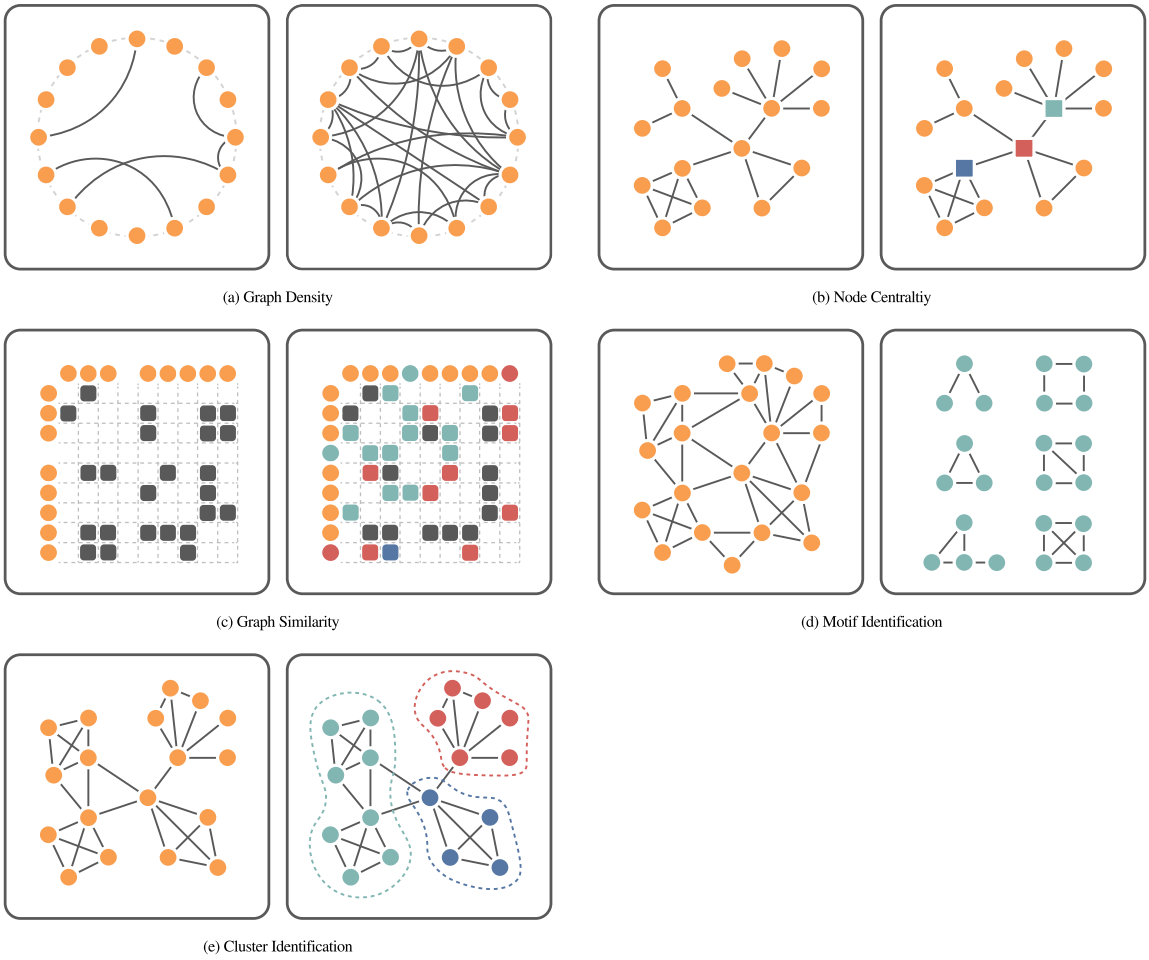



**Fig. 4.** Visual illustration of the 5 discussed graph analysis metrics and approaches, i.e Visual illustration of the 5 discussed graph analysis metrics and approaches, i.e (a) *graph density*, (b) *node centrality*, (c) *graph similarity*, (d) *motif identification*, and (e) *(node) cluster identification*. (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 some second graph (rights) 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.

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 [78] 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.

***Degree centrality*** [79] simply measures how well-connected a vertex $v \in V$ is:

$$C_{degree}(v) = deg(v), \tag{2}$$

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 [23]. For example, Zotenko et al. [75] 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 [71] *miRNET* and Kuijpers et al.'s [40] *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 $V \setminus \{v\}$ [80]:

$$C_{closeness}(v) = \frac{1}{\sum_{w \in V \setminus \{v\}} dist(v,w)}, \tag{3}$$

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 [2,23], though it also has seen use in gene regulatory networks [76]. For example, da Silva et al. [81] 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 [77] and evolution [82] of metabolic pathways.

***Betweenness Centrality*** [83] quantifies how many of the shortest paths between any two vertices pass through the vertex $v$ of interest:

$$C_{Betweenness}(v) = \frac{\sigma_{wu}(v)}{\sigma_{wu}}, \tag{4}$$

where $\sigma_{wu}$ 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 [2,23]. An example thereof can be seen in Joy et al.'s [84] 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 [85] and essential genetic drug targets [86].

***Eigenvector Centrality*** quantifies how connected a vertex is to important other vertices [3], i.e. the more important a vertex's neighbors, the higher its centrality value. Bonacich [87,88] 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 [89]. It has also been used to identify as well as predict gene–disease associations [90] as well as study connectivity patterns in human brain fMRI data [91].

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

$$C_{Eccentricity}(v) = \frac{1}{\max_{w \in V}(dist(v,w))} \tag{5}$$

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 [78]. 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 [92].

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* [93,94], *DynoVis* [40], and *OmicsNet* [95], 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* [39] features very similar ranking and highlighting functionality to *VANTED*, but tailored to miRNA data specifically. Within the context of protein–protein interaction networks, *cytoNCA* [96] provides eight centrality measures with which to screen and highlight vertices of interest. Lastly, *CentiBin* [97] 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* [98] 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.

### 4.3. 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, Fig. 4(c): 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 [99], which Sugiyama et al. [100] 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. [101] for an overview of available approaches. Owing to their predominance in biological application, we opt to focus on comparing networks with known-vertex correspondences, i.e. two or more networks that share the same set of vertices [102]. More specifically, in this section we aim to provide an overview of three such known-vertex correspondence approaches discussed by Tantardini et al. [101], 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 [101].

The ***DeltaCon Measure*** computes pairwise vertex similarities within each network, which can then be used to calculate the similarity between the two networks [103]. First, the similarity between vertices $v$ and $u$ is defined as $S[vu] = [I + \epsilon^2 D - \epsilon A]^{-1}$, where $\epsilon$ 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 [101]. 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 [103]. 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 [103]:

$$d = d_{Matusita}(S_1, S_2) = \sqrt{\sum_{v \in V} \sum_{w \in V} (\sqrt{S_1[vu]} - \sqrt{S_2[vu]})^2} \tag{6}$$

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 [103].

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 [104], measures similarity based on the differences in cut edge weights across all possible bi-partitions of $S$ and $T$ in graphs $G_1$ and $G_2$ [101]. 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:

$$d(G_1, G_2) = \max_{S \subset V} \frac{1}{|V|} \left| e_{G_1}(S, S^c) - e_{G_2}(S, S^c) \right| \tag{7}$$

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 $G$ [105]. 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 [101].

Despite many visualization tools being published for the purpose of comparing biological networks, for example *NetConfer* [106], *CoExpNetViz* [107], or *DynoVis* [40], such tools primarily provide a *visual* comparison of networks and their meta-data only. Owing to adjacency matrices' straightforward visualization and interpretation, a number of tools, across various biochemical domains, have been published which 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 MultiPiles* [108] provides an interactive visualization of adjacency matrices; viewable juxtaposed or interchangeably [21]. On the other hand, for brain connectivity studies, Yang et al. [109] visualize brain subgraphs across experimental conditions as side-by-side adjacency matrices. Similarly, New et al. [110] 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* [111], 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* [67], 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.

### 4.4. Motif identification

Motifs are structures that are statistically over-represented relative to some null model which describe a particular pattern of interaction between vertices in a network [2]. Sometimes described as the "building blocks" of larger networks [112], 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 [113]. To appreciate motifs' nature as the building blocks of a network, consider Fig. 4(d): 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 [113,114].

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 sub-graphs increases exponentially with the maximal motif size [114]. 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 [115,116]. A discussion of these algorithmic approaches is beyond the scope of this STAR, but the reader is referred to Ehrlich et al. [117] 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 is 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 [114]. 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" [114]. 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 [118]. Often called "bootstrapping" approaches, these rely on repeatedly drawing a random (sub-)set of vertices from the original graph [119], or specific sub-regions of the original network [120].

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. [121] identify motifs in order to better understand both gene expression and functional associations. For metabolic networks, Droste et al. [122] 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. [93] and Zander et al. [123] provide motif identification to facilitate more complete topology-based analyses of multi-omics networks. In Protein–Protein Interaction Networks, Spirin et al. [68] identify motifs in order to discover novel molecular modules in protein–protein interaction networks. Within the context of genomic variation graphs, Guarracino et al. [124] utilized motif identification to understand complex pangenomic relationships between sequences of DNA. Lastly, Al-Awami et al. [125]'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 [67,126,127]

### 4.5. Cluster identification

Clusters are vertices grouped together based on common properties [23]. This grouping attempts to ensure objects within the cluster are as homogeneous as possible while separating objects with different properties into distinct clusters [128]. Consider, as an example, the clustering shown in Fig. 4(e): 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 [129]. 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 [128]; though only a handful are realistically applicable to the large problems encountered in modern biological applications [2]. We highlight three noteworthy categories of techniques; (i) *k*-partition approaches, (ii) hierarchical approaches, and (iii) density-based methods [130,131]. 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 [23].

***k*-Partition Clustering**, as the name implies, aims to partition the graph's vertices into *k* clusters [23]. 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 [132]. 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 [131]. 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 [131,133]. 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. [134] 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. [135] *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 [2]. In such a case, ***Hierarchical Clustering*** approaches can be useful to create multilevel groupings based on vertex similarities [136]. 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 [23]. 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 [2].

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. [137] implemented hierarchical clustering to allow users to explore different relationships of a node across clustering hierarchies in dynamic gene expression data. Similarly, Bartell et al. [138] and Varemo et al. [139] provide similar clustering though for gene-set and SNP datasets respectively. Beyond molecular networks, Riaz et al. [140]'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 [131]. As illustrated by Kriegel et al. [141], 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 [141]. 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 arbitrary numbers of clusters of arbitrary shape [141]. 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 produced visualizations. First, (hierarchical) clustering is a common choice to determine the order of rows and columns of matrix representations of graphs [110,142–144], 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. [145,146] 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. [147] made use of edge-bundling and clustering of nodes-clustering to "declutter" visualizations of metabolic networks. However, clustering is also a useful tool to provide guidance to users. For example, Herandez de Diego et al. [148] 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. [149]'s tool *Caleydo*, which features several different clustering techniques for pre-filtering and highlighting gene expression data.

## 5. Network visualization

Abstract graph drawing algorithms form the core of network visualization methods, whether they deal with biological networks [18] or networks in other domains. A graph drawing algorithm takes as input a graph $G = (V, E)$ (Section 3), 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 [150,151] and surveys [152].

### 5.1. Straight-line node-link diagrams

One of the least constrained and most popular styles is the straight-line node-link diagram [40,153,154], where vertices can be placed anywhere in the plane and edges are drawn as straight-line segments, see Fig. 5(a). 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 [12,155], in which we search for a low-energy configuration or the definition of a stress function to be minimized [156]. 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

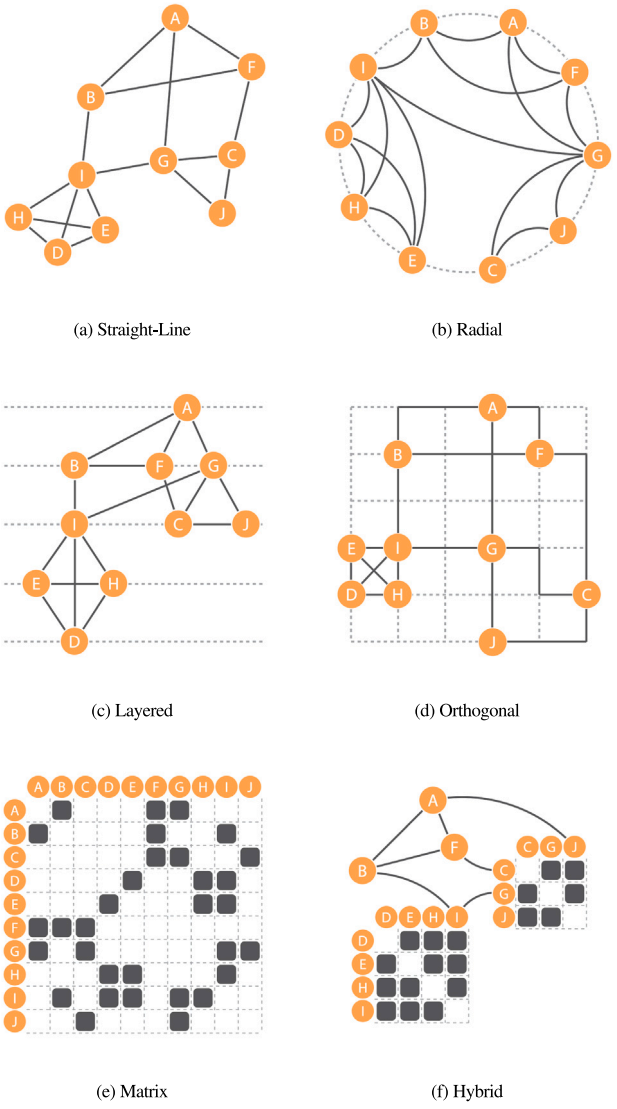



**Fig. 5.** Six typical layout styles of graph drawings common in both general and biological network visualization, i.e. (a) *straight-line node-link diagrams*, (b) *radial node-link diagrams*, (c) *layered node-link diagrams*, (d) *orthogonal (schematic) graph drawings*, (e) *adjacency matrix representations*, and (f) *hybrid 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.

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 [157–159].

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 [160–163]. 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 [164]. 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 [165], 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 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 [13], owing to the many local minima present in the underlying physical model [12].

Force-directed straight-line node-link diagrams are one of the most popular approaches to network visualization across biochemical domains (Fig. 2), likely owing to their broad applicability, computational scalability, and intuitive interpretability [13]: from *gene-regulatory networks* [71,139], through *protein interaction networks* [111] (Fig. 13) and *metabolic pathways* [154], to *multi-omics networks* [40]. Likely owing to their poor visual scalability with increasing numbers of edges and nodes [13], 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* [166] and *GeRNET* [126] (Fig. 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 results, *xiNET* [167] 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.

### 5.2. Radial graph drawings

A more constrained layout style is the radial (or circular) drawing style [166,168], which restricts all vertex positions to a given circle, see Fig. 5(b). 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 [169]. 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 [170]. 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* [167], 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* [168] 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* [71,171] provides a concentric option in which miRNA fragments are grouped into, for example, functional modules [172], 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 [173] or *GeneNetVR*'s time-series view [174]. 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 [126,166]. However, as circular drawings scale poorly with increasing numbers of vertices and edges [175], edge bundling is frequently also employed in order to provide less cluttered drawings and also make group-level connectivity clearer.

### 5.3. 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 Fig. 5(c). 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 [176]. 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 [177] 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 [178,179]. If the graph to be drawn is actually a tree, specific tree drawing algorithms can be applied [180], 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 (Fig. 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 [181]. Commonly, these vertex layers are arranged horizontally and encode either simply the depth within the tree, seen in *MEGAN* [182] or *Phylo.io* [183], or, for example, the time domain, as seen in *NextStrain* [184]. However, these layers need not be arranged horizontally or vertically only. Tools such as the *LifeMap* [185] 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* [181] uses this additional space to visualize the network's metadata.

### 5.4. 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 [186], see Fig. 5(d) 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 [49]. 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 [187].

Together with *straight-line node-link diagrams*, schematic graph drawings form one of the most common visual representations of biological networks (Fig. 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 are familiar to them. For example, *Entourage* [188] aims to produce augmented drawings identical to those found in the *KEGG: Kyoto Encyclopedia of Genes and Genomes* [41] in order to allow domain experts to better explore and analyze their data. Similarly, Blucher et al. [189] 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* [167] 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. [190] 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* [125] which represents the brain's 3D topology as a subway map. Similarly, Beyer et al.'s *Sequence Tube Maps* [191] (Fig. 19) also make use of a visual metro-map metaphor, albeit to communicate genome sequencing reads more intuitively. Lastly, *Metabopolis* [49] (Fig. 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.

### 5.5. Matrix representations

A very different way to show networks is to use a matrix representation [121] 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 Fig. 5(e). 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 [142,143]. 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 [192], 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 (Fig. 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 [121] or instead of [193] (Fig. 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* [149], *VizEpis* [166], or *3Omics* [194], provide the experimental-data-derived correlation or co-expression network displayed as a (clustered) heatmap [121]; though, similar correlation network visualizations can be found in *genomic variation graph* [138] and *brain network* [195] 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* [108] 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 [110] (Fig. 9) allows for the identification and comparison of multiple subgraphs in side-by-side triangular matrix heatmap representations.

### 5.6. 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* [196] and *ChordLink* [146]. 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 Fig. 5(f). 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 (Fig. 2). However, Henry et al.'s *NodeTrix* [196] 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 [109]. Similarly, *ChordLink* [145,146] has also found adoption in biological network visualization [127,144], as it allows for an uncluttered view into the connectivity between groups of vertices to identify broader trends. Lastly, *Caleydo* [149] provides a 3D view linking multiple *schematic graph representations* with *straight-line node-links* between their nodes.

### 5.7. Discussion

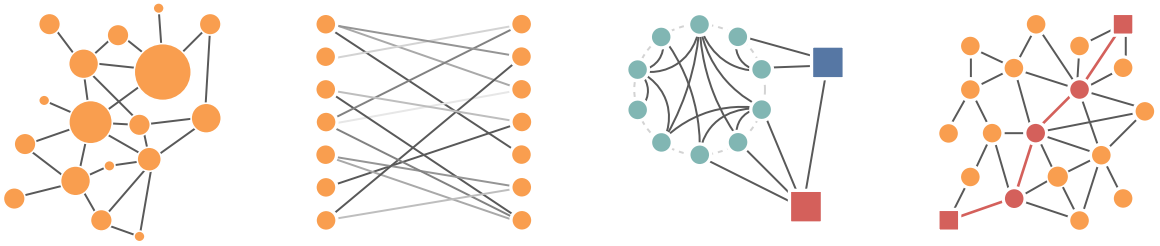
With the different types of network visualizations discussed above (recall Fig. 5), a key question to a domain expert wanting to visualize their biological network data is to select the most suitable of these visualization styles. This decision depends on multiple aspects of the data, in particular, (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 5.1, 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, also radial layouts may be a way of placing all vertices in a well-structured manner along a circle, 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 [142]. 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 5.6. This, however, usually comes at the cost of requiring significantly more implementation and adaptation effort than using general-purpose layout algorithms.

## 6. 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 consequentially 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 [197–199] and ones tailored to general network analysis [200]. Focusing on particular types of networks, there are even more specialized taxonomies detailing, for example, temporal network evolution [201,202], networks of overlapping sets [203], and biological pathways [22].

In this work we scope the tasks used in the analyzed tools using a taxonomy adapted from Lee et al. [200]. 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 [197], on the one hand, are not specific enough to the objective of network visualization. On the other hand, more specific network task taxonomies [201], 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. [22] is mentioned as well, as the application domain is of high relevance.



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

**Fig. 6.** Illustrative examples of Lee et al.'s [204] 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* [181], (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).

### 6.1. Object focus of tasks

Lee et al. [200] taxonomy describes the notion of focal objects for the definition of tasks. A focal point describes a central element of the network visualization which 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*** (Fig. 6(a)). Many biological network visualizations feature nodes as their focal object. For example, many applications featuring *simple straight-line node-link diagrams*, such as *STRING* [205], utilize nodes to display proteins in a *simple straight-line node-link diagrams*, 8.1.2 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* [182] and *NeuroLines* [125] 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*** (Fig. 6(b)). 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 [205] 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 [126] — 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 [109].

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 (Fig. 6(c)). *iTOL* [181], 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. [137]'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 (Fig. 6(d)). 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. [188], for example, show important paths in different related pathways based on experimental data. Metabopolis [49] 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 [191], for example, displays identical regions as nodes and variations in sequence as diverging paths.

### 6.2. Complex task focus

Besides the fundamental objects that are the subject of a given analysis task, Lee et al. [204] further categorized graph analysis tasks into groups of complex tasks.

#### 6.2.1. 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. [22] also similarly describe *Attribute tasks* for biological networks. *PaintOmics* [148], 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* [95] 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* [111], on the one hand, uses categorical color coding to communicate types of interactions in a protein–protein interaction network. *STRING* [205], on the other, utilizes its color coding to describe evidence types for its protein–protein interactions. Beyond color, tree visualizations like *Nextstrain* [184] 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.

#### 6.2.2. 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* [67], 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* [94] or *Omix* [122] 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. [22] define the *Grouping* task as a *Relationship* task. There are several examples of applications, such as *ClusterViz* [153], 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. [137] focuses on the temporal evolution of clusters in networks. A task that is also reflected in the specialized taxonomies of Ahn et al. [201] and Kerracher et al. [202], e.g. the analysis of *Growh and Contraction* of groups or as a *Q4* task respectively. With *MultiPiles* [108] Bach et al. aggregate matrices by similarity to a customizable degree, allowing interactive investigation of the effect of abstracting the data.

Determining 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 [205] or Biolinker [111]. Fining neighbors is also relevant for gene-regulatory networks [206] or heterogeneous regulation networks [95,173], 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 Section 6.2.3, 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 [49,94,188] and genome graphs [124,191,207]. 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. [22]. 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* [111] also feature pathfinding, to locate interactions between two proteins.

#### 6.2.3. 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. [204] 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 (Fig. 2). The majority of tools opt to visualize networks using some form of node-link diagram (Fig. 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 amplifies this effect. In *miRNet2.0* [71] 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* [124] 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* [148] on the other hand uses an abstracted but otherwise conventional node-link diagram in which each pathway represented by a single node in combination with node colorings to give an overview over a set of pathways. Another example is the visualization by Cruz et al. [137] which 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. [22] also describe *Comparison* task, based on *Attributes* of the graph entities. One such example is Phylo.io [183] which was created explicitly to compare different phylogenetic trees using a side-by-side view. *Cerebral* [134] 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.

## 7. Human–computer interactions in biological network visualization

To perform the tasks outlined in Section 6, 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. [16]. 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 (Fig. 2), but instead, point out the connections to the described tasks and mention examples for the corresponding interaction types.

### 7.1. 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 [16]. A very prevalent realization of this select interaction in network visualization is the simple (cursor-based) selection of vertices or edges (see Fig. 7, (a)). One such example can be seen in *xiNET* [167], 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* [126] allows for clicking and dragging to select a subset of the displayed entities. Fuijiwara et al. [195] 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 [111] or genes [208], 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*.

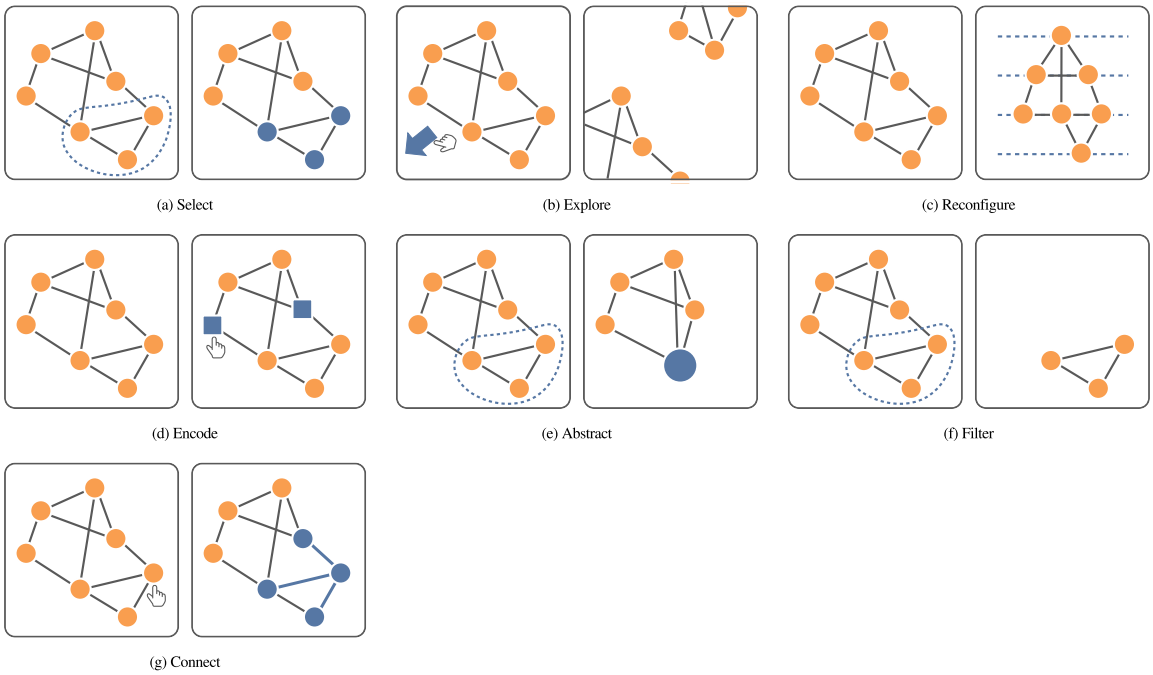



**Fig. 7.** Human–Computer Interactions in Biological Network Visualization. (a)–(g) Examples for seven general categories of interaction techniques [16], i.e. (a) select, (b) explore, (c) reconfigure, (d) encode, (e) abstract, (f) filter, and (g) connect. All hypothetical user interactions and changes are shown in blue.

### 7.2. Explore

***Exploration*** interaction techniques enable users to navigate varying subsets of data entities, e.g., depending on the *object-focus* [16], 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 BioLinker [111], 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 StemCellNet [208], 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 some 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 [93,127,134,137,167,209] (Fig. 7(b)). 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.

### 7.3. 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 [16]. *PathwayMatrix* [193] 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. [193] 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 Fig. 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, Fig. 7(c). 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* [134], 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* [153], 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. [71] 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.

### 7.4. Encode

While the *reconfigure interaction* maintains the data encoding, changing the ***encoding*** adjusts the visual representation, offering a different view of the data Fig. 7(d) [16]. 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. [110] utilize this approach to communicate time-dependent changes in gene co-expression networks, as shown in Fig. 9. The network is first represented as a matrix [110]. 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* [134] provides sliders for users to adjust edge curviness, label density, and group

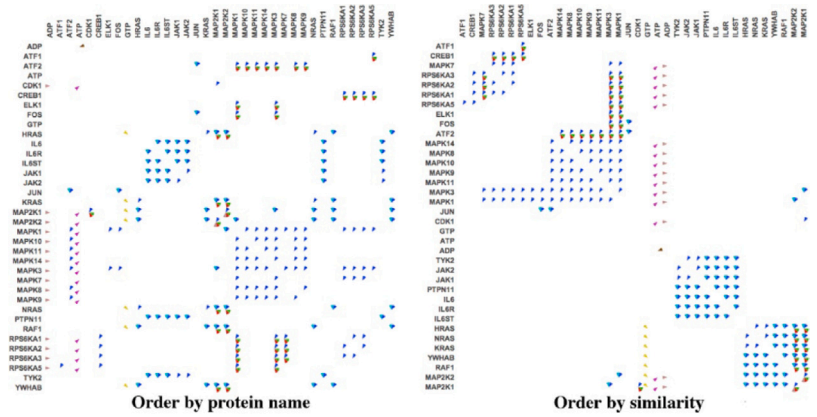



**Fig. 8.** Two different protein orderings in the RAF-MAP Kinase Cascade pathway [193].

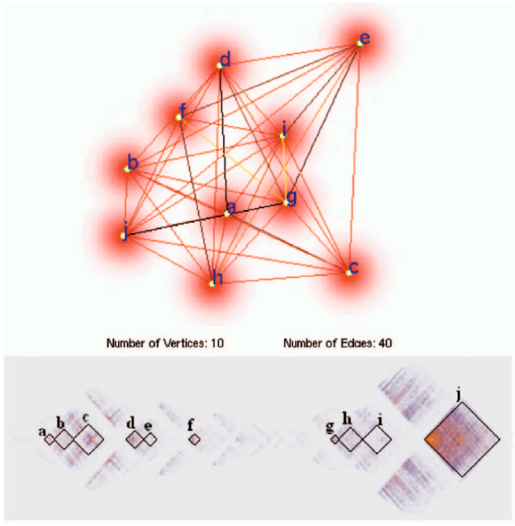



**Fig. 9.** A 2D LoD graph created from brushed BTD belt selections to show correlations among BTD structures [110].

label size to enable the user to find a proper encoding scheme that best fits their research questions and tasks. *NAP* [67] 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*.

### 7.5. 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 7(e) [16]. 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 [38,67,93,126,127,134,173]. For example, Fujiwara et al. [195] 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* [149], 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*.

*Tool-tips* 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 [137,140,195], an associated gene annotation [126], or other biological information [40,210]. Other tooltips, like the ones in BioLinker [111], show information of the hovered node and the statistics of its immediate neighbors (see Fig. 10).

### 7.6. 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 [16], as shown in Fig. 7(f). 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. [209] allow users to remove biologically uninteresting proteins using extendable filters to simplify the visual representation of the network. In MAPPS [140] users can filter the visualization on pathway length. Other applications offer filtering with multiple options. In StemCellNet [208], the filter options include filtering by species, interaction types, co-expressions, and evidence. Additionally, all unconnected vertices (post-filtering) are removed to avoid orphan vertices cluttering the networks. Some applications also offer filtering based on graph topology. Zhou et al. [211], 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*.

### 7.7. Connect

Finally, we discuss ***Connect*** interactions. This interaction technique highlights the related data items or shows hidden, but contextually relevant, data items [16]. 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 (Fig. 7(g)). Jianu et al. [209], 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 *NetWorkAnalyst* [144], users can click on a vertex to zoom in to see its position and interconnectivity within the current

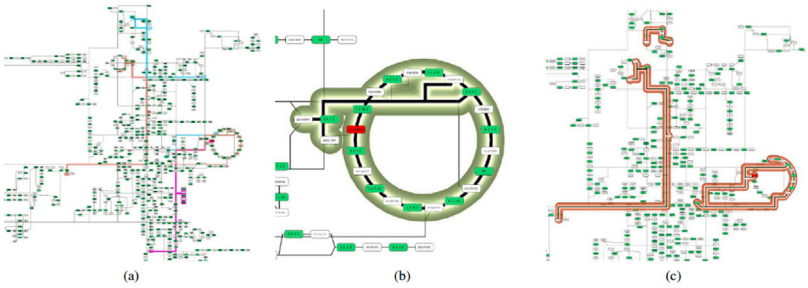



**Fig. 10.** Retrieving pathway information using dedicated focus+context interaction techniques [147].

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. [137], hovering over a vertex with the mouse highlights other connected vertices. Fujiwara et al. [195] 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 BioLinker [111]: 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* .

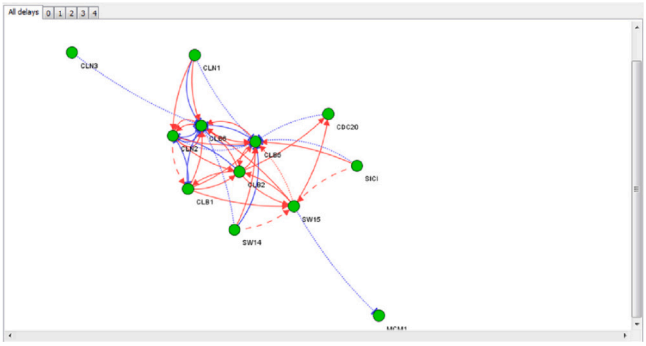



**Fig. 11.** *Node-link* diagram of *GeRNet* [126].

## 8. Application areas & existing tools

As different biological and medical sub-domains deal with different data, they also deal with different research questions, and challenges [212]. 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 [188]. 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 [134].

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 to 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.

### 8.1. 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.

#### 8.1.1. 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 [213]. Additionally, they can reduce the amount of experimental preliminary studies by providing computational entry points for experimental biological research [213].

One example for the visualization of gene regulatory networks is *GeRNet* [126]. *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 Fig. 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 [206], while others include matrix visualizations [121].

#### 8.1.2. 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 [68, 69]. Additionally such networks can be analyzed further to find the more loosely defined *modules*, which are used to attribute larger-scale cellular functions [70,214]. The main source of information regarding

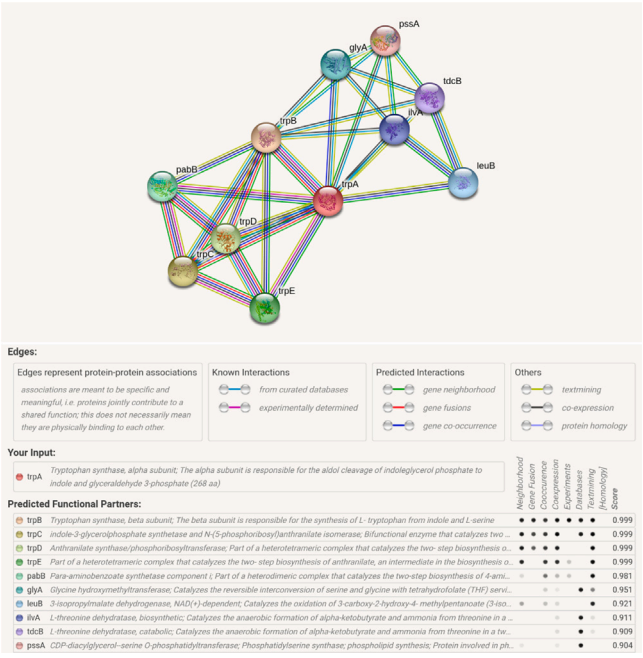



**Fig. 12.** An exemplary association network from the STRING homepage [205].

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 [215]. Databases like the *STRING* [205] database, thus not only keep 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 [205]. 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 Fig. 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 Fig. 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*).

Another such system, designed to visualize protein–protein interaction networks is *Biolinker* [111]. *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 Fig. 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 Fig. 13).

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 [175], inferring PPI-Networks from existing data [216], showing *node-link* egographs [217] for comparing the similarity of PPI-Networks [218], or approaches using *matrix* views instead of *node-link* diagrams [193].

### 8.1.3. 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* [219] or, with added interactivity, *ReconMap* [42]. Additionally, databases like KEGG [41], BioCyc [220] or Reactome [221], 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* [49], 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 layouts them by solving a constrained *floor-plan problem* afterward intra- and inter-block connections are drawn finalizing the visualization (see Fig. 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* [222] 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 Fig. 15). *VisANT*, however, is not the only platform offering systems biology functionality, with many tools offering avenues to generate hypotheses (e.g. [210,223]).

Another task, common for metabolic pathway networks, is the comparison of different experimental conditions or the comparisons of similar pathway segments in the grand ensemble of metabolic pathways. This task introduces an additional challenge for graph visualizations, as they require visualization to facilitate graph comparisons.

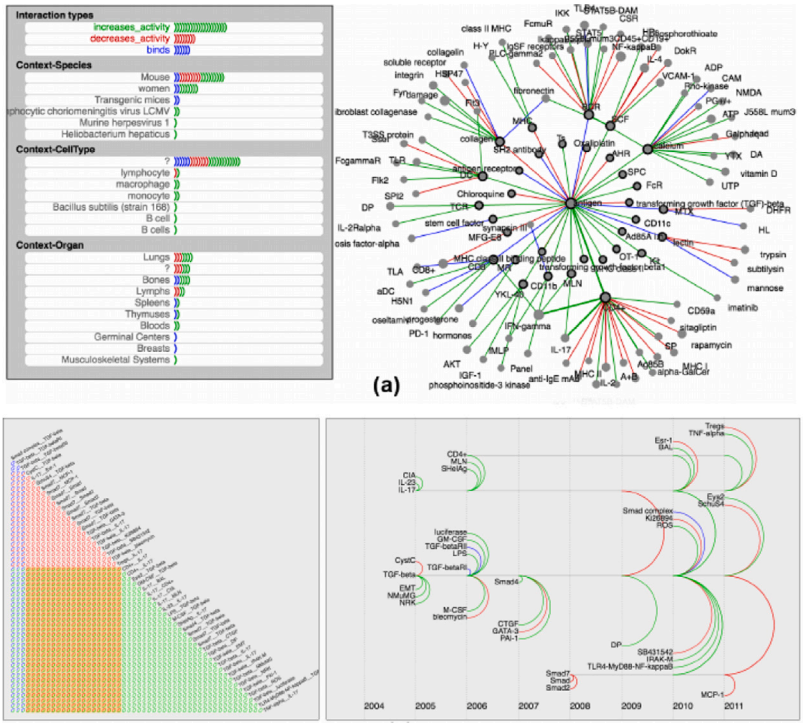



**Fig. 13.** Main view of Biolinker, Conflict matrix and literature arcs of Biolinker [111].

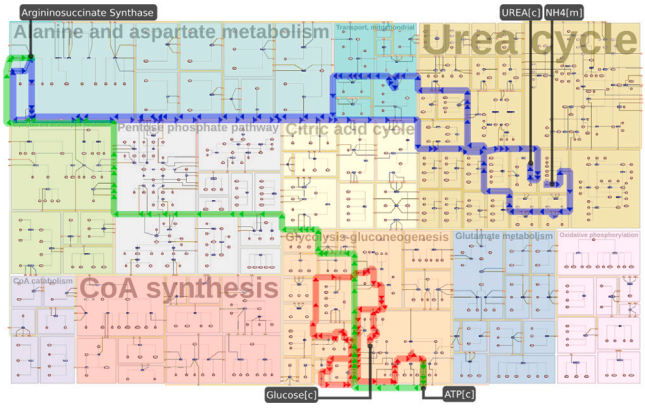



**Fig. 14.** Metabolic pathway visualization in Metabopolis [49].

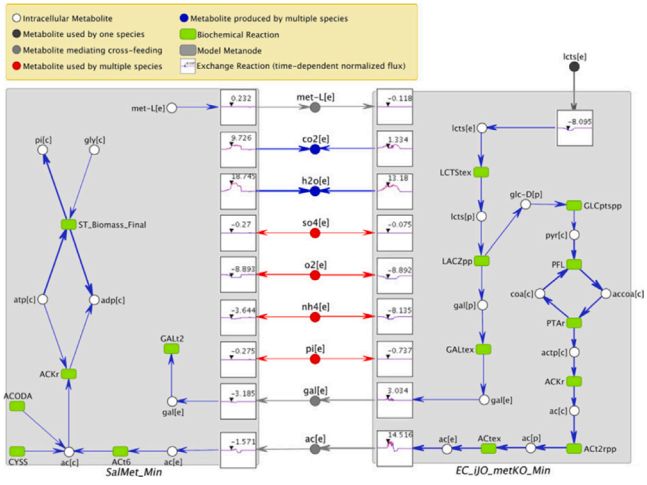



**Fig. 15.** Visualization of two interacting bacterial metabolic pathways in VisANT [222].

Entourage [188], 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 [134] 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. [122,140]). Also, many tools for biological networks feature add-ons allowing the visualization of metabolic pathways (e.g. [224]). Additionally, some of the general-purpose network visualization frameworks like yEd [225] and Cytoscape [226] can be used to draw metabolic pathways (e.g. [189,190]).

### 8.1.4. 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 [227–229] other biological domains, for example phylogenetics [230] or plant physiology [123] 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* [95] features the application of such a multi-layer layout. Using gene, protein, miRNA,

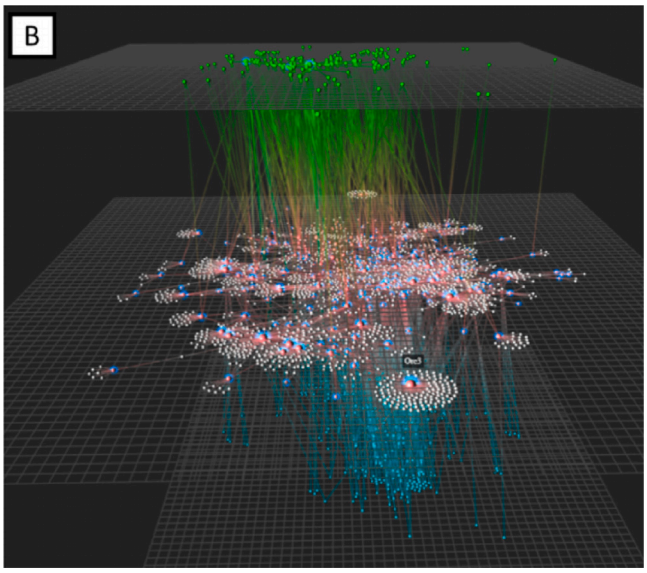



**Fig. 16.** Multilayernetwork of Transcription factors, proteins and miRNAs (top to bottom) in *OmicsNet* [95].

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 factor and miRNAs targeting proteins can be visualized as additional layers (Fig. 16).

As this kind of multi-layer visualization often introduces the need for 3D visualizations it also introduces its drawbacks [51,197]. Thus other multi-omics applications aim to integrate the different omics layers in a two-dimensional visualization [148,194]. *PaintOmics 3* [148] 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 Fig. 17).

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 [93]. *VANTED* offers a combination of manual and automatic layout functions (e.g. schematic node-link diagram as seen in Fig. 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 [93]) 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. [153]).

### 8.2. 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 [232] 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* [191] 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* [207] tool results in more compact layouts, as it does not adhere to the strict "train track" metaphor but allows for arbitrary, curved layouts. *ODGI* [124] includes a similar visualization. It is optimized for scalability and can create layouts for whole pangenome graphs. The recent review by Eizenga et al. [231] provides further details on pangenome graphs and compares the abovementioned visualization approaches (Fig. 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 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 [233]. 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* [234], *QTLNetworkR* [235], or *solQTL* [236]. However, this simple visualization often does not capture the full interaction network. Jiang et al. [237] 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 Section 8.1. The *Reveal* tool by Jäger et al. [238] 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 Fig. 20). The tool was originally developed for the BioVis Challenge 2011, which focused on QTL expression data [138]. For this challenge, Paquette and Lum clustered SNPs and visualized the results as *node-link*-diagrams using the *Iris* tool (Ayasdi, Inc.) [138]. Here, the vertices represent clusters, which are connected by edges if they share at least one SNP.

### 8.3. 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 evolutional difference. Besides the analysis of different species, phylogenies are also used to derive taxonomies of different species. The online tool *Lifemap* by de Vienne [185] 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 [184], 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. [239] can compute phylogenetic trees for hundreds

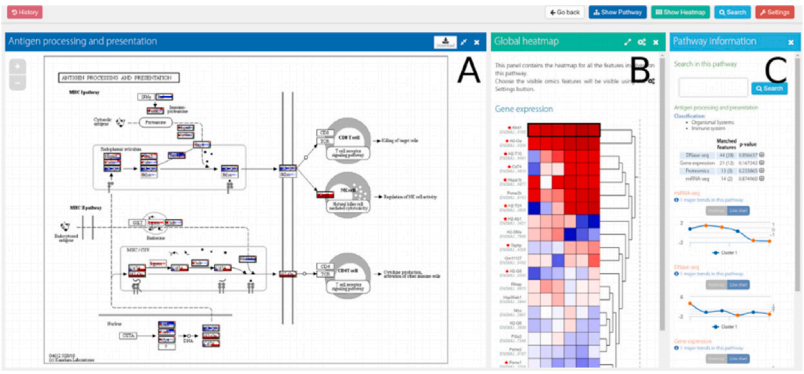



**Fig. 17.** Multilayernetwork of Transcription factors, proteins and miRNAs (top to bottom) [148].

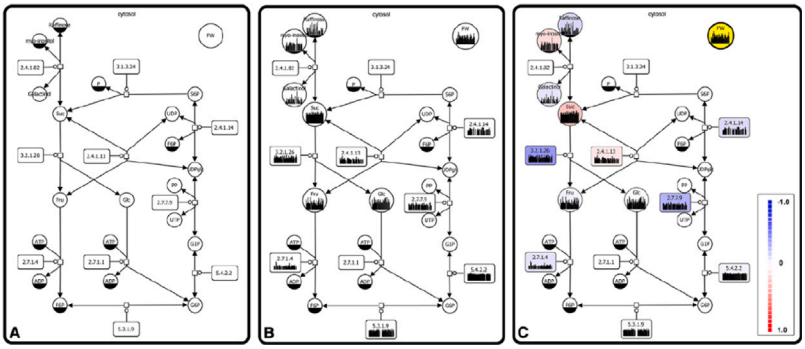



**Fig. 18.** Metabolic pathway visualization [93].

of thousands of samples. The results can be visualized using various applications such as the online tool *iTOL* [181], 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. [240], which allows for the comparison of the structure of large phylogenetic trees with hundreds of thousands of vertices. The trees are visualized as dendrogram and a specialized *focus-and-context* technique is used to ensure the visibility of differences. Similarly, the online tool *Phylo.io* by Robinson et al. [183] also uses juxtaposition of dendrograms for a comparative visual analysis. Another example is the metagenome analyzer *MEGAN6* by Huson et al. [182], 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 (see Fig. 21).

### 8.4. 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 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. [108] 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 Fig. 22(a)). In addition, various visualization techniques are used, such as node-trix [109] (see Fig. 22(b)), which utilizes a *hybrid matrix-node-link* layout, a visual metaphor called NeuroLine [125], and edge bundling [241] 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 [195].

*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 combine visualization techniques and visual analysis approaches with mathematical methods for the calculation of interspecies interactions in ecology [242].

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 [243]. Because of the need to interpret graphs along with heterogeneous data, Zhou et al. [244] 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.

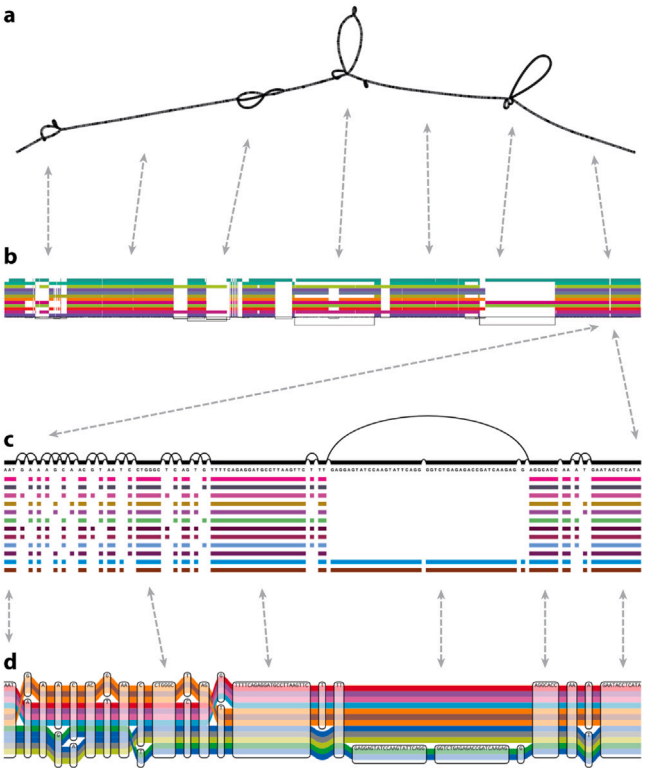



**Fig. 19.** Genomic variation graph visualizations from different applications: (a) Bandage, (b) ODGI viz, (c) VG viz, (d) Sequence Tube Map. The gray arrows highlight the correspondence and scalability of the different approaches.
*Source:* Image source: [231].

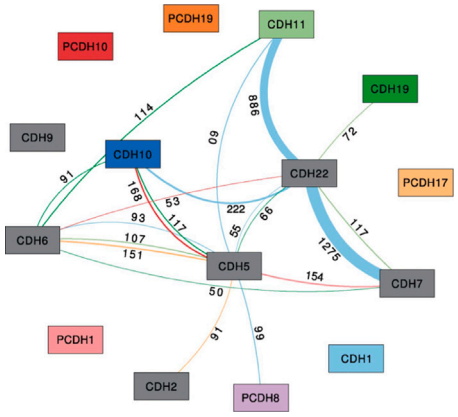



**Fig. 20.** The Reveal tool derives gene association graphs from eQTL data and visualizes them as node-link diagrams.
*Source:* Image source: [238].

## 9. Research challenge and discussion

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 [33] underlying biological network visualizations across domains (Fig. 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 (Fig. 2), as well as

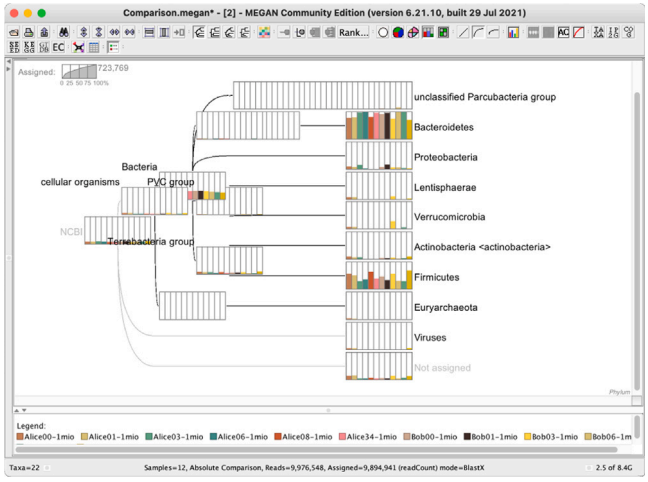



**Fig. 21.** Screenshot of the comparative phylogenetic tree visualization offered by MEGAN6.
*Source:* Image courtesy of D. Huson.

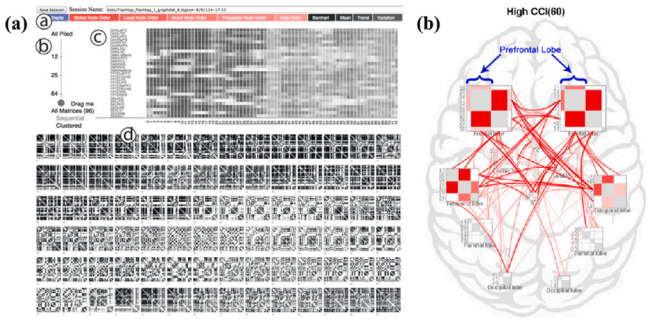



**Fig. 22.** (a) Screenshot of the MultiPiles application by Bach et al. image from [108]. (b) NodeTrix visualization of the brain network by Yang et al..
*Source:* Image from [109].

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.

### 9.1. Beyond schematic and straight-line node-link diagrams

Our collection of papers (Fig. 2) indicates that most biological graphs are either (manually) drawn schematic representations, as seen in *NeuroLines* [125], *ODGI* [124], or *Sequence Tube Maps* [191], or drawn as force-directed node-link diagrams, as seen in *miDerma* [245], *xiNET* [167], or *DORMAN* [154]. 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 [41,42].

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 [12], forming so-called "*hairballs*" [15]. 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 [13] or edge–angle-ratios [246], but also empirically evaluated in several comparative,

quantitative user studies [192,247]. 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 [12] which in turn impact user performance [248,249].

Beyond their larger sizes and higher densities, many biological networks also feature node groupings, based on their cellular location [134], (predicted) function [135], biochemical pathway association [188], or some data-driven clustering [250]. 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 [20,21]. At least within the context of biological network visualization, these options seem to have not 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 meta-data associated with both nodes and edges, application-specific hybrid approaches (Fig. 2) could allow for a more effective visual exploration of networks and the multivariate data attached to them [19].

### 9.2. Incorporating uncertainty in network visualizations

Uncertainty can arise at many stages in biological network visualization, from data collection to the conducted automated numerical analyses [251]. Especially important are the uncertainties and weights attached to networks obtained from online knowledge graph repositories, such as *KEGG* [41] or *Reactome* [56], 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 [252], despite an understanding that it can improve awareness, analysis results, and thrust [253]. 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* [205], 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.

### 9.3. 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 few employed or supported numerical graph analysis (Fig. 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* [193] or *Kiwi* [139], in order to guide users to biologically meaningful parts of the data,
2. a means of simplifying a network's visual representation, such as the coloring and grouping of nodes, e.g. Cruz et al.'s visualization of dynamic time-series data [137], the breaking up of a graph into simpler sub-components, as seen in *ClusterViz* [153] or New et al.'s visualization of dynamic co-expression data [110], or the bundling of edges in node-link diagrams, e.g. Lambert et al.'s visualization of metabolic networks [147],
3. a mechanism for data-driven hypotheses generation, as seen in *Caleydo* [149] and *Mapps* [140].

Even straightforward and comparatively simple graph descriptive metrics, such as vertex centrality, or graph density were surprisingly uncommon (Fig. 2. When utilized, these metrics are primarily used either for the straightforward ranking/filtering of entities [39,254] and clusters [69,109], or the visual highlighting of nodes [40], 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* [140] for edge prediction in metabolic pathways, and *VANTED* [94] 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.

### 9.4. 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 [48,255], for example, the similarity of biochemical (reaction) topologies across different databases [111], individuals/samples [195], biological compartments [125], pathways [140,188], time points [137, 206], or experimental conditions [134] 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 superimpositioning [21,256]. Within the collected set of literature, most modern tools opt for a simple side-by-side juxtaposition of (sub)graphs, as seen in Cerebral [134]'s visualizing of graphs across experimental conditions. An approach that utilizes both *networks similarity* and supports *graph comparison* is the *MultiPiles* approach by Bach et al. [108]. 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 [109–111,195]. Other approaches, with the exception of *DynoVis* [40], 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 4.3). Second, most visual graph comparison approaches identified here, 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* [93,94], 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.

### 9.5. 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 [257]. 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 [258]. 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 [259,260].

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.

### 9.6. 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 [261–263], the visualization of dynamic biological networks is still in its infancy. While some examples hereof exist, such as Kuijper et al.'s [40] *DynoVis* framework for the visualization of dose-over-time effects in biological networks, Perkins and Daniels' [206] efforts to visualize dynamic gene expression data, or Hartman and Schreiber's [46] 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. [262], 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. [108] *Small Multiples* approach which opts to visualize dynamic networks as an interactive stack of adjacency matrices, or Peng et al.'s [264] *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 [19], such as gene expression fold changes [265], labels of metabolites and reactions [154], or the database origin and type of a particular protein–protein interaction [111]. Many possible solutions to this well-known problem have been put forth [266]. Additionally, within the context of multi-omics data integration and analysis (Section 8.1.4), a domain expert may be faced with so-called multilayer networks [51], in which a network is comprised of multiple, separate inhomogeneous omics networks [148,227,228]. 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.

## 10. Conclusions and limitations

Based on Card et al.'s [33] 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 literatureTasks*. 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 straight-forward 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 paper, such as community detection algorithms and their application to biological networks [267]. Additionally, the scalability of the various visualization and analysis approaches would justify a follow-up survey in and of itself.

Nonetheless, in order to address the identified outstanding and multifaceted challenges, network scientists, bioinformaticians, and domain experts will need to work closely together. This points toward an exciting set of future opportunities for collaborative research.

## CRediT authorship contribution statement

**Henry Ehlers:** Writing – review & editing, Writing – original draft, Visualization, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Nicolas Brich:** Writing – review & editing, Writing – original draft, Visualization, Investigation, Data curation.

**Michael Krone:** Writing – review & editing, Supervision, Investigation, Data curation. **Martin Nöllenburg:** Writing – review & editing, Writing – original draft, Investigation, Data curation. **Jiacheng Yu:** Writing – review & editing, Writing – original draft, Investigation, Data curation. **Hiroaki Natsukawa:** Writing – original draft, Investigation. **Xiaoru Yuan:** Conceptualization. **Hsiang-Yun Wu:** Writing – review & editing, Writing – original draft, Supervision, Investigation, Data curation, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix. Approach to collect papers in this STAR

First, we fetched papers relevant to biological network visualization, by utilizing various journals' APIs in conjunction with logical search on paper content, keywords, and citations. On the one hand, to find surveys from the last 20 years, we utilized the following query: *("visualization"* OR *"visualization"* OR *"visual computing"* OR *"visual analytics")* AND *("graph"* OR *"network"* OR *"pathway")* AND *("protein-protein interaction"* OR *"Signal Transduction Networks"* OR *"gene expression"* OR *"gene co-expression"* OR *"gene regulatory"* OR *"gene regulation"* OR *"multivariate"* OR *"metabolic"* OR *"metabolomic"* OR *"interactomics"* OR *"multilayer"* OR *"multi-layer"* OR *"biology"* OR *"biological")* AND *("review"* OR *"survey"* OR *"state of the art"* OR *"overview")*. On the other hand, for tools and application papers from the past 10 years, we utilized this query: *("visualization"* OR *"visualization"* OR *"visual computing"* OR *"visual analytics")* AND *("graph"* OR *"network"* OR *"pathway")* AND *("protein-protein interaction"* OR *"Signal Transduction Networks"* OR *"gene expression"* OR *"gene co-expression"* OR *"gene regulatory"* OR *"gene regulation"* OR *"multivariate"* OR *"metabolic"* OR *"metabolomic"* OR *"interactomics"* OR *"multilayer"* OR *"multi-layer"* OR *"biology"* OR *"biological")* AND *("tool"* OR *"application"* OR *"software")*. Second, we screened the collected body of literature to classify papers into highly, intermediately, and hardly relevant to the topic at hand. Note that duplicated works, such as a conference paper also published as a journal one, were removed. Third, for papers that are categorized as highly relevant, an annotation procedure was conducted according to a pre-discussed table, which contains features of the components in our survey (Fig. 2).

## Data availability

Data will be made available on request.

## References


[1] Chuang H-Y, Hofree M, Ideker T. A decade of systems biology. Annu Rev Cell Dev Biol 2010;26(1):721–44 .

[2] Koutrouli M, Karatzas E, Paez-Espino D, Pavlopoulos GA. A guide to conquer the biological network era using graph theory. Front Bioeng Biotechnol 2020;8:34.

[3] Pavlopoulos GA, Wegener A-L, Schneider R. A survey of visualization tools for biological network analysis. BioData Min 2008;1(1):12, Publisher: Springer Nature.

[4] Krassowski M, Das V, Sahu SK, Misra BB. State of the field in multi-omics research: From computational needs to data mining and sharing. Front Genet 2020;11. Publisher: Frontiers.

[5] Albrecht M, Kerren A, Klein K, Kohlbacher O, Mutzel P, Paul W, et al. On open problems in biological network visualization. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, et al., editors. In: Graph drawing, vol. 5849, Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 256–67, Series Title: Lecture Notes in Computer Science.

[6] Nishizuka Y. The role of protein kinase C in cell surface signal transduction and tumour promotion. Nature 1984;308(5961):693–8.

[7] Davidson EH, Rast JP, Oliveri P, Ransick A, Calestani C, Yuh C-H, et al. A genomic regulatory network for development. Science 2002;295(5560):1669–78, Publisher: American Association for the Advancement of Science.

[8] Gehlenborg N, O'Donoghue SI, Baliga NS, Goesmann A, Hibbs MA, Kitano H, et al. Visualization of omics data for systems biology. Nature Methods 2010;7(3 Suppl):S56–68.

[9] Scala A, D'Agostino G. Networks of networks: the last frontier of complexity. In: Springer complexity, Springer; 2014.

[10] Barabási A-L, Gulbahce N, Loscalzo J. Network medicine: a network-based approach to human disease. Nature Rev Genet 2011;12(1):56–68, Number: 1 Publisher: Nature Publishing Group.

[11] Figueiras A. Towards the understanding of interaction in information visualization. In: 2015 19th international conference on information visualisation. 2015, p. 140–7.

[12] Kobourov S. Force-directed drawing algorithms. In: Tamassia R, editor. Handbook of graph drawing and visualization. CRC Press; 2014.

[13] Kobourov SG, Pupyrev S, Saket B. Are crossings important for drawing large graphs? In: Bayro-Corrochano E, Hancock E, editors. Progress in pattern recognition, image analysis, computer vision, and applications, vol. 8827. Cham: Springer International Publishing; 2014, p. 234–45, Series Title: Lecture Notes in Computer Science.

[14] Suh A, Hajij M, Wang B, Scheidegger C, Rosen P. Persistent homology guided force-directed graph layouts. IEEE Trans Vis Comput Graphics 2020;26(1):697–707, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[15] Yoghourdjian V, Dwyer T, Klein K, Marriott K, Wybrow M. Graph thumbnails: Identifying and comparing multiple graphs at a glance. IEEE Trans Vis Comput Graphics 2018;24(12):3081–95, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[16] Yi JS, Kang Ya, Stasko J, Jacko J. Toward a deeper understanding of the role of interaction in information visualization. IEEE Trans Vis Comput Graphics 2007;13(6):1224–31, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[17] Keim D. Information visualization and visual data mining. IEEE Trans Vis Comput Graphics 2002;8(1):1–8.

[18] Bachmaier C, Brandes U, Schreiber F. Biological networks. In: Tamassia R, editor. Handbook of graph drawing and visualization. CRC Press; 2014, p. 621–51.

[19] Nobre C, Meyer M, Streit M, Lex A. The state of the art in visualizing multivariate networks. Comput Graph Forum 2019;38(3):807–32, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13728.

[20] Vehlow C, Hasenauer J, Theis FJ, Weiskopf D. Visualizing edge-edge relations in graphs. In: 2013 IEEE Pacific visualization symposium. 2013, p. 201–8.

[21] Ehlers H, Marin D, Wu H-Y, Raidou R. 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, vol. 1 - GRAPP, HUCAPP and IVAPP: IVAPP. Rome; 2024, p. 697–708.

[22] Murray P, McGee F, Forbes AG. A taxonomy of visualization tasks for the analysis of biological pathway data. BMC Bioinformatics 2017;18(Suppl 2):21, Publisher: Springer Science and Business Media LLC.

[23] Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, Aerts J, et al. Using graph theory to analyze biological networks. BioData Min 2011;4(1):10.

[24] Motaleb Faysal MA, Arifuzzaman S. A comparative analysis of large-scale network visualization tools. In: 2018 IEEE international conference on big data. 2018, p. 4837–43.

[25] Filipov V, Arleo A, Miksch S. Are we there yet? A roadmap of network visualization from surveys to task taxonomies. Comput Graph Forum 2023;n/a(n/a). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14794.

[26] Wu H-Y, Nöllenburg M, Viola I. 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. 2019.

[27] O'Donoghue SI, Gavin A-C, Gehlenborg N, Goodsell DS, Hériché J-K, Nielsen CB, et al. Visualizing biological data—now and in the future. Nature Methods 2010;7(3):S2–4, Number: 3 Publisher: Nature Publishing Group.

[28] Dimara E, Perin C. What is interaction for data visualization? IEEE Trans Vis Comput Graphics 2020;26(1):119–29, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[29] Sacha D, Stoffel A, Stoffel F, Kwon BC, Ellis G, Keim DA. Knowledge generation model for visual analytics. IEEE Trans Vis Comput Graphics 2014;20(12):1604–13, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[30] Lu Y, Garcia R, Hansen B, Gleicher M, Maciejewski R. The state-of-the-art in predictive visual analytics. Comput Graph Forum 2017;36(3):539–62, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13210.

[31] Moreland K. A survey of visualization pipelines. IEEE Trans Vis Comput Graphics 2012;19(3):367–78, Publisher: IEEE.

[32] Wang X-M, Zhang T-Y, Ma Y-X, Xia J, Chen W. A survey of visual analytic pipelines. J Comput Sci Tech 2016;31(4):787–804.

[33] Card M. Readings in information visualization: Using vision to think. Morgan Kaufmann; 1999, Google-Books-ID: wdh2gqWfQmgC.

[34] Vulliard L, Menche J. Complex networks in health and disease. Syst Med 2021;26–33.

[35] Hancock J, Menche J. Structure and function in complex biological networks. In: Wolkenhauer O, editor. Systems medicine. Oxford: Academic Press; 2021, p. 8–16.

[36] Wang L, Dash S, Ng CY, Maranas CD. A review of computational tools for design and reconstruction of metabolic pathways. Synth Syst Biotechnol 2017;2(4):243–52.

[37] Sakr S, Bonifati A, Voigt H, Iosup A, Ammar K, Angles R, et al. The future is big graphs: a community view on graph processing systems. Commun ACM 2021;64(9):62–71.

[38] Bonnet E, Viara E, Kuperstein I, Calzone L, Cohen DPA, Barillot E, et al. NaviCell Web Service for network-based data visualization. Nucleic Acids Res 2015;43(W1):W560–5, Publisher: Oxford University Press (OUP).

[39] Fan Y, Siklenka K, Arora SK, Ribeiro P, Kimmins S, Xia J. miRNet - dissecting miRNA-target interactions and functional associations through network-based visual analysis. Nucleic Acids Res 2016;44(W1):W135–41.

[40] Kuijpers TJM, Wolters JEJ, Kleinjans JCS, Jennen DGJ. DynOVis: a web tool to study dynamic perturbations for capturing dose-over-time effects in biological networks. BMC Bioinformatics 2019;20(1):417, Publisher: Springer Science and Business Media LLC.

[41] Kanehisa M, Goto S. KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 2000;28(1):27–30.

[42] Noronha A, Daníelsdóttir AD, Gawron P, Jóhannsson F, Jónsdóttir S, Jarlsson S, et al. ReconMap: an interactive visualization of human metabolism. Bioinformatics (Oxford, England) 2017;33(4):605–7.

[43] Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 2003;19(4):524–31 .

[44] Demir E, Cary MP, Paley S, Fukuda K, Lemer C, Vastrik I, et al. The BioPAX community standard for pathway data sharing. Nature Biotechnol 2010;28(9):935–42.

[45] Steuer R, Junker BH. Computational models of metabolism: Stability and regulation in metabolic networks. In: Rice SA, editor. Advances in chemical physics. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2008, p. 105–251.

[46] Hartmann A, Schreiber F. Integrative analysis of metabolic models – from structure to dynamics. Front Bioeng Biotechnol 2015;2.

[47] Frishman Y, Tal A. Dynamic drawing of clustered graphs. In: IEEE symposium on information visualization. 2004, p. 191–8.

[48] Symons S, Nieselt K. MGV: a generic graph viewer for comparative omics data. Bioinformatics 2011;27(16):2248–55, Publisher: Oxford University Press (OUP).

[49] Wu H-Y, Nöllenburg M, Sousa F, Viola I. Metabopolis: Scalable network layout for biological pathway diagrams in urban map style. BMC Bioinformatics 2019;20.

[50] Kivelä M, Arenas A, Barthelemy M, Gleeson JP, Moreno Y, Porter MA. Multilayer networks. J Complex Netw 2014;2(3):203–71.

[51] McGee F, Ghoniem M, Melançon G, Otjacques B, Pinaud B. The state of the art in multilayer network visualization. Comput Graph Forum 2019;38(6):125–49 .

[52] Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, et al. Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Res 2003;13(11):2498–504.

[53] Wiese R, Eiglsperger M, Kaufmann M. yFiles: Visualization and automatic layout of graphs. In: Goos G, Hartmanis J, Van Leeuwen J, Mutzel P, Jünger M, Leipert S, editors. In: Graph drawing, vol. 2265, Berlin, Heidelberg: Springer Berlin Heidelberg; 2002, p. 453–4.

[54] Heirendt L, Arreckx S, Pfau T, Mendoza SN, Richelle A, Heinken A, et al. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. Nat Protoc 2019;14(3):639–702.

[55] Funahashi A, Morohashi M, Kitano H, Tanimura N. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. BIOSILICO 2003;1(5):159–62.

[56] Vastrik I, D'Eustachio P, Schmidt E, Joshi-Tope G, Gopinath G, Croft D, et al. Reactome: a knowledge base of biologic pathways and processes. Genome Biol 2007;8(3):R39.

[57] Wu G, Dawson E, Duong A, Haw R, Stein L. ReactomeFIViz: a Cytoscape app for pathway and network-based data analysis. F1000Research 2014;3:146.

[58] Paley S, Karp PD. The BioCyc metabolic network explorer. BMC Bioinformatics 2021;22(1):208.

[59] Paley S, Billington R, Herson J, Krummenacker M, Karp PD. Pathway tools visualization of organism-scale metabolic networks. Metabolites 2021;11(2):64.

[60] Martens M, Ammar A, Riutta A, Waagmeester A, Slenter DN, Hanspers K, et al. WikiPathways: connecting communities. Nucleic Acids Res 2021;49(D1):D613–21 .

[61] Kutmon M, Van Iersel MP, Bohler A, Kelder T, Nunes N, Pico AR, et al. PathVisio 3: An extendable pathway analysis toolbox. In: Murphy RF, editor. PLoS Comput Biol 2015;11(2):e1004085.

[62] Bader GD, Hogue CW. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 2003;4(1):2.

[63] Li M, Li D, Tang Y, Wu F, Wang J. CytoCluster: A cytoscape plugin for cluster analysis and visualization of biological networks. Int J Mol Sci 2017;18(9):1880, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[64] Hu Z, Hung J-H, Wang Y, Chang Y-C, Huang C-L, Huyck M, et al. VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. Nucleic Acids Res 2009;37(suppl_2):W115–21.

[65] Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 2008;9(1):559.

[66] Koutrouli M, Karatzas E, Papanikolopoulou K, Pavlopoulos GA. NORMA: The network makeup artist — A web tool for network annotation visualization. Genom Proteom Bioinform 2021.

[67] Theodosiou T, Efstathiou G, Papanikolaou N, Kyrpides NC, Bagos PG, Iliopoulos I, et al. NAP: The Network Analysis Profiler, a web tool for easier topological analysis and comparison of medium-scale biological networks. BMC Res Notes 2017;10(1):278.

[68] Spirin V, Mirny LA. Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci USA 2003;100(21):12123–8.

[69] Nepusz T, Yu H, Paccanaro A. Detecting overlapping protein complexes in protein-protein interaction networks. Nature Methods 2012;9(5):471–2.

[70] Chen J, Yuan B. Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics (Oxford, England) 2006;22(18):2283–90.

[71] Chang L, Zhou G, Soufan O, Xia J. miRNet 2.0: network-based visual analytics for miRNA functional analysis and systems biology. Nucleic Acids Res 2020;48(W1):W244–51, Publisher: Oxford University Press (OUP).

[72] Ebbels TMD, Buxton BF, Jones DT. springScape: visualisation of microarray and contextual bioinformatic data using spring embedding and an 'information landscape'. Bioinformatics 2006;22(14):e99–107.

[73] Zhang J, Li W, Zeng M, Meng X, Kurgan L, Wu F-X, et al. NetEPD: A network-based essential protein discovery platform. Tsinghua Sci Technol 2020;25(4):542–52, Conference Name: Tsinghua Science and Technology.

[74] Roffo G, Melzi S. Feature selection via eigenvector centrality. Contrib atti convegno 2016;4(1):13.

[75] Zotenko E, Mestre J, O'Leary DP, Przytycka TM. Why do hubs in the yeast protein interaction network tend to be essential: Reexamining the connection between the network topology and essentiality. PLoS Comput Biol 2008;4(8):e1000140, Publisher: Public Library of Science.

[76] Koschützki D, Schreiber F. Centrality analysis methods for biological networks and their application to gene regulatory networks. Gene Regul Syst Biol 2008;2:GRSB.S702, Publisher: SAGE Publications Ltd STM.

[77] Ma H-W, Zeng A-P. The connectivity structure, giant strong component and centrality of metabolic networks. Bioinformatics 2003;19(11):1423–30.

[78] Ghasemi M, Seidkhani H, Tamimi F, Rahgozar M, Masoudi-Nejad A. Centrality measures in biological networks. Curr Bioinform 2014;9(4):426–41.

[79] Bonacich P. Power and centrality: A family of measures. Am J Sociol 1987;92(5):1170–82, Publisher: University of Chicago Press.

[80] Newman MEJ. 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: Newman M, editor. Networks: an introduction. Oxford University Press; 2010.

[81] da Silva MR, Ma H, Zeng A-P. Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. Proc IEEE 2008;96(8):1411–20, Conference Name: Proceedings of the IEEE.

[82] Mazurie A, Bonchev D, Schwikowski B, Buck GA. Evolution of metabolic network organization. BMC Syst Biol 2010;4(1):59.

[83] Freeman LC. A set of measures of centrality based on betweenness. Sociometry 1977;40(1):35–41, Place: US Publisher: American Sociological Assn.

[84] Joy MP, Brock A, Ingber DE, Huang S. High-betweenness proteins in the yeast protein interaction network. J Biomed Biotechnol 2005;2005(2):96–103.

[85] Li J, Zhang W. Identification of crucial metabolites/reactions in tumor signaling networks. Netw Biol 2013;3(4):13.

[86] Durón C, Pan Y, Gutmann DH, Hardin J, Radunskaya A. Variability of betweenness centrality and its effect on identifying essential genes. Bull Math Biol 2019;81(9):3655–73.

[87] Bonacich P. Some unique properties of eigenvector centrality. Social Networks 2007;29(4):555–64.

[88] Bonacich P. Factoring and weighting approaches to status scores and clique identification. J Math Sociol 1972;2(1):113–20. http://dx.doi.org/10.1080/0022250X.1972.9989806, Publisher: Routledge.

[89] Negre CFA, Morzan UN, Hendrickson HP, Pal R, Lisi GP, Loria JP, et al. Eigenvector centrality for characterization of protein allosteric pathways. Proc Natl Acad Sci 2018;115(52):E12201–8, Publisher: Proceedings of the National Academy of Sciences.

[90] Özgür A, Vu T, Erkan G, Radev DR. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. Bioinformatics 2008;24(13):i277–85.

[91] Lohmann G, Margulies DS, Horstmann A, Pleger B, Lepsien J, Goldhahn D, et al. Eigenvector centrality mapping for analyzing connectivity patterns in fMRI data of the human brain. PLOS ONE 2010;5(4):e10232, Publisher: Public Library of Science.

[92] Jalili M, Salehzadeh-Yazdi A, Gupta S, Wolkenhauer O, Yaghmaie M, Resendis-Antonio O, et al. Evolution of centrality measurements for the detection of essential proteins in biological networks. Front Physiol 2016;7.

[93] Rohn H, Junker A, Hartmann A, Grafahrend-Belau E, Treutler H, Klapperstück M, et al. VANTED v2: a framework for systems biology applications. BMC Syst Biol 2012;6(1):139.

[94] Hartmann A, Jozefowicz AM. VANTED: A tool for integrative visualization and analysis of -omics data. Methods Mol Biol 2018;1696:261–78.

[95] Zhou G, Xia J. OmicsNet: a web-based tool for creation and visual analysis of biological networks in 3D space. Nucleic Acids Res 2018;46(W1):W514–22.

[96] Tang Y, Li M, Wang J, Pan Y, Wu F-X. CytoNCA: A cytoscape plugin for centrality analysis and evaluation of protein interaction networks. Biosystems 2015;127:67–72.

[97] Junker BH, Koschützki D, Schreiber F. Exploration of biological network centralities with CentiBiN. BMC Bioinformatics 2006;7(1):219.

[98] Xia J, Wishart DS. Metpa: a web-based metabolomics tool for pathway analysis and visualization. Bioinformatics 2010;26(18):2342–4.

[99] Emmert-Streib F, Dehmer M, Shi Y. Fifty years of graph matching, network alignment and network comparison. Inform Sci 2016;346–347:180–97.

[100] Sugiyama M, Ghisu ME, Llinares-López F, Borgwardt K. Graphkernels: R and python packages for graph comparison. Bioinformatics 2018;34(3):530–2 .

[101] Tantardini M, Ieva F, Tajoli L, Piccardi C. Comparing methods for comparing networks. Sci Rep 2019;9(1):17557.

[102] Soundarajan S, Eliassi-Rad T, Gallagher B. A guide to selecting a network similarity method. In: Proceedings of the 2014 SIAM international conference on data mining. Society for Industrial and Applied Mathematics; 2014, p. 1037–45.

[103] Koutra D, Shah N, Vogelstein JT, Gallagher B, Faloutsos C. DELTACON: Principled massive-graph similarity function with attribution. In: ACM transactions on knowledge discovery from data, vol. 10, 2016, p. 1–43.

[104] Frieze A, Kannan R. Quick approximation to matrices and applications. Combinatorica 1999;19(2):175–220.

[105] Liu Q, Dong Z, Wang E. Cut based method for comparing complex networks. Sci Rep 2018;8(1):5134, Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science;Statistics Subject_term_id: computer-science;statistics.

[106] Nagpal S, Baksi KD, Kuntal BK, Mande SS. NetConfer: a web application for comparative analysis of multiple biological networks. BMC Biol 2020;18(1):53.

[107] Tzfadia O, Diels T, De Meyer S, Vandepoele K, Aharoni A, Van de Peer Y. CoExpNetViz: Comparative co-expression networks construction and visualization tool. Front Plant Sci 2016;6.

[108] Bach B, Henry-Riche N, Dwyer T, Madhyastha T, Fekete J-D, Grabowski T. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. Comput Graph Forum 2015;34(3):31–40, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12615.

[109] Yang X, Shi L, Daianu M, Tong H, Liu Q, Thompson P. Blockwise human brain network visual comparison using NodeTrix representation. IEEE Trans Vis Comput Graph 2017;23(1):181–90.

[110] New J, Kendall W, Huang J, Chesler E. Dynamic visualization of coexpression in systems genetics data. IEEE Trans Vis Comput Graphics 2008;14(5):1081–95, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[111] Dang T, Murray P, Forbes A. BioLinker: Bottom-up exploration of protein interaction networks. In: 2017 IEEE Pacific visualization symposium. 2017, p. 265–9.

[112] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. Science 2002;298(5594):824–7.

[113] Patra S, Mohapatra A. Review of tools and algorithms for network motif discovery in biological networks. IET Syst Biol 2020;14(4):171–89, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-syb.2020.0004.

[114] Wong E, Baur B, Quader S, Huang C-H. Biological network motif detection: principles and practice. Brief Bioinform 2012;13(2):202–15 .

[115] Bonnici V, Giugno R, Pulvirenti A, Shasha D, Ferro A. A subgraph isomorphism algorithm and its application to biochemical data. BMC Bioinformatics 2013;14(S7):S13.

[116] Cordella L, Foggia P, Sansone C, Vento M. A (sub)graph isomorphism algorithm for matching large graphs. IEEE Trans Pattern Anal Mach Intell 2004;26(10):1367–72, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[117] Ehrlich H-C, Rarey M. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. WIREs Comput Mol Sci 2011;1(1):68–79, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.5.

[118] Mirshahvalad A, Beauchesne OH, Archambault É, Rosvall M. Resampling effects on significance analysis of network clustering and ranking. PLOS ONE 2013;8(1):e53943, Publisher: Public Library of Science.

[119] Snijders TA, Borgatti SP, et al. Non-parametric standard errors and tests for network statistics. Connections 1999;22(2):161–70, Publisher: Citeseer.

[120] Chen Y, Gel RY, Lyubchich V, Nezafati K. Snowboot: Bootstrap methods for network inference. R J 2019;10(2):95.

[121] Zarnegar A, Jelinek HF, Vamplew P, Stranieri A. 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: Association for Computing Machinery; 2019, event-place: Sydney, NSW, Australia.

[122] Droste P, Nöh K, Wiechert W. Omix – A visualization tool for metabolic networks with highest usability and customizability in focus. Chem Ing Tech 2013;85(6):849–62, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201200234.

[123] Zander M, Lewsey MG, Clark NM, Yin L, Bartlett A, Saldierna Guzmán JP, et al. Integrated multi-omics framework of the plant response to jasmonic acid. Nat Plants 2020;6(3):290–302.

[124] Guarracino A, Heumos S, Nahnsen S, Prins P, Garrison E. ODGI: understanding pangenome graphs. Tech. rep., Genomics Research Centre, Human Technopole, Milan, Italy; 2021, 2021.11.10.467921.

[125] Al-Awami AK, Beyer J, Strobelt H, Kasthuri N, Lichtman JW, Pfister H, et al. NeuroLines: A subway map metaphor for visualizing nanoscale neuronal connectivity. IEEE Trans Vis Comput Graphics 2014;20(12):2369–78, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[126] Dussaut JS, Gallo CA, Cravero F, Martínez MJ, Carballido JA, Ponzoni I. GeRNet: a gene regulatory network tool. Biosystems 2017;162:1–11.

[127] Miao L, Yin R-X, Huang F, Yang S, Chen W-X, Wu J-Z. Integrated analysis of gene expression changes associated with coronary artery disease. Lipids Health Dis 2019;18(1):92, Publisher: Springer Science and Business Media LLC.

[128] Sharma A, Ali HH. Analysis of clustering algorithms in biological networks. In: 2017 IEEE international conference on bioinformatics and biomedicine. 2017, p. 2303–5.

[129] Milenković T, Pržulj N. Uncovering biological network function via graphlet degree signatures. Cancer Inform 2008;6:CIN.S680.

[130] Wiwie C, Baumbach J, Röttger R. Comparing the performance of biomedical clustering methods. Nature Methods 2015;12(11):1033–8, Number: 11 Publisher: Nature Publishing Group.

[131] Andreopoulos B, An A, Wang X, Schroeder M. A roadmap of clustering algorithms: finding a match for a biomedical application. Brief Bioinform 2009;10(3):297–314.

[132] Likas A, Vlassis N, J. Verbeek J. The global k-means clustering algorithm. Pattern Recognit 2003;36(2):451–61.

[133] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett 2010;31(8):651–66.

[134] Barsky A, Munzner T, Gardy J, Kincaid R. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. IEEE Trans Vis Comput Graphics 2008;14(6):1253–60, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[135] Mildau K, Ehlers H, Oesterle I, Pristner M, Warth B, Doppler M, et al. Tailored mass spectral data exploration using the SpecXplore interactive dashboard. Anal Chem 2024;96(15):5798–806, Publisher: American Chemical Society.

[136] Khan BS, Niazi MA. Network community detection: A review and visual survey. Tech. rep., Department of Computer Science, COMSATS Institute of Information Technology, Islamabad, Pakistan; 2017, p. 39.

[137] Cruz A, P Arrais J, Machado P. Interactive network visualization of gene expression time-series data. In: 2018 22nd international conference information visualisation. Fisciano, Italy: IEEE; 2018, p. 574–80.

[138] Bartlett CW, Yeon Cheong S, Hou L, Paquette J, Yee Lum P, Jäger G, et al. An eQTL biological data visualization challenge and approaches from the visualization community. BMC Bioinformatics 2012;13(S8):S8.

[139] Väremo L, Gatto F, Nielsen J. Kiwi: a tool for integration and visualization of network topology and gene-set analysis. BMC Bioinformatics 2014;15(1):408.

[140] Riaz MR, Preston GM, Mithani A. MAPPS: A web-based tool for metabolic pathway prediction and network analysis in the postgenomic era. ACS Synth Biol 2020;9(5):1069–82, Publisher: American Chemical Society (ACS).

[141] Kriegel H-P, Kröger P, Sander J, Zimek A. Density-based clustering. WIREs Data Min Knowl Discov 2011;1(3):231–40, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.30.

[142] Behrisch M, Bach B, Henry Riche N, Schreck T, Fekete J-D. Matrix reordering methods for table and network visualization. Comput Graph Forum 2016;35(3):693–716.

[143] Liiv I. Seriation and matrix reordering methods: An historical overview. Stat Anal Data Min ASA Data Sci J 2010;3(2):70–91, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sam.10071.

[144] Xia J, Gill EE, Hancock REW. NetworkAnalyst for statistical, visual and network-based meta-analysis of gene expression data. Nat Protoc 2015;10(6):823–44, Publisher: Springer Science and Business Media LLC.

[145] Angori L, Didimo W, Montecchiani F, Pagliuca D, Tappini A. 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. Berlin, Heidelberg: Springer-Verlag; 2019, p. 276–90.

[146] Angori L, Didimo W, Montecchiani F, Pagliuca D, Tappini A. Hybrid graph visualizations with ChordLink: Algorithms, experiments, and applications. IEEE Trans Vis Comput Graphics 2020;1.

[147] Lambert A, Dubois J, Bourqui R. Pathway preserving representation of metabolic networks. Comput Graph Forum 2011;30(3):1021–30, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2011.01951.x.

[148] Hernández-de Diego R, Tarazona S, Martínez-Mira C, Balzano-Nogueira L, Furió-Tarí P, Pappas Jr GJ, et al. PaintOmics 3: a web resource for the pathway analysis and visualization of multi-omics data. Nucleic Acids Res 2018;46(W1):W503–9.

[149] Lex A, Streit M, Kruijff E, Schmalstieg D. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In: 2010 IEEE Pacific visualization symposium. 2010, p. 57–64.

[150] Battista GD, Eades P, Tamassia R, Tollis IG. Graph drawing: Algorithms for the visualization of graphs. 1st ed. USA: Prentice Hall PTR; 1998.

[151] Tamassia R, editor. Handbook of graph drawing and visualization. Discrete mathematics and its applications, Boca Raton: CRC Press, Taylor & Francis Group; 2014.

[152] von Landesberger T, Kuijper A, Schreck T, Kohlhammer J, van Wijk J, Fekete J-D, et al. Visual analysis of large graphs: State-of-the-art and future research challenges. Comput Graph Forum 2011;30(6):1719–49, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2011.01898.x.

[153] Wang J, Zhong J, Chen G, Li M, Wu F-X, Pan Y. ClusterViz: A Cytoscape APP for cluster analysis of biological network. IEEE/ACM Trans Comput Biol Bioinform 2015;12(4):815–22, Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[154] Ozden F, Siper MC, Acarsoy N, Elmas T, Marty B, Qi X, et al. DORMAN: Database of reconstructed MetAbolic networks. IEEE/ACM Trans Comput Biol Bioinform 2021;18(4):1474–80.

[155] Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Softw - Pract Exp 1991;21(11):1129–64.

[156] Gansner ER, Koren Y, North S. Graph drawing by stress majorization. In: Proceedings of the 12th international conference on graph drawing. Berlin, Heidelberg: Springer-Verlag; 2004, p. 239–50.

[157] Ortmann M, Klimenta M, Brandes U. A sparse stress model. J Graph Algorithms Appl 2017;21(5):791–821.

[158] Kruiger JF, Rauber PE, Martins RM, Kerren A, Kobourov S, Telea AC. Graph layouts by t-SNE. Comput Graph Forum 2017;36(3):283–94 .

[159] Hu Y. Efficient, high-quality force-directed graph drawing. Math J 2006;10(1):37–71.

[160] Kwon S, Kim H, Kim HS. Identification of pharmacologically tractable protein complexes in cancer using the R-based network clustering and visualization program MCODER. BioMed Res Int 2017;2017:1016305, Publisher: Hindawi.

[161] Wang Y, Jin Z, Wang Q, Cui W, Ma T, Qu H. DeepDrawing: A deep learning approach to graph drawing. IEEE Trans Vis Comput Graphics 2020;26(1):676–86, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[162] Grötschla F, Mathys J, Veres R, Wattenhofer R. CoRe-GD: A hierarchical framework for scalable graph visualization with GNNs. 2024, arXiv:2402.06706 [cs].

[163] Tiezzi M, Ciravegna G, Gori M. Graph neural networks for graph drawing. IEEE Trans Neural Netw Learn Syst 2024;35(4):4668–81, Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[164] Nielsen SS, Ostaszewski M, McGee F, Hoksza D, Zorzan S. Machine learning to support the presentation of complex pathway graphs. IEEE/ACM Trans Comput Biol Bioinform 2021;18(3):1130–41, Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[165] Gansner ER, Hu Y, Kobourov S. GMap: Visualizing graphs and clusters as maps. In: 2010 IEEE Pacific visualization symposium. 2010, p. 201–8.

[166] Kwon M-S, Lee S, Kim Y, Park T. VizEpis : A visualization and mapping tool for interpreting epistasis. In: 2015 IEEE international conference on bioinformatics and biomedicine. Washington, DC, USA: IEEE; 2015, p. 1363–6.

[167] Combe CW, Fischer L, Rappsilber J. xiNET: cross-link network maps with residue resolution. Mol Cell Proteom 2015;14(4):1137–47, Publisher: Elsevier BV.

[168] Zhu Y, Sun L, Garbarino A, Schmidt C, Fang J, Chen J. PathRings: a web-based tool for exploration of ortholog and expression data in biological pathways. BMC Bioinformatics 2015;16(1):165, Publisher: Springer Science and Business Media LLC.

[169] Baur M, Brandes U. Crossing reduction in circular layouts. In: Hromkovič J, Nagl M, Westfechtel B, editors. Graph-theoretic concepts in computer science. Berlin, Heidelberg: Springer; 2005, p. 332–43.

[170] Krzywinski M, Schein J, Birol İ, Connors J, Gascoyne R, Horsman D, et al. Circos: An information aesthetic for comparative genomics. Genome Res 2009;19(9):1639–45.

[171] Fan Y, Xia J. MiRNet-functional analysis and visual exploration of miRNA-target interactions in a network context. Methods Mol Biol 2018;1819:215–33.

[172] Tong Y, Ru B, Zhang J. miRNACancerMAP: an integrative web server inferring miRNA regulation network for cancer. Bioinformatics 2018;34(18):3211–3.

[173] Chang J, Cho H, Chou H-H. Mango: combining and analyzing heterogeneous biological networks. BioData Min 2016;9(1):25, Publisher: Springer Science and Business Media LLC.

[174] Fernández ÁM, Bongo LA, Pedersen E. GeneNet VR: Interactive visualization of large-scale biological networks using a standalone headset. 2021, arXiv:2109.02937 [cs].

[175] Alanis-Lobato G, Mier P, Andrade-Navarro M. The latent geometry of the human protein interaction network. Bioinformatics 2018;34(16):2826–34.

[176] Binucci C, Chimani M, Didimo W, Liotta G, Montecchiani F. Placing arrows in directed graph drawings. In: Hu Y, Nöllenburg M, editors. Graph drawing and network visualization. Lecture notes in computer science, Cham: Springer International Publishing; 2016, p. 44–51.

[177] Sugiyama K, Tagawa S, Toda M. Methods for visual understanding of hierarchical system structures. IEEE Trans Syst Man Cybern 1981;11(2):109–25, Conference Name: IEEE Transactions on Systems, Man, and Cybernetics.

[178] Jünger M, Mutzel P. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. J Graph Algorithms Appl 1997;1(1):1–25.

[179] Healy P, Nikolov NS. Hierarchical drawing algorithms. In: Tamassia R, editor. Handbook on graph drawing and visualization. CRC Press; 2014.

[180] Schulz H-J. Treevis.net: A tree visualization reference. IEEE Comput Graph Appl 2011;31(6):11–5.

[181] Letunic I, Bork P. Interactive tree of life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic Acids Res 2021;49(W1):W293–6.

[182] Huson DH, Beier S, Flade I, Górska A, El-Hadidi M, Mitra S, et al. MEGAN community edition - interactive exploration and analysis of large-scale microbiome sequencing data. PLoS Comput Biol 2016;12(6):e1004957.

[183] Robinson O, Dylus D, Dessimoz C. Phylo.io : Interactive viewing and comparison of large phylogenetic trees on the web. Mol Biol Evol 2016;33(8):2163–6.

[184] Hadfield J, Megill C, Bell SM, Huddleston J, Potter B, Callender C, et al. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics 2018;34(23):4121–3.

[185] Vienne DMd. Lifemap: Exploring the entire tree of life. PLOS Biol 2016;14(12):e2001624, Publisher: Public Library of Science.

[186] Eiglsperger M, Fekete SP, Klau GW. Orthogonal graph drawing. In: Kaufmann M, Wagner D, editors. In: Drawing graphs, vol. 2025, Berlin, Heidelberg: Springer Berlin Heidelberg; 2001, p. 121–71.

[187] Schulze CD, Spönemann M, von Hanxleden R. Drawing layered graphs with port constraints. J Vis Lang Comput 2014;25(2):89–106.

[188] Lex A, Partl C, Kalkofen D, Streit M, Gratzl S, Wassermann AM, et al. Entourage: Visualizing relationships between biological pathways using contextual subsets. IEEE Trans Vis Comput Graphics 2013;19(12):2536–45, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[189] Blucher AS, McWeeney SK, Stein L, Wu G. Visualization of drug target interactions in the contexts of pathways and networks with ReactomeFIViz. F1000Research 2019;8:908, Publisher: F1000 ( Faculty of 1000 Ltd).

[190] Livigni A, O'Hara L, Polak ME, Angus T, Wright DW, Smith LB, et al. A graphical and computational modeling platform for biological pathways. Nat Protoc 2018;13(4):705–22.

[191] Beyer W, Novak AM, Hickey G, Chan J, Tan V, Paten B, et al. Sequence tube maps: making graph genomes intuitive to commuters. Bioinformatics 2019;35(24):5318–20.

[192] Ghoniem M, Fekete J-D, Castagliola P. A comparison of the readability of graphs using node-link and matrix-based representations. In: IEEE symposium on information visualization. 2004, p. 17–24.

[193] Dang TN, Murray P, Forbes AG. PathwayMatrix: visualizing binary relationships between proteins in biological pathways. BMC Proc 2015;9(S6):S3.

[194] Kuo T-C, Tian T-F, Tseng YJ. 3Omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. BMC Syst Biol 2013;7(1):64, Publisher: Springer Science and Business Media LLC.

[195] Fujiwara T, Chou J-K, McCullough AM, Ranganath C, Ma K-L. A visual analytics system for brain functional connectivity comparison across individuals, groups, and time points. In: 2017 IEEE Pacific visualization symposium. 2017, p. 250–9.

[196] Henry N, Fekete J-D, McGuffin MJ. NodeTrix: a hybrid visualization of social networks. IEEE Trans Vis Comput Graphics 2007;13(6):1302–9, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[197] Munzner T. Visualization analysis and design. New York: A K Peters/CRC Press; 2014.

[198] Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE symposium on visual languages. Boulder, CO, USA: IEEE Comput. Soc. Press; 1996, p. 336–43.

[199] Amar R, Eagan J, Stasko J. Low-level components of analytic activity in information visualization. In: IEEE symposium on information visualization, 2005. IEEE; 2005, p. 111–7.

[200] Lee B, Plaisant C, Parr CS, Fekete J-D, Henry N. Task taxonomy for graph visualization. In: Proceedings of the 2006 AVI workshop on beyond time and errors: novel evaluation methods for information visualization. New York, NY, USA: Association for Computing Machinery; 2006, p. 1–5.

[201] Ahn J-w, Plaisant C, Shneiderman B. A task taxonomy for network evolution analysis. IEEE Trans Vis Comput Graphics 2014;20(3):365–76, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[202] Kerracher N, Kennedy J, Chalmers K. A task taxonomy for temporal graph visualisation. IEEE Trans Vis Comput Graphics 2015;21(10):1160–72, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[203] Rodgers P, Stapleton G, Alsallakh B, Michallef L, Baker R, Thompson S. A task-based evaluation of combined set and network visualization. Inform Sci 2016;367–368:58–79.

[204] Lee JB, Rossi R, Kong X. Graph classification using structural attention. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. London United Kingdom: ACM; 2018, p. 1666–74.

[205] Szklarczyk D, Gable AL, Lyon D, Junge A, Wyder S, Huerta-Cepas J, et al. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res 2019;47(D1):D607–13.

[206] Perkins M, Daniels K. Visualizing dynamic gene interactions to reverse engineer gene regulatory networks using topological data analysis. In: 2017 21st international conference information visualisation. London: IEEE; 2017, p. 384–9.

[207] Wick RR, Schultz MB, Zobel J, Holt KE. Bandage: interactive visualization of de novo genome assemblies. Bioinformatics 2015;31(20):3350–2.

[208] Pinto JP, Reddy Kalathur RK, Machado RSR, Xavier JM, Bragança J, Futschik ME. StemCellNet: an interactive platform for network-oriented investigations in stem cell biology. Nucleic Acids Res 2014;42(Web Server issue):W154–60, Publisher: Oxford University Press (OUP).

[209] Jianu R, Yu K, Cao L, Nguyen V, Salomon AR, Laidlaw DH. Visual integration of quantitative proteomic data, pathways, and protein interactions. IEEE Trans Vis Comput Graphics 2010;16(4):609–20, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[210] Noronha A, Vilaça P, Rocha M. An integrated network visualization framework towards metabolic engineering applications. BMC Bioinformatics 2014;15(1):420, Publisher: Springer Science and Business Media LLC.

[211] Zhou G, Xia J. Using OmicsNet for network integration and 3D visualization. Curr Protoc Bioinf 2019;65(1):e69, Publisher: Wiley.

[212] Reina G, Childs H, Matković K, Bühler K, Waldner M, Pugmire D, et al. The moving target of visualization software for an increasingly complex world. Comput Graph 2020.

[213] Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. Nature Rev Mol Cell Biol 2008;9(10):770–80.

[214] Hartwell LH, Hopfield JJ, Leibler S, Murray AW. From molecular to modular cell biology. Nature 1999;402(6761 Suppl):C47–52.

[215] Skrabanek L, Saini HK, Bader GD, Enright AJ. Computational prediction of protein-protein interactions. Mol Biotechnol 2008;38(1):1–17.

[216] Jiang Y, Liang Y, Wang D, Xu D, Joshi T. A dynamic programing approach to integrate gene expression data and network information for pathway model generation. Bioinformatics 2020;36(1):169–76, Publisher: Oxford University Press (OUP).

[217] Ehlers H, Pahr D, Filipov V, Wu H-Y, Raidou RG. Me! Me! Me! Me! A study and comparison of ego network representations. Computers & Graphics 2024;104123.

[218] Brich N, Harbig T, Paz MW, Nieselt K, Krone M. ProtEGOnist: Visual analysis of interactions in small world networks using ego-graphs. Comput Graph Forum 2024;43(3):e15078, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.15078.

[219] Michal G. Roche - Biochemical pathways. 2017.

[220] Karp PD, Billington R, Caspi R, Fulcher CA, Latendresse M, Kothari A, et al. The BioCyc collection of microbial genomes and metabolic pathways. Brief Bioinform 2019;20(4):1085–93.

[221] Bohler A, Wu G, Kutmon M, Pradhana LA, Coort SL, Hanspers K, et al. Reactome from a WikiPathways perspective. PLoS Comput Biol 2016;12(5):e1004941.

[222] Granger BR, Chang Y-C, Wang Y, DeLisi C, Segrè D, Hu Z. Visualization of metabolic interaction networks in microbial communities using VisANT 5.0. PLoS Comput Biol 2016;12(4):e1004875, Publisher: Public Library of Science.

[223] Funahashi A, Matsuoka Y, Jouraku A, Morohashi M, Kikuchi N, Kitano H. CellDesigner 3.5: A versatile modeling tool for biochemical networks. Proc IEEE 2008;96(8):1254–65, Conference Name: Proceedings of the IEEE.

[224] Aichem M, Czauderna T, Zhu Y, Zhao J, Klapperstück M, Klein K, et al. Visual exploration of large metabolic models. Bioinformatics 2021;37(23):4460–8.

[225] yWorks GmbH. yEd - Graph editor. 2022.

[226] Smoot ME, Ono K, Ruscheinski J, Wang P-L, Ideker T. Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 2011;27(3):431–2, Publisher: Oxford University Press (OUP).

[227] Hasin Y, Seldin M, Lusis A. Multi-omics approaches to disease. Genome Biol 2017;18(1):83.

[228] Subramanian I, Verma S, Kumar S, Jere A, Anamika K. Multi-omics data integration, interpretation, and its application. Bioinform Biol Insights 2020;14:117793221989905.

[229] Huang S, Chaudhary K, Garmire LX. More is better: Recent progress in multi-omics data integration methods. Front Genet 2017;8.

[230] Hultman J, Waldrop MP, Mackelprang R, David MM, McFarland J, Blazewicz SJ, et al. Multi-omics of permafrost, active layer and thermokarst bog soil microbiomes. Nature 2015;521(7551):208–12.

[231] Eizenga JM, Novak AM, Sibbesen JA, Heumos S, Ghaffaari A, Hickey G, et al. Pangenome graphs. Annu Rev Genom Hum Genet 2020;21(1):139–62. http://dx.doi.org/10.1146/annurev-genom-120219-080406.

[232] Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, Dawson ET, et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nature Biotechnol 2018;36(9):875–9, Bandiera_abtest: a Cg_type: Nature Research Journals Number: 9 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational biology and bioinformatics;Genome informatics;Sequencing;Software Subject_term_id: computational-biology-and-bioinformatics;genome-informatics;sequencing;software.

[233] Rifkin SA. Quantitative trait loci (QTL): Methods and protocols, In: Methods in molecular biology, 1st ed. vol. 871, New York, NY, USA: Humana; 2012.

[234] Yang J, Hu C, Hu H, Yu R, Xia Z, Ye X, et al. QTLNetwork: mapping and visualizing genetic architecture of complex traits in experimental populations. Bioinformatics 2008;24(5):721–3.

[235] Zheng W-j, Yang J, Zhu J. QTLNetworkR: an interactive R package for QTL visualization. J Zhejiang Univ Sci B 2010;11(7):512–5.

[236] Tecle IY, Menda N, Buels RM, van der Knaap E, Mueller LA. Solqtl: a tool for QTL analysis, visualization and linking to genomes at SGN database. BMC Bioinformatics 2010;11(1):525.

[237] Jiang L, Shi H, Sang M, Zheng C, Cao Y, Zhu X, et al. A computational model for inferring QTL control networks underlying developmental covariation. Front Plant Sci 2019;10:1557.

[238] Jäger G, Battke F, Nieselt K. Reveal—visual eQTL analytics. Bioinformatics 2012;28(18):i542–8.

[239] Price MN, Dehal PS, Arkin AP. FastTree 2 – Approximately maximum-likelihood trees for large alignments. PLOS ONE 2010;5(3):e9490, Publisher: Public Library of Science.

[240] Munzner T, Guimbretière F, Tasiran S, Zhang L, Zhou Y. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In: ACM SIGGRAPH 2003 papers. ACM; 2003, p. 453–62.

[241] Böttger J, Schäfer A, Lohmann G, Villringer A, Margulies DS. Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain. IEEE Trans Vis Comput Graphics 2014;20(3):471–80, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[242] Natsukawa H, Deyle ER, Pao GM, Koyamada K, Sugihara G. A visual analytics approach for ecosystem dynamics based on empirical dynamic modeling. IEEE Trans Vis Comput Graphics 2021;27(2):506–16, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[243] Pijuan-Sala B, Griffiths JA, Guibentif C, Hiscock TW, Jawaid W, Calero-Nieto FJ, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. Nature 2019;566(7745):490–5, Publisher: Nature Publishing Group.

[244] Zhou Y, Chalapathi N, Rathore A, Zhao Y, Wang B. Mapper interactive: A scalable, extendable, and interactive toolbox for the visual exploration of high-dimensional data. 2021, arXiv:2011.03209 [cs].

[245] Chakraborty R, Hasija Y. miDerma: An integrated database and tool for analysis of miRNAs associated with dermatological disorders. In: 2018 international conference on bioinformatics and systems biology. Allahabad, India: IEEE; 2018, p. 170–3.

[246] Purchase HC. Metrics for graph drawing aesthetics. J Vis Lang Comput 2002;13(5):501–16.

[247] Okoe M, Jianu R, Kobourov S. Revisited experimental comparison of node-link and matrix representations. In: Frati F, Ma K-L, editors. Graph drawing and network visualization. Lecture notes in computer science, Cham: Springer International Publishing; 2018, p. 287–302.

[248] Purchase H. Which aesthetic has the greatest effect on human understanding? In: DiBattista G, editor. Graph drawing. Berlin, Heidelberg: Springer; 1997, p. 248–61.

[249] Ehlers H, Villedieu A, Raidou RG, Wu H-Y. Improving readability of static, straight-line graph drawings: A first look at edge crossing resolution through iterative vertex splitting. Comput Graph 2023.

[250] Kaimal V, Bardes EE, Tabar SC, Jegga AG, Aronow BJ. ToppCluster: a multiple gene list feature analyzer for comparative enrichment clustering and network-based dissection of biological systems. Nucleic Acids Res 2010;38(suppl_2):W96–102.

[251] Kamal A, Dhakal P, Javaid AY, Devabhaktuni VK, Kaur D, Zaientz J, et al. Recent advances and challenges in uncertainty visualization: a survey. J Vis 2021;24(5):861–90.

[252] Hullman J. Why authors don't visualize uncertainty. IEEE Trans Vis Comput Graphics 2020;26(1):130–9, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[253] Sacha D, Senaratne H, Kwon BC, Ellis G, Keim DA. The role of uncertainty, awareness, and trust in visual analytics. IEEE Trans Vis Comput Graphics 2016;22(1):240–9, Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[254] Dubois J, Cottret L, Ghozlane A, Auber D, Bringaud F, Thebault P, et al. 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. Montpellier, France: IEEE; 2012, p. 204–13 .

[255] Alper B, Bach B, Henry Riche N, Isenberg T, Fekete J-D. Weighted graph comparison techniques for brain connectivity analysis. In: Proceedings of the SIGCHI conference on human factors in computing systems. New York, NY, USA: Association for Computing Machinery; 2013, p. 483–92, event-place: Paris, France.
[256] Vehlow C, Reinhardt T, Weiskopf D. Visualizing fuzzy overlapping communities in networks. IEEE Trans Vis Comput Graphics 2014;19(12):2486–95.
[257] Mayr E, Hynek N, Salisu S, Windhager F. Trust in information visualization. In: EuroVis workshop on trustworthy visualization. 2019, p. 5.
[258] Kelton K, Fleischmann KR, Wallace WA. Trust in digital information. J Am Soc Inf Sci Technol 2008;59(3):363–74.
[259] Spinner T, Schlegel U, Schäfer H, El-Assady M. explAIner: A visual analytics framework for interactive and explainable machine learning. IEEE Trans Vis Comput Graphics 2020;26(1):1064–74, Conference Name: IEEE Transactions on Visualization and Computer Graphics.
[260] Chatzimparmpas A, Martins RM, Jusufi I, Kucher K, Rossi F, Kerren A. The state of the art in enhancing trust in machine learning models with the use of visualizations. Comput Graph Forum 2020;39(3):713–56, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14034.
[261] Beck F, Burch M, Diehl S, Weiskopf D. The state of the art in visualizing dynamic graphs. The Eurographics Association; 2014.
[262] Beck F, Burch M, Diehl S, Weiskopf D. A taxonomy and survey of dynamic graph visualization. Comput Graph Forum 2017;36(1):133–59, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12791.
[263] Linhares CDG, Ponciano JR, Paiva JGS, Travençolo BAN, Rocha LEC. Visualisation of structure and processes on temporal networks. In: Holme P, Saramäki J, editors. Temporal network theory. Cham: Springer International Publishing; 2019, p. 83–105.
[264] Peng D, Tian W, Lu B, Zhu M. DMNEVis: A novel visual approach to explore evolution of dynamic multivariate network. In: 2018 IEEE international conference on systems, man, and cybernetics. Miyazaki, Japan: IEEE; 2018, p. 4304–11.
[265] Liu J, Jing L, Tu X. Weighted gene co-expression network analysis identifies specific modules and hub genes related to coronary artery disease. Sci Rep 2021;11(1):6711, Publisher: Springer Science and Business Media LLC.
[266] Kale B, Sun M, Papka ME. The state of the art in visualizing dynamic multivariate networks. Comput Graph Forum 2023;42(3):471–90, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14856.
[267] Calderer G, Kuijjer ML. Community detection in large-scale bipartite biological networks. Front Genet 2021;12. Publisher: Frontiers.