

Interactive and coordinated visualization approaches for biological data analysis

António Cruz, Joel P. Arrais and Penousal Machado

Corresponding author: António Cruz, CISUC, Department of Informatics Engineering, University of Coimbra, Pólo II—Pinhal de Marrocos, 3030-290 Coimbra, Portugal. Tel.: +351 239 790 000; Fax: +351 239 701 266; E-mail: antonioc@dei.uc.pt

Abstract

The field of computational biology has become largely dependent on data visualization tools to analyze the increasing quantities of data gathered through the use of new and growing technologies. Aside from the volume, which often results in large amounts of noise and complex relationships with no clear structure, the visualization of biological data sets is hindered by their heterogeneity, as data are obtained from different sources and contain a wide variety of attributes, including spatial and temporal information. This requires visualization approaches that are able to not only represent various data structures simultaneously but also provide exploratory methods that allow the identification of meaningful relationships that would not be perceptible through data analysis algorithms alone. In this article, we present a survey of visualization approaches applied to the analysis of biological data. We focus on graph-based visualizations and tools that use coordinated multiple views to represent high-dimensional multivariate data, in particular time series gene expression, protein–protein interaction networks and biological pathways. We then discuss how these methods can be used to help solve the current challenges surrounding the visualization of complex biological data sets.

Key words: coordinated multiple views; multivariate visualization; time series data; gene expression

Introduction

A vast quantity of visualization tools has emerged over the past decade as a response to the need for analyzing large, unstructured data sets, particularly in the field of Biology [1]. The data sets that are often the target of modern research in the field of computational biology are often complex because of possessing multiple characteristics found in ‘big data’. While the description of big data may vary, it is typically characterized by four main properties, which are at the center of current visualization challenges: the sheer volume of data; the variety of formats, data structures and variable types; the velocity at which the data are retrieved, analyzed and represented; and the task of determining the validity of the data [2].

In regards to volume, the amount of research data and the speed at which it is gathered have been increasing along with the technological developments that have taken place across multiple fields. The representation of large volumes of data

may lead to slow performance, particularly in interactive environments, as well as a large amount of noise. Beck *et al.* [3] presented a survey, which identifies visual scalability as one of the main challenges in developing scientific graphs, emphasizing a lack of visualization methods that work along with data reduction methods, particularly when handling time series data or dynamic structures [4]. Wang *et al.* [5] also argues that graphs derived from scientific data sets can be significantly large and complex, advocating for graph simplification and data mining to more easily find community structures and track features over time.

The heterogeneity of biological data can be particularly difficult to manage as the data sets are often multivariate with varied structures and temporal or spatial attributes [6]. Additionally, these data sets are often complimented with data integrated from external databases. The challenge in representing diverse data sets stems from choosing graphical elements

António Cruz is a PhD student currently studying and developing interactive visualization methods for biological data analysis.

Joel Perdiz Arrais is a professor interested in developing algorithms to model biological problems, in particular through pattern recognition and machine learning methods.

Penousal Machado is a professor whose research interests include computational design, artificial intelligence and evolutionary computation.

Submitted: 30 November 2017; Received (in revised form): 24 January 2018

© The Author(s) 2018. Published by Oxford University Press. All rights reserved.

For permissions, please email: journals.permissions@oup.com

that can properly convey the values, properties and relationships to the user. This may warrant the exploration of new visualization metaphors that are able to better integrate different types of biological data, such as integrating network-based metabolic pathways with gene expression data [7]. In regards to time series data and multiple experiments performed under different conditions, the development of techniques for visualizing changes in the data is an ongoing challenge, particularly between two or more networks [3].

In this article, we provide an overview of visualization approaches in the context of the challenges that most impact data visualization, which stems from the large quantities of multivariate data that often characterize biological data sets. We directed our focus toward graph-based visualization of relational data, such as protein–protein interaction (PPI) networks and biological pathways, as these structures are often the most impacted by the volume of large data sets. In regards to the heterogeneity of biological data sets, we gave preference to tools that use coordinated multiple views (CMV) to represent high-dimensional multivariate data, in particular time series gene expression. Using these criteria, we reviewed visualization models and interactive techniques found across 50 biological visualization tools that vary in focus and approach (listed in the Supplementary Table). In contrast to past surveys published on multivariate biological visualization tools [6, 8, 9], we focused on the approaches implemented by the surveyed tools, reviewing the range of options for representing biological data and categorizing interactive methods used in its analysis.

We begin by overviewing data analysis methods and basic visualization models used to explore biological data. This is followed by a review of comparative visualization layouts, where multiple visualization models are composed and shown simultaneously to represent relationships between diverse data sets, as well as methods to simplify data representation to aid the comparison between large volumes of data. We then categorize the interaction methods found in the surveyed tools, highlighting those that can be coordinated across multiple views to facilitate knowledge discovery. Finally, we present a discussion on how the surveyed models and methods can be used to tackle the current challenges surrounding the representation and analysis of complex biological data sets.

Visualization background

In this section, we will go over visualization models and layouts prominently used in the representation of biological data. Although it is not the focus, we also provide an overview of data analysis methods, as these are at the base of many visualization techniques seeking to extract knowledge from complex data sets. Visualization is a significant step in knowledge discovery, as it can be used in conjunction with data analysis to highlight and identify patterns, trends and outliers while engaging users through aesthetic graphical representations and helping them make decisions [10]. As such, understanding the available visualization models and how visual elements are perceived is necessary when choosing those that are most appropriate for representing a data set.

Data analysis

Data analysis is an essential part of the process of extracting knowledge from the complex biological data sets characterized as big data. Feature selection, dimensionality reduction and clustering algorithms are used to filter and order data, reducing

computational loading times by removing or hiding irrelevant data groups and highlighting patterns or new information that can support the user's navigation and queries.

Dimensionality reduction methods map data to a lower dimensional space to minimize redundant variance in the data, reducing the number of variables with minimal loss of information [11]. Owing to the high dimensionality in omics data sets, they have been used in exploratory analysis to better understand molecular pathways in cells and their role in diseases [12], as well as in extracting patterns from gene expression data where there is typically a large amount of noise [13–15].

Feature selection and pattern extraction are used to identify meaningful characteristics from a set of candidates while avoiding noise [16, 17]. These are used to identify underlying processes in complex biological networks, such as finding differentially expressed genes, detecting genotype patterns and discovering network motifs [18, 19]. Motifs are common subgraphs or patterns identified in networks [20] that have been used to predict interaction patterns of proteins in PPI networks [21] and in analyzing gene regulation networks [22].

Clustering is the unsupervised classification of data, with the goal of creating discernable groups composed of highly similar elements. There are multiple surveys of clustering algorithms directed at the analysis of gene expression data [23, 24] and PPI networks, as well as for time series [25] and big data [26]. Hierarchical clustering algorithms stand out by producing nested series of partitions by that can be used to recurrently split or merge clusters [27]. They are used in the analysis of gene expression matrices to group genes that exhibit similar expression patterns over time or different experimental conditions [28, 29]. Biclustering algorithms can be used to perform simultaneous clustering on the row and column dimensions of the gene expression matrix and identify similar subgroups of genes under specific subsets of conditions, but can result in overlapping clusters [30].

Visualization models

Linear representations are among the simplest visualization models, being used to map data points to observe general patterns or trends. These are often used in biological visualization tools as secondary views that portray additional information about sections of the data set, such as line charts [31, 32], bar charts [33, 34], histograms [29, 35] and scatterplots [36, 37].

Parallel coordinates have a similar representation to line charts but are able to represent a higher number of variables [34, 38], listing each variable across a separate axis with values ordered based on properties, such as connectivity, density, centrality or quantitative annotation. A notable variation of parallel coordinate plots is Hive Plots [39], where the axis is arranged in a circle and the edges are drawn as Bezier curves, resulting in a more compact visualization.

Heatmaps are prevalent among biological visualization tools [40, 41], typically used in the representation of time series gene expression [42]. While variations to heatmaps are not common, a hexagonal grid layout was presented by GATE [43]. Clustering algorithms can be used to sort heatmaps by ordering the rows of genes, so that those with similar temporal patterns of expression are placed close to each other. If a heatmap is hierarchically clustered, it can be accompanied by a tree structure that shows where each cluster either merged or split, known as a dendrogram [35, 44].

Networks, or node-link diagrams, are particularly proficient at displaying multivariate data and its relationships, being used

in biology to represent PPI, gene regulation and biological pathways [9]. Nodes can represent multiple attributes through basic visual properties like color, shape and size, as well as labels [45, 46]. However, a greater number of variables can be represented simultaneously through complex representations, known as glyphs [47]. Relationships are portrayed with edges, which can be characterized using visual attributes such as color, direction and weight. Edges with clear directions are characterized as directed, while those with a discernable width can be described as ribbons.

A prominent problem with networks is the representation of large quantities of edges, which often results in unintelligible ‘hairballs’ [39, 48]. To resolve this, the positions of nodes can be calculated through clustering algorithms and force-based layouts [49], with objectives such as edge crossing minimization and grouping similar elements [46, 50]. Additionally, edge bundling reduces visual clutter by drawing their paths closer to those with similar directions [51], creating organic bundles of edges with clear directions that are easier to follow. This is useful in circular layouts [52, 53] because of nodes having fixed positions. Alternatively to networks, pathways can be represented orderly through path lists [54], but these can be extensive, and nodes present on multiple paths will be repeated.

Hierarchical networks, also known as trees, are used in phylogenetic analysis [55, 56] and in the visualization of microarray data [57]. However, layered visualizations can also be used to order data hierarchically. Arena3D [58] presents a three-dimensional approach to a layout comprised of multiple layers, where two-dimensional networks are displayed parallel to each other, each comprising one level, interconnected through edges. Cerebral [59] has a layered layout option that splits a network into well-defined levels, each representing a different class of proteins. Other hierarchical structures include sunburst and icicle visualizations, used by Taylor *et al.* [60] to visualize gene expression experiments on the developmental mouse.

Encoding and perception

The perceptual phenomena that occur when visualizing graphical elements have been the target of studies that seek to better understand the role of encoding in problem-solving. Bertin [61] helped create the foundation for visual encoding by establishing the effectiveness of graphical proprieties in representing diverse variables, such as how a quantitative value should be mapped to an element’s position or size but not to its shape. This is complemented by principles that describe the perception of elements based on these properties. For instance, the Gestalt laws explain how elements that are close together or share graphical proprieties tend to be associated as being a part of the same group [62]. As such, these guidelines are at the base of approaches that group similar elements, such as clustering and edge bundling. Vehlou *et al.* [63] presented a comprehensive survey that categorizes the representation of groups on graphs based on the visualization model, structure and visual attributes.

While we are interested in how to portray more easily perceptible relationships and patterns, uncertainty should be minimized. This means that encoding should prioritize data fidelity while avoiding the creation of vague or unintelligible elements. In this regard, Tufte [64] established principles for ‘graphical excellency’, which favors clarity and precision while avoiding distorting the data for the purpose of aesthetics. Furthermore, Dasgupta *et al.* [65] proposed a taxonomy that comprehensively describes cases of uncertainty at both an encoding level, which includes handling missing values and graphical limitations of

the screen, and at a decoding level, such as indiscernible relationships in cluttered graphs or clusters.

Comparative visualization

A visualization environment that enables the representation of several multivariate data sets simultaneously can help identify new meaningful relationships, but the number of variables and types of relationships in the data may not fit any visualizations. In such cases, data visualization tools often use CMV, an exploratory visualization technique that enables the representation of diverse data sets simultaneously by composing multiple visualization models, which can have user interactions coordinated between them [66]. CMV can be effective in discovering patterns and unforeseen relationships, identifying and understanding outliers and gaining insight from comparing multiple data sets [67]. In this section, we review the layouts of visualization models found in the surveyed biological visualization tools that use CMV. Additionally, we identify strategies to explore relationships between large volumes of data, such as their abstraction into manageable elements that can be compared across views.

Guidelines for multiple views

The amount of information that can be displayed on screen is limited by both the available space and the user’s cognitive limits when interpreting large amounts of diverse data representations. It is important to consider how the placement and availability of the views impact the user’s navigation and cognitive ability to understand the data to extract new knowledge. In this sense, design decisions benefit from understanding how the workspace environment is perceived. Baldonado *et al.* [66] present guidelines on the use of multiple views, noting that they should be used when there are multiple types of attributes, models, user profiles, levels of abstraction or genres. Additionally, they are effective in the discovery of correlations or disparities in the data as they allow for data to be extracted and compared on the screen rather than mentally, which is less straining on the user. While an overview of the data can be helpful, not only can it be cognitively overwhelming but the user may also benefit from isolating and visualizing a particular section of the data. As such, a complex visualization should be divided into multiple views that provide more detail and easier management. However, the addition of views should be justified, as it introduces additional complexity to both the program and the user.

Composition and layout

The composition of the visualization environment determines how diverse visualization models can be compared and how their relationships will be perceived. Previous surveys have categorized the composition of graphical elements or groups of elements into: juxtaposition, superimposition, nesting and encoding [68, 69].

Juxtaposition is the most common composition mechanism, where elements are displayed side by side. To compare between views, each can be assigned a space by either dividing the work space [29, 34] or through individual windows [36, 70]. The advantage of windows is that each visualization can be contained even when it is larger than the available space and be navigated through interaction. Additionally, their size, position and visibility can be manipulated by the user. Comparison through juxtaposition can be used to identify patterns and relationships through common graphical properties, or for contextual

information. In an overview + detail layout, one view provides an overview of the data set, while other views focus on specific sections of the visualization with additional details [67, 71]. This encourages navigation, as users can drill-down on different sections without having to roll-up in between.

When the same type of visualization models is displayed multiple times in smaller sizes in a sequence or grid, they are referred to as small multiples [72]. This layout can be used to create static or dynamic overviews of data representations that have various states, such as time series [43, 71] or experiments performed with different parameters [73]. However, new views have a cognitive impact on users, and smaller changes can go unnoticed, as they shift their attention between views, particularly between small multiples.

Superimposition is an approach for highlighting structural similarities and differences by stacking the same visualization models [74, 75], helping users identify small changes more effectively than juxtaposition. Additional relationships can also be shown by superimposing new elements, such as drawing edges on top of matrices [43, 76]. However, considering that the information is overlapped, it may be difficult to identify individual elements. As such, the scalability of this approach is reliant on the existence of proper interaction techniques to navigate the data [69].

Nesting, in this context, consists of embedding a visualization model into another one's structure. Unlike in superimposition, the nested model is treated as an element of the parent visualization. Some biological tools use linear visualizations as glyphs, embedding line charts [77, 78] and bar charts [79] into networks, having been used to represent temporal data associated to each gene. Basic node graphical properties can still be altered, such as using different background colors to represent an extra variable [80]. It is also possible to embed more complex visualizations, such as glyphs, to represent relationships between multivariate data sets, such as heatmaps or other networks [34, 81].

Alternatively, differences and similarities can be computed and encoded in a new visualization where regions of interest are explicitly highlighted [69], making them easily identifiable. Encoding time series through animation is a natural way of conveying changes over time, but it is limited by human perception capabilities [82]. Transitions between successive states can be smoothly animated by interpolating values of properties like color and size [32], but details go unnoticed in short transitions, and it is difficult to compare between time points. On the other hand, a time line is able to simultaneously represent multiple time points through a variety of scales, shapes and layouts [83], but graphical representations are limited because of space. By using a time line as a navigation element, the user can switch between states in another view and focus on key moments.

When representing diverse data sets, multiple composition mechanisms can be used simultaneously to show different relationships and compare data. For instance, Pathline [84] and MulteeSum [85] represent temporal gene expression profiles through a curvemap, a visualization composed of a grid of area plots. To help analyze these data, the final column and row superimpose their respective plots for an overall comparison. Despite this, the amount of information shown can be overwhelming, and the user would benefit from visual indicators that highlight regions of interest, particularly in Pathline. However, the curvemap is also used to relate time series to other types of data. Pathline uses a pathway visualization that encodes genes and metabolites, which can be selected to get added to the curvemap. MulteeSum implemented the curvemap alongside a plot visualization to relate the time series gene expression to the spatial location of their respective cells.

Managing visual complexity

Reducing the apparent size of the information space is a key strategy in managing complexity in data visualization, particularly in large networks where visual clutter is a frequent obstacle [86]. For instance, Kiwi [50] reduces complex interaction networks by isolating significant gene sets, calculating the shortest path length between each pair and drawing only the best edges.

Managing visual complexity can also be achieved through data aggregation, where multiple data points are converted into a single one [82]. Groups can be created through clustering algorithms and then represented proprieties associated with the whole, such as an average of values from a selected variable. STEM [74], BiGGesTS [44] and Cerebral [59] apply this concept to clustering time series gene expression data. By calculating the mean of every value over every time point, clusters can be represented with line charts that represent the average variation over time. These tools then list the line charts to the user sequentially as small multiples that can be selected to either highlight the respective cluster on another view or open a profile view that superimposes every gene. VisBricks [73] stands out by representing time series gene expression clusters using multiform visualizations, allowing each cluster to be represented as either a line chart, a histogram or a colored compact view of the data. These are displayed as glyphs in a parallel coordinates visualization where each axis corresponds to a different gene expression data set, connecting shared data between clusters in different data sets with ribbons. The advantage of this layout is that clusters can be sorted along each axis, but relationships to other axis can only be directly drawn between with those one each side.

Groups can also be determined directly from the structure of relationships in the data. Dunne and Shneiderman [87] propose motif simplification, where common network motifs are identified and then replaced with a simple symbol that is representative of the layout of each respective subgraph. Maguire et al. [88] also propose aggregating motifs but instead using glyphs that can be represented using three varying levels of detail, which portray not just the structure of the motif but also attributes of individual nodes. The scalability of this approach is limited, as individual attributes can be hard to discern for large motifs, but glyphs can also represent the average attributes of the group.

In regards to abstracting temporal structures, Bach et al. [89] present time curves. This is a visualization model where time lines are folded in two-dimensional space by moving similar time points closer to each other, while a continuous line connects each time point. This approach preserves the temporal order of the data points, while their position is used to represent similarity. It can portray patterns of evolution that reflect how the data changes over time, including slow progressions, sudden changes and reversals to previous states. Similarly, Elzen et al. [90] propose that each state of a network can be converted into a point in a two-dimensional plane where its position is based on the values of the attributes and relationship structure of each state. In the case of time series data, the evolution order can be conveyed through edges and color.

Coordinated interaction

Interaction and dynamic visualization environments play a major role in analyzing complex biological data, as the user needs to be able to navigate through diverse data sets, compare data points and identify relationships [91]. In an environment with CMV, user interactions with a visualization model, such as selections and filters, can be dynamically applied to similar

data points represented across other views. Coordinating interaction helps users keep track of data and more quickly identify significant relationships and patterns, particularly between diverse visualization structures [92].

Ideally, interaction should be fluid, meaning continuous or smooth. The concept of fluid interaction is based on a set of general principles, which can be applied to support the user's immersion and involvement [93], such as: using animated transitions, providing immediate visual feedback, integrating interface components into the visualization and allowing the user to make changes intuitively. Baldonado *et al.* [66] described additional guidelines for designing coordinated interaction. These highlight the importance of the time costs of each interaction, such as the computational time necessary to process each change. Time costs also include the time the user takes to understand and switch between visualization models, which can be reduced by using consistent graphical elements and attributes between visualization models. Additionally, the user's attention can also be diverted to regions of interest through perceptual cues, such as animation, sounds and highlighting.

Navigation

Interactive visualizations, particularly large-scale graphs and maps, are traditionally navigated using panning and zooming techniques where the visualization is moved or transformed. As such, they can be used for position-based filtering, bringing specific elements into focus while moving others off the screen or de-emphasizing them. Cerebral [59] uses coordinated navigation, applying panning and zooming across every small multiple representing different states of the same biological network, so that they focus on the same region to compare between temporal instances.

Zooming is characterized as geometric when elements of the visualization are resized without any changes to their content. Zooming in not only moves elements off-screen but also increases the distance between them. Semantic zooming methods take advantage of this increase in white space by adapting the content to the current scale [94]. This can involve the addition of labels with details or contextual information [79], or even drastically changing the structure of the visualization to add new points and relationships related to the data in focus [95].

If the visualization is zoomed in enough, panning to another location may require the user to zoom out first before moving and then zooming in again, which can break the user's workflow, as the each step may require a significant amount of time to process. By using an overview + detail layout, the visualization overviews the data set can allow the user to directly choose the new location [96, 97]. Alternatively, focus + context methods increase the amount of detail on an element or a group of elements, without reducing the amount of information on screen [98]. This is commonly achieved by distorting the position of the elements, expanding the area in focus, while the surrounding elements are contracted, instead of being pushed off the screen [99]. TreeJuxtaposer [56] presents trees side by side for comparison, allowing the user to select a rectangular area and then enlarge it freely by dragging the corners, which not only dynamically adapts the size of the rest of the tree but can also be coordinated to enlarge the same area across other trees. Notably, this tool also implements a draw order where the nodes and branches in focus are drawn first when rendering changes, which maintains the user's attention on regions of interest. Tominski *et al.* [100] presented a survey on lens-based

focus + context approaches, showing how these can be used as interactive objects that can filter data.

In a top-down approach to navigation, big data visualizations can initially present the user with an overview of the data created through aggregation and sampling methods, while interactive functions allow the user to drill-down and access the original data. Hierarchical aggregation results in a network with clearly defined levels of detail [101], which can be navigated through recursive expansions or reductions by selecting parent nodes to either add or remove child nodes, as used by VisANT [77] and AVOCADO [102]. iHAT [103] presents a hierarchical table approach that combines the visualization of sequence and expression data, in which the user can aggregate rows and columns of a heatmap interactively. However, the visualization can get cluttered, as the user drills down and expands groups. To prevent this, either a separate view can be created to represent the contents of the expanded set of aggregated data [33] or multiple views can be defined to represent a set of number scale levels [104]. Alternatively, aggregation can be combined with semantic zooming to balance the amount of graphical elements on screen in relation to the scale level. For instance, as the user zooms in on a section, the data elements in focus are expanded, while the remainder goes offscreen.

In general, navigation methods should not just give users freedom to explore but also guide them. This includes the addition of constraints that establish limits to prevent users from going out of bounds, as well as visual hints that keep users aware of where information of interest is located. Schulz *et al.* [105] present a table-based approach for visualizing bipartite biological networks, where the scroll bars contain selection markers that indicate regions of interest, which would otherwise be offscreen because of the vertical length of the tables.

Data queries

Throughout their session, users may need to perform queries to find and focus on sections of the data that are specific to their current objectives. Queries are requests from the user that involve one or multiple constraints, which can either be categorized as searches, when the objective is to find and emphasize a specific element or group, or as filters, when the user seeks to de-emphasize or remove elements from the view and reduce visual clutter [98]. They are performed through inputs that can be characterized as indirect or direct.

Indirect inputs consist of actions usually performed through the user interface. Qualitative or discrete data can be listed as options using interface elements like tables, dropdowns and checkboxes that are used to switch between what data are visible [31, 38]. Search bars occupy a small amount of space and are useful for finding elements through partial or full names [45, 106] but have the disadvantage of requiring previous knowledge from the user. Numerical input boxes and sliders are more commonly used for handling queries over continuous data, where the user can pick values to establish thresholds, such as upper and lower limits [28].

Direct inputs involve interacting with the visualization, either by selecting the graphical representations of the data, known as brushing, or by manipulating elements through handles or widgets. Handles are sections of graphical elements that the user can select and drag to either move or resize that element [71], change the value of a property [104] or establish thresholds on the visualization [75, 84]. When the user is meant to have control over several properties, then widgets can be

used instead. These consist of small interface elements embedded into the visualization with multiple handles, buttons or input fields [73]. There are also other types of complex queries, such as the grid-based query technique implemented by PivotSlice [107], which subdivides the visualization into meaningful sections.

Brushing is usually performed by using the mouse as a brush. If the brush is a point then data elements can be chosen individually, such as hovering over them to display labels [108, 109] or adding them to another view [84]. Alternatively, multiple elements can be brushed simultaneously by using a line or an area, which is commonly drawn either as a rectangle or with a freehand lasso. TimeSearcher [71], a time series visualization tool, presents an area brushing method: timeboxes. These are rectangular regions drawn by the user on a two-dimensional display of time series data to perform queries. After being drawn, timeboxes have handles that allow the user to resize them or move to a new position. The result set from the queries will only consist of patterns that are within the constraints of the active timeboxes. In the Hierarchical Clustering Explorer [35], a visualization tool for exploring multidimensional data, a user can draw a line pattern to query similar profiles and set a distance measure to establish how similar other profiles need to be to get added to the result set. Additionally, the user can establish upper and lower thresholds directly on the visualization using the mouse. While both these tools are not specifically designed for exploring biological data, the described methods can be applied in the identification of time series gene expression profiles.

When brushing is used to concurrently highlight information across other views that is related to the selected elements, then it is known as linked brushing. This type of coordination is advantageous for the user, as it maintains consistency through the use of visual characteristics [35, 38]. This allows the user to identify equal or similar elements across views, find outliers and keep track of changes between groups, such as the addition or deletion of data elements [110]. enRoute [54], ConTour [111] and Pathfinder [48] present table-based approaches for pathway analysis coordinated with network visualizations, where path lists can be extracted to the table through brushing nodes, or selected from the table to be highlighted in the network. Owing to the amount of possible paths, pathway analysis through path lists is particularly reliant on queries. As such, these tools provide sorting methods that bring interesting pathways to the top and linear representations embedded into the cells for easier comparisons between attributes.

More complex brushing techniques have also been developed, such as the compound brushing system developed by Chen [112], where multiple brushing actions can be combined to include, exclude or reverse selections through logical operations (OR, AND, XOR). Another compound brushing technique is presented by Wright and Roberts [113], named click and brush. This method consists appending brushed elements to a list to then highlight intersections and correlations, while discoveries can be depicted in additional views.

Editing

Data analysis algorithms and sorting methods, such as clustering and force-directed layouts, are essential in the organization and visualization of large data sets. However, these techniques often do not provide perfect results, and it may be necessary to introduce the input of a user, as humans still sometimes

outperform machines in the interpretation of complex patterns. As such, not only should the user be given the option to fine-tune the visualization but should also be provided with analytical tools that increase the transparency of the implemented methods to better understand the problem and more quickly solve it [114]. In particular, tools that provide clustering approaches may allow the manual definition of clustering parameters [51, 77]. In particular, MLCut [28] uses sliders to identify and refine gene expression clusters, while Furby [81] lets the user to locally or globally adjust the threshold of the biclusters membership values to create well-defined clusters.

In regards to editing the visualization, some tools allow the selection of which variable is mapped to the color and size of nodes and edges [38, 45]. The visualization environment developed by Abello et al. [115] stands out by allowing the user to set multiple thresholds within an attribute and mapping each of these intervals to a color. A direct method for manipulating the position of elements is dragging, which has been used in networks to rearrange nodes [59, 79], and to move data across multiple views [37, 71]. Further control over biological networks can involve drawing new pathways and managing their visual properties [70, 116].

The ability to undo and redo actions is necessary to allow users to edit and explore options and parameters without fear of mistakes. Contour [111] provides a list of the user's previous actions, while PivotSlice [107] shows a visual user history by saving a thumbnail of the visualization to a separate panel whenever the user performs an edit, which can be used to go back to any of the recorded points. This concept can be explored further by providing methods to record the actions performed in the current and previous sessions and share this history with other users to confirm and extend discoveries.

Discussion

Visualization tools seek to provide users with the means to analyze biological data sets to find meaningful relationships that could lead to advances in research. However, these relationships are often complex, and their discovery is hindered by the characteristics of these data sets, in particular their heterogeneity and volume, which are at the base of prominent challenges in data visualization. In Figure 1, we showcase the variety in visualization approaches that have been used in the analysis of biological data in the context of these challenges.

Visualization tools must contend with the representation of large volumes of data, but representing hundreds of thousands of data points simultaneously is computationally demanding and can obscure relationships in the data, particularly in network representations. For instance, when a user enlarges a section of a tree in TreeJuxtaposer [56], the remainder of the visualization is dynamically shrunk and kept on screen, which is also applied to other views. While this is a notable use of focus+context, it is difficult to discern information because of the quantity and size of the shrunk elements. As such, overviews of the data should avoid overwhelming users with information and instead point them toward regions of interest, while facilitating comparisons. This is a problem of scalability, where the number of visual elements should be reduced while taking into consideration all available data, which may be achieved through statistical analysis and clustering.

Dunne and Shneiderman [87] and Maguire et al. [88] presented approaches to network reduction by representing

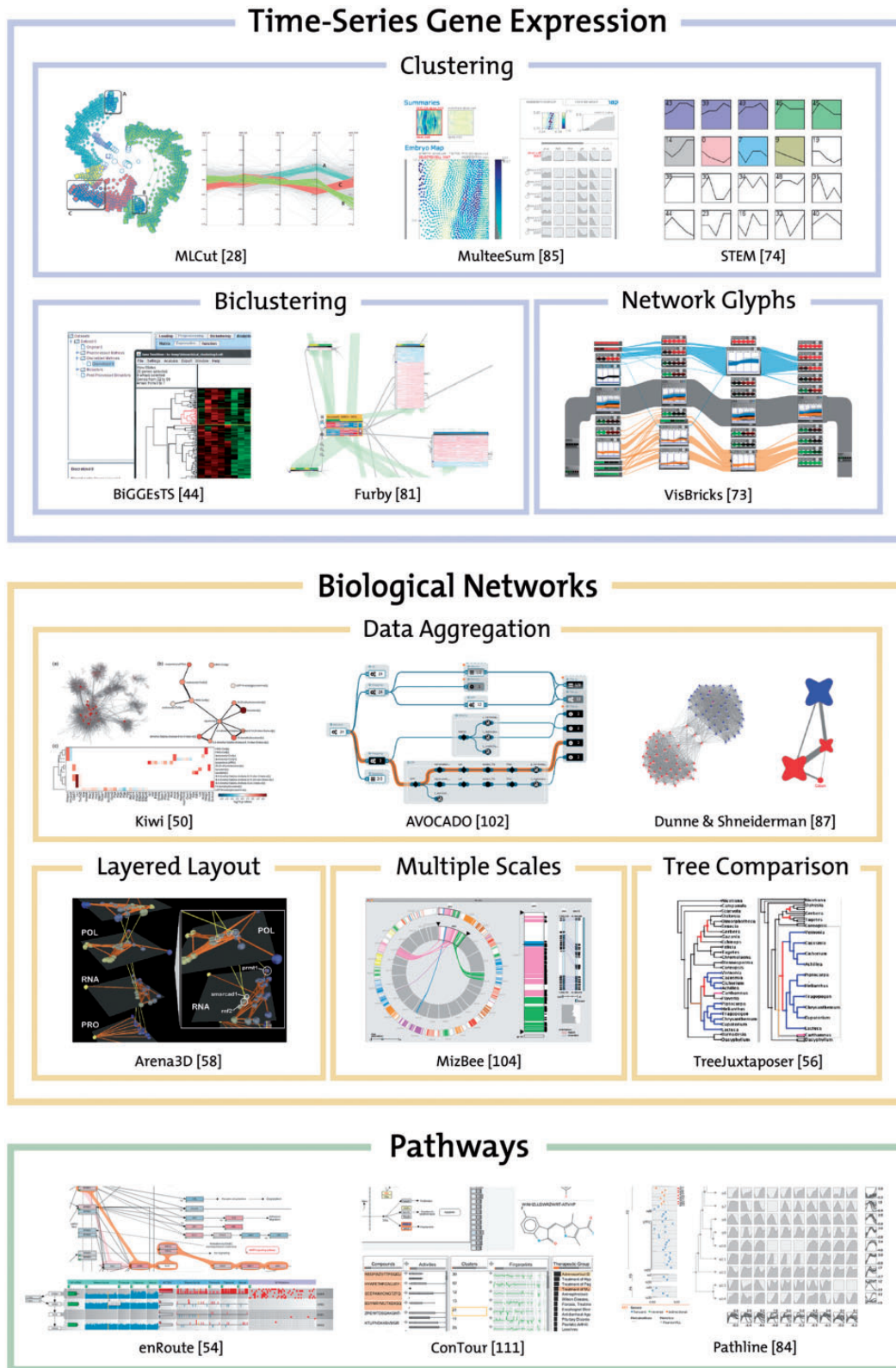


Figure 1. Thumbnails showcasing the variety of visualization tools and approaches from those surveyed, organized by the type of represented data, layouts and analysis methods.

common motifs through symbols and glyphs with graphical attributes that represent the motif's properties. However, while these methods aggregate partitions of a larger network, Bach *et al.* [89] and Elzen *et al.* [90] proposed approaches that are able to translate states of the data into points on a 2D plane, which can portray behaviors such as outliers and cycles. In the surveyed tools, network aggregation was applied through hierarchy. For instance, VisANT [77] uses interactive aggregated nodes to open or close lower levels of the network. However, the lack of descriptive visual characteristics on the nodes hinders the user's navigation, unlike in AVOCADO [102], where nodes are labeled with symbols and quantities. Additionally, semantic approaches are underexplored, as they could be used in this context to dynamically aggregate elements not in focus and keep them for context when the user drills down on demand.

Many biological visualization tools provide a range of models and layouts to represent diverse data structures, including annotations from external databases, but the depiction of relationships between them is still an issue. The most common solution is CMV, as multiple models can organize and represent individual data sets, while coordinated interaction is used to highlight common data points. For instance, ConTour [111] uses coordinated tables, networks and compound visualizations in the analysis of pathways. Its table-based approach is capable of organizing diverse attributes through columns that can be nested, and by using various graphical representation forms within each cell. Alternatively, CMV can be used to navigate through different levels of a data set. Mizbee [104] simultaneously displays four different interactive scales for comparing two sets of chromosomes.

However, encoding data is a prevalent challenge, as different variables should be graphically consistent and easily identifiable. In particular, time series gene expression has been a target of multiple visualization approaches that seek to represent relationships between genes that have similar expression patterns. While heatmaps have been a standard approach in representing time series, line charts have risen in popularity in biological visualization tools developed during the past decade. Not only are the values of expression profiles more easily interpreted through a line chart than a row of colors, but these can also be compared by overlaying multiple charts. Superimposition is useful for detecting trends, but it is also a significant source of uncertainty, as individual elements may be difficult to distinguish, so it should be used purposefully. For instance, Matse [75] uses superimposition to overview time series profiles and allow the user to directly apply thresholds based on the resulting visualization, while MLCut [28] encodes superimposed profiles with color to distinguish between the clusters created by the user's choice of parameters. Alternatively, by calculating the average between time series, the profile of a group can be represented with a single linear visualization. Cerebral [59] makes use of this to list clustered time series profiles through small multiples, which can be selected to highlight each group of genes in a network model.

While the temporal attributes from gene expression can be clustered and compared through compact graphical representations, we must also work toward new visualization metaphors to analyze relationships between other attributes and data sets. In this regard, Pathline [84] and MulteeSum [85] relate a curvemap, a grid of time series profiles that shows trends using superimposition, with both pathway and spatial data through interactive coordination. However, these tools highlight the need for encoded visualizations that directs

users toward patterns that may be of interest, in particular between views. As such, this warrants the exploration of new models that integrate multiple biological data sets, designed with the purpose of portraying their relationships. For instance, while VisBricks [73] uses heatmaps to represent multiple gene expression data sets, time series can be clustered and represented through small multiform visualizations, where relationships are drawn between the clusters across data sets. These are integrated in a single visualization with a parallel coordinates layout, but this does present a limitation, as each data set can only be compared against those on each side.

In summary, tools for visualizing multidimensional data are becoming more comprehensive and flexible but still present limitations in their visualization approaches. Ideally, the development of new visualization tools should focus on user-centered interaction and coordinated environments with new visualization metaphors capable of showing patterns, key changes and outliers by enabling the comparison between large multivariate data sets. Data can be aggregated into simple graphical representations that provide an informative overview of the data, while interactivity provides users with the ability to navigate through different levels of detail by drilling down and access details on demand through brushing. Future-developed visualization environments should automatically support users in their queries by predicting regions of interest and dynamically adapting the visualization to the type and amount of information on screen. While most of the surveyed tools still use static representations, force-based layouts can be used to react to dynamically changes in the environment and user inputs in real time using fluid transitions. At the same time, users should have control to make manual adjustments, both in customizing the visualization and fine-tuning parameters, as human input may help discover key relationships between elements and groups, which would not be easily discernable solely through data analysis.

Key Points

- In this article, we provided an overview of current visualization challenges, graphical representations and interactive methods applied to biological data, complemented with a discussion on their use in the development of new tools.
- Thorough exploration of large biological data sets is dependent on the development of new visualization metaphors supported by an interactive environment.
- Coordinated interaction between multiple visualization models promotes knowledge discovery in heterogeneous data sets.

Supplementary Data

Supplementary data are available online at <https://academic.oup.com/bib>.

Funding

This work was supported by the Fundação para a Ciência e Tecnologia (FCT) (grant number SFRH/BD/124538/2016 to A.C.).

References

- Kerren A, Kucher K, Li YF, et al. BioVis Explorer: a visual guide for biological data visualization techniques. *PLoS One* 2017;12(11):e0187341.
- Greene AC, Giffin KA, Greene CS, et al. Adapting bioinformatics curricula for big data. *Brief Bioinform* 2016;17(1):43–50.
- Beck F, Burch M, Diehl S, et al. A taxonomy and survey of dynamic graph visualization. *Comput Graph Forum* 2017;36(1):133–59.
- Andrienko G, Andrienko N. Coordinated multiple views: a critical view. In: *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007. CMV'07. IEEE, Zurich, Switzerland, 2007, 72–4.
- Wang C, Tao J. Graphs in scientific visualization: a survey. *Comput Graph Forum* 2017;36(1):263–87.
- Secrier M, Schneider R. Visualizing time-related data in biology, a review. *Brief Bioinform* 2014;15(5):771–82.
- Rezola A, Pey J, Tobalina L, et al. Advances in network-based metabolic pathway analysis and gene expression data integration. *Brief Bioinform* 2015;16(2):265–79.
- Dunn W, Jr, Burgun A, Krebs MO, et al. Exploring and visualizing multidimensional data in translational research platforms. *Brief Bioinform* 2017;18:1044–56.
- Pavlopoulos GA, Malliarakis D, Papanikolaou N, et al. Visualizing genome and systems biology: technologies, tools, implementation techniques and trends, past, present and future. *Gigascience* 2015;4(1):38.
- Heer J, Bostock M, Ogievetsky V. A tour through the visualization zoo. *Queue* 2010;8(5):20.
- Guyon I, Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res* 2003;3:1157–82.
- Meng C, Zeleznik OA, Thallinger GG, et al. Dimension reduction techniques for the integrative analysis of multi-omics data. *Brief Bioinform* 2016;17(4):628–41.
- Venna J, Kaski S. Comparison of visualization methods for an atlas of gene expression data sets. *Inf Vis* 2007;6(2):139–54.
- Jolliffe IT. Principal component analysis and factor analysis. In: *Principal Component Analysis*. Springer, New York, 2002, 150–66.
- Bijnens L, Lewi P, Goehlmann H, et al. Spectral map analysis—a method to analyze gene expression data. In: 2003 Proceedings of the American Statistical Association, Biopharmaceutical Section [CDROM]. American Statistical Association, Alexandria, VA, 2004, 553–9.
- Jain AK, Duin RP, Mao J. Statistical pattern recognition: a review. *IEEE Trans Pattern Anal Mach Intell* 2000;22(1):4–37.
- Xu R, Wunsch D. Survey of clustering algorithms. *IEEE Trans Neural Netw* 2005;16(3):645–78.
- Lazar C, Taminau J, Meganck S, et al. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans Comput Biol Bioinform* 2012;9(4):1106–19.
- Liu X, Wu J, Gu F, et al. Discriminative pattern mining and its applications in bioinformatics. *Brief Bioinform* 2015;16(5):884–900.
- Milo R, Shen-Orr S, Itzkovitz S, et al. Network motifs: simple building blocks of complex networks. *Science* 2002;298(5594):824–7.
- Albert I, Albert R. Conserved network motifs allow protein-protein interaction prediction. *Bioinformatics* 2004;20(18):3346–52.
- Tran NT, Mohan S, Xu Z, et al. Current innovations and future challenges of network motif detection. *Brief Bioinform* 2015;16(3):497–525.
- Jiang D, Tang C, Zhang A. Cluster analysis for gene expression data: a survey. *IEEE Trans Knowl Data Eng* 2004;16:1370–86.
- Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. *J Comput Biol* 1999;6(3–4):281–97.
- Rani S, Sikka G. Recent techniques of clustering of time series data: a survey. *Int J Comput Appl* 2012;52(15):1.
- Fahad A, Alshatri N, Tari Z, et al. A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Trans Emerg Top Comput* 2014;2(3):267–79.
- Jain AK, Murty MN, Flynn PJ. Data clustering: a review. *ACM Comput Surv* 1999;31(3):264–323.
- Vogogias A, Kennedy J, Archaumbault D, et al. Mlcute: Exploring multi-level cuts in dendrograms for biological data. In: *Proceedings of the Conference on Computer Graphics & Visual Computing, CGVC '16*. Eurographics Association, Goslar, Germany, 2016, 1–8.
- Qlucore. Qlucore Omics Explorer 3.3, 2017. [https://www.qlucore.com/sites/default/files/2017-09/Qlucore Omics Explorer 3.3 feature overview A_0.pdf](https://www.qlucore.com/sites/default/files/2017-09/Qlucore%20Omics%20Explorer%203.3%20feature%20overview%20A_0.pdf) (21 November 2017, date last accessed).
- Madeira SC, Oliveira AL. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans Comput Biol Bioinform* 2004;1(1):24–45.
- Dalziel B, Yang H, Singh R, et al. Xmas: an experiential approach for visualization, analysis, and exploration of time series microarray data. In: *Bioinformatics Research and Development*. Springer, Berlin, Heidelberg, 2008, 16–31.
- Theocharidis A, van Dongen S, Enright AJ, et al. Network visualization and analysis of gene expression data using BioLayout Express(3D). *Nat Protoc* 2009;4(10):1535–50.
- Ding H, Wang C, Huang K, et al. iGPSe: a visual analytic system for integrative genomic based cancer patient stratification. *BMC Bioinformatics* 2014;15(1):203.
- Lex A, Streit M, Schulz H-J, et al. StratomeX: visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Comput Graph Forum* 2012;31(3 Pt 3):1175–84.
- Seo J, Shneiderman B. A rank-by-feature framework for interactive exploration of multidimensional data. *Inf Vis* 2005;4(2):96–113.
- Hibbs MA, Dirksen NC, Li K, et al. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics* 2005;6:115.
- Angelelli P, Oeltze S, Haász J, et al. Interactive visual analysis of heterogeneous cohort-study data. *IEEE Comput Graph Appl* 2014;34(5):70–82.
- Santamaría R, Therón R, Quintales L. BicOverlapper 2.0: visual analysis for gene expression. *Bioinformatics* 2014;30(12):1785–6.
- Krzywinski M, Birol I, Jones SJ, et al. Hive plots—rational approach to visualizing networks. *Brief Bioinform* 2012;13(5):627–44.
- Bhuvaneshwar K, Belouali A, Singh V, et al. G-DOC plus—an integrative bioinformatics platform for precision medicine. *BMC Bioinformatics* 2016;17(1):193.
- Niederer C, Stitz H, Hourieh R, et al. TACO: visualizing changes in tables over time. *IEEE Trans Vis Comput Graph* 2018;24:677–86.
- Eisen MB, Spellman PT, Brown PO, et al. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* 1998;95(25):14863–8.
- MacArthur BD, Lachmann A, Lemischka IR, et al. GATE: software for the analysis and visualization of high-dimensional time series expression data. *Bioinformatics* 2010;26(1):143–4.

44. Gonçalves JP, Madeira SC, Oliveira AL. BiGGesTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Res Notes* 2009;2(1):124.
45. Köhler J, Baumbach J, Taubert J, et al. Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics* 2006;22(11):1383–90.
46. Emig D, Salomonis N, Baumbach J, et al. AltAnalyze and DomainGraph: analyzing and visualizing exon expression data. *Nucleic Acids Res* 2010;38:W755–62.
47. Chernoff H. The use of faces to represent points in k-dimensional space graphically. *J Am Stat Assoc* 1973;68(342):361–8.
48. Partl C, Gratzl S, Streit M, et al. Pathfinder: visual analysis of paths in graphs. *Comput Graph Forum* 2016;35(3):71–80.
49. Fruchterman TM, Reingold EM. Graph drawing by force-directed placement. *Softw Pract Exp* 1991;21(11):1129–64.
50. Våremo 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.
51. Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003;13(11):2498–504.
52. Krzywinski M, Schein J, Birol Í, et al. Circos: an information aesthetic for comparative genomics. *Genome Res* 2009;19(9):1639–45.
53. Curtis RE, Yuen A, Song L, et al. TVNViewer: an interactive visualization tool for exploring networks that change over time or space. *Bioinformatics* 2011;27(13):1880–1.
54. Partl C, Lex A, Streit M, et al. enRoute: dynamic path extraction from biological pathway maps for exploring heterogeneous experimental datasets. *BMC Bioinformatics* 2013;14(Suppl 19):S3.
55. Pavlopoulos GA, Soldatos TG, Barbosa-Silva A, et al. A reference guide for tree analysis and visualization. *Biodata Mining* 2010;3(1):1.
56. Munzner T, Guimbretière F, Tasiran S, et al. Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans Graph* 2003;22(3):453–62.
57. Saldanha AJ. Java Treeview—extensible visualization of microarray data. *Bioinformatics* 2004;20(17):3246–8.
58. Secrier M, Pavlopoulos GA, Aerts J, et al. Arena3D: visualizing time-driven phenotypic differences in biological systems. *BMC Bioinformatics* 2012;13:45.
59. Barsky A, Munzner T, Gardy J, et al. Cerebral: visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans Vis Comput Graph* 2008;14(6):1253–60.
60. Taylor A, McLeod K, Armit C, et al. Visualization of gene expression information within the context of the mouse anatomy. *arXiv preprint arXiv: 1407.2117*, 2014. <https://arxiv.org/abs/1407.2117>.
61. Bertin J. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, Wisconsin, 1983.
62. Graham L. Gestalt theory in interactive media design. *J Human Soc Sci* 2008;2(1).
63. Vehlou C, Beck F, Weiskopf D. The state of the art in visualizing group structures in graphs. In: *Eurographics Conference on Visualization, 2015. (EuroVis)-STARS, v.2*. The Eurographics Association, Cagliari, Italy, 2015.
64. Tufte ER. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press, 2001.
65. Dasgupta A, Chen M, Kosara R. Conceptualizing visual uncertainty in parallel coordinates. *Comput Graph Forum* 2012;31(3 Pt 2):1015–24.
66. Wang Baldonado MQ, Woodruff A, Kuchinsky A. Guidelines for using multiple views in information visualization. In: *Proceedings of the Working Conference on Advanced Visual Interfaces. AVI '00*. ACM, New York, NY, 2000, 110–19.
67. Roberts JC. State of the art: Coordinated & multiple views in exploratory visualization. In: *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07*. IEEE, 2007, 61–71.
68. Hadlak S, Schumann H, Schulz HJ. A survey of multi-faceted graph visualization. In: *Eurographics Conference on Visualization, 2015. EuroVis*. The Eurographics Association, Cagliari, Sardinia, Italy, 2015, 1–20.
69. Gleicher M, Albers D, Walker R, et al. Visual comparison for information visualization. *Inf Vis* 2011;10(4):289–309.
70. Funahashi A, Matsuoka Y, Jouraku A, et al. CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proc IEEE* 2008;96(8):1254–65.
71. Hochheiser H, Baehrecke EH, Mount SM, et al. Dynamic querying for pattern identification in microarray and genomic data. In: *2003 International Conference on Multimedia and Expo. ICME'03*. IEEE, 2003, 453–6.
72. Scherr M. Multiple and coordinated views in information visualization. *Trends Inf Vis* 2008;38:749–67.
73. Lex A, Schulz HJ, Streit M, et al. VisBricks: multiform visualization of large, inhomogeneous data. *IEEE Trans Vis Comput Graph* 2011;17(12):2291–300.
74. Ernst J, Bar-Joseph Z. STEM: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics* 2006;7:191.
75. Craig P, Cannon A, Kukla R, et al. MaTSE: the gene expression time-series explorer. *BMC Bioinformatics* 2013;14 (Suppl 19): S1.
76. Shen Y, Maz KL. Path visualization for adjacency matrices. In: *Proceedings of the 9th Joint Eurographics/IEEE VGTC Conference on Visualization, 2007. EUROVIS'07*. Aire-la-Ville. Eurographics Association, Switzerland, Switzerland, 83–90.
77. Hu Z, Mellor J, Wu J, et al. VisANT: data-integrating visual framework for biological networks and modules. *Nucleic Acids Res* 2005;33:W352–7.
78. Gerasch A, Faber D, Küntzer J, et al. BiNA: a visual analytics tool for biological network data. Porollo A, ed. *PLoS One* 2014; 9(2):e87397.
79. Kono N, Arakawa K, Ogawa R, et al. Pathway projector: web-based zoomable pathway browser using KEGG atlas and google maps API. Aziz RK, ed. *PLoS One* 2009;4(11):e7710.
80. Rohn H, Junker A, Hartmann A, et al. VANTED v2: a framework for systems biology applications. *BMC Syst Biol* 2012;6: 139.
81. Streit M, Gratzl S, Gillhofer M, et al. Furby: fuzzy force-directed bicluster visualization. *BMC Bioinformatics* 2014; 15(Suppl 6):S4.
82. Von Landesberger T, Kuijper A, Schreck T, et al. Visual analysis of large graphs: state-of-the-art and future research challenges. *Comput Graph Forum* 2011;30(6):1719–49.
83. Brehmer M, Lee B, Bach B, et al. Timelines revisited: a design space and considerations for expressive storytelling. *IEEE Trans Vis Comput Graph* 2017;23(9):2151–64.
84. Meyer M, Wong B, Styczynski M, et al. Pathline: a tool for comparative functional genomics. *Comput Graph Forum* 2010; 29(3):1043–52.
85. Meyer M, Munzner T, DePace A, et al. MulteeSum: a tool for comparative spatial and temporal gene expression data. *IEEE Trans Vis Comput Graph* 2010;16(6):908–17.
86. Noel S, Jajodia S. Managing attack graph complexity through visual hierarchical aggregation. In: *Proceedings of the 2004*

- ACM Workshop on Visualization and Data Mining for Computer Security, 2004. VizSEC/DMSEC '04. ACM, New York, NY, 109–18.
87. Dunne C, Shneiderman B. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013. CHI '13. ACM, New York, NY, 3247–56.
 88. Maguire E, Rocca-Serra P, Sansone SA, et al. Visual compression of workflow visualizations with automated detection of macro motifs. *IEEE Trans Vis Comput Graph* 2013;19(12):2576–85.
 89. Bach B, Shi C, Heulot N, et al. Time curves: folding time to visualize patterns of temporal evolution in data. *IEEE Trans Vis Comput Graph* 2016;22(1):559–68.
 90. van den Elzen S, Holten D, Blaas J, et al. Reducing snapshots to points: a visual analytics approach to dynamic network exploration. *IEEE Trans Vis Comput Graph* 2016;22(1):1–10.
 91. Przytycka TM, Singh M, Slonim DK. Toward the dynamic interactome: it's about time. *Brief Bioinform* 2010;11(1):15–29.
 92. Shimabukuro MH, Flores EF, de Oliveira MC. Coordinated views to assist exploration of spatiotemporal data: a case study. In: *Second International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2004. IEEE, London, England, 2004, 107–17.
 93. Elmqvist N, Moere AV, Jetter HC, et al. Fluid interaction for information visualization. *Inf Vis* 2011;10(4):327–40.
 94. Perlin K, Fox D. Pad: An alternative approach to the computer interface. In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 1993. SIGGRAPH '93. ACM, New York, NY, 1993, 57–64.
 95. Behrisch M, Davey J, Fischer F, et al. Visual analysis of sets of heterogeneous matrices using projection-based distance functions and semantic zoom. *Comput Graph Forum* 2014;33(3):411–20.
 96. Gómez J, García LJ, Salazar GA, et al. BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics* 2013;29(8):1103–4.
 97. Westenberg MA, Hijum SA, Lulko AT, et al. Interactive visualization of gene regulatory networks with associated gene expression time series data. *Vis Med Life Sci* 2008;293–311.
 98. Herman I, Melançon G, Marshall MS. Graph visualization and navigation in information visualization: a survey. *IEEE Trans Vis Comput Graph* 2000;6(1):24–43.
 99. Tominski C, Abello J, Van Ham F, et al. Fisheye tree views and lenses for graph visualization. In: *Tenth International Conference on Information Visualization*, 2006. IV 2006. IEEE, London, UK, 2006, 17–24.
 100. Tominski C, Gladisch S, Kister U, et al. Interactive lenses for visualization: an extended survey. *Comput Graph Forum* 2017;36(6):173–200.
 101. Elmqvist N, Fekete JD. Hierarchical aggregation for information visualization: overview, techniques, and design guidelines. *IEEE Trans Vis Comput Graph* 2010;16(3):439–54.
 102. Stitz H, Luger S, Streit M, et al. AVOCADO: visualization of workflow-derived data provenance for reproducible biomedical research. *Comput Graph Forum* 2016;35(3):481–90.
 103. Heinrich J, Vehlow C, Battke F, et al. iHAT: interactive hierarchical aggregation table for genetic association data. *BMC Bioinformatics* 2012;13(Suppl 8):S2.
 104. Meyer M, Munzner T, Pfister H. MizBee: a multiscale synteny browser. *IEEE Trans Vis Comput Graph* 2009;15(6):897–904.
 105. Schulz HJ, John M, Unger A, et al. Visual analysis of bipartite biological networks. In: *Eurographics Workshop on Visual Computing for Biomedicine*, 2008.
 106. Kutmon M, Riutta A, Nunes N, et al. WikiPathways: capturing the full diversity of pathway knowledge. *Nucleic Acids Res* 2016;44:D488–94.
 107. Zhao J, Collins C, Chevalier F, et al. Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Trans Vis Comput Graph* 2013;19(12):2080–9.
 108. Joshi-Tope G, Gillespie M, Vastrik I, et al. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res* 2005;33:D428–32.
 109. Thimm O, Bläsing O, Gibon Y, et al. MAPMAN: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant J* 2004;37(6):914–39.
 110. Lawrence M, Lee EK, Cook D, et al. Explorase: exploratory data analysis of systems biology data. In: *International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2006. 2006, 14–20.
 111. Partl C, Lex A, Streit M, et al. ConTour: data-driven exploration of multi-relational datasets for drug discovery. *IEEE Trans Vis Comput Graph* 2014;20(12):1883–92.
 112. Chen H. Compound brushing explained. *Inf Vis* 2004;3(2):96–108.
 113. Wright MA, Roberts JC. Click and Brush: A Novel Way of Finding Correlations and Relationships in Visualizations. In: *Theory and Practice of Computer Graphics*, TPCG 2005. Eurographics Association, Canterbury, UK, 2005, 179–86.
 114. Holzinger A, Plass M, Holzinger K, et al. A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop. *arXiv preprint arXiv: 1708.01104*, 2017. <https://arxiv.org/abs/1708.01104>.
 115. Abello J, Hadlak S, Schumann H, et al. A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Trans Vis Comput Graph* 2014;20(3):337–50.
 116. Kutmon M, van Iersel MP, Bohler A, et al. PathVisio 3: an extendable pathway analysis toolbox. *PLoS Comput Biol* 2015;11(2):e1004085.