IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 June 2022, accepted 19 July 2022, date of publication 26 July 2022, date of current version 1 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194072

RESEARCH ARTICLE

Multivariate Data Exploration Through Coordinated Views

ANTÓNIO CRUZ, JOEL P. ARRAIS, AND PENOUSAL MACHADO
Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
LASI—Intelligent Systems Associate Laboratory, 3030-290 Coimbra, Portugal

Corresponding author: António Cruz (antonioc@dei.uc.pt)

This work was supported by the FCT—Foundation for Science and Technology, I.P./Ministro da Ciência, Tecnologia e Ensino Superior (MCTES) through National Funds Programa de Investimento e Despesas de Desenvolvimento da Administração Central (PIDDAC), within the scope of the Centre for Informatics and Systems of the University of Coimbra (CISUC) Research and Development Unit—UIDB/00326/2020 or Project Code UIDP/00326/2020. The work of António Cruz was supported by the FCT—Foundation for Science and Technology under Grant SFRH/BD/124538/2016 and Grant COVID/BD/152131/2021.

ABSTRACT Many fields of study still face the challenges inherent to the analysis of complex multidimensional datasets, such as the field of computational biology, whose research of infectious diseases must contend with large protein-protein interaction networks with thousands of genes that vary in expression values over time. In this paper, we explore the visualization of multivariate data through CroP, a data visualization tool with a coordinated multiple views framework where users can adapt the workspace to different problems through flexible panels. In particular, we focus on the visualization of relational and temporal data, the latter being represented through layouts that distort timelines to represent the fluctuations of values across complex datasets, creating visualizations that highlight significant events and patterns. Moreover, CroP provides various layouts and functionalities to not only highlight relationships between different variables, but also dig-down into discovered patterns in order to better understand their sources and their effects. These methods are demonstrated through multiple experiments with diverse multivariate datasets, with a focus on gene expression time-series datasets. In addition to a discussion of our results, we also validate CroP through model and interface tests performed with participants from both the fields of information visualization and computational biology.

INDEX TERMS Computational biology, data visualization, human-computer interaction, interactive systems, time-series analysis.

I. INTRODUCTION

The graphical representation of information has long been used by many fields of study to record their research, usually through the use of abstract representations or metaphors that help portray the relationships between data and the real world. As such, Information Visualization has been and continues to be applied within these domains not only to document and organize large quantities of data, but also to provide the means to explore, analyze and extract new information from it. In particular, we want to focus on the challenges that are inherent to the study of complex datasets that contain networks of relationships and temporal variables, requiring the analysis of multiple processes changing over time simultaneously [1], [2]. Such challenges are prominent in the field of computational biology, where data from measuring various biological systems is being gathered increasingly faster [3], [4] due to biological related technologies and data mining techniques [5]. These datasets include protein-protein interaction networks (PPI), metabolic pathways and regulatory networks, commonly represented through graph visualizations. Additionally, these datasets may also describe processes that change over time, such as the variation of gene expression values in infected cells. Proper analysis of such data may lead to new knowledge regarding basic molecular mechanisms in cells and the behaviors of infections, as well as a better understanding of the underlying biology and therefore to the development of new treatments [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

Researchers are faced with the task of exploring increasingly larger and more complex datasets in order to discover any meaningful relationships that could lead to the extraction of new information, and so they turn to visualization tools that allow them to model and study the relationships of various processes [7]. However, the ability to present complete sets of data to the user in a comprehensive manner still presents a significant bottleneck. Despite the advancements of the past decades, the representation and analysis of relational and temporal data continue to be pertinent topics of research within the field of data visualization, with relevancy towards various problems from different domains that have no singular solution. This is our main motivation to explore and develop novel paradigms and visualization techniques. More specifically, we have focused our research on the development of dynamic visualization models for networks and time-series data, as well as approaches that can facilitate their exploration, analysis and pattern discovery.

In this paper, we present CroP, a visualization tool that utilizes a multiple coordinated views layout, designed as a platform that can receive external datasets and represent them through comprehensible visualizations, while providing methods that facilitate their navigation and analysis. Working in an interactive environment enables the amount of information on screen to be controlled by the users, allowing them to switch to the most appropriate visualization model, navigate between different levels of detail, and filter less relevant data points to highlight those that are more significant. Moreover, CroP integrates dynamic layouts that promote self-organization between points in network models, revealing relationships between variables that would otherwise remain hidden. In particular, we are interested in applying such methods to time-series visualization, not only to represent how the data behaves over time, but also to highlight moments or periods that denote significant events. For instance, the Time Curves layout [8] is a relevant layout that warrants further exploration as it distorts timelines to represent temporal behaviors, achieved by using multi-dimensional scaling to position time points relatively to their similarity. As noted previously, challenges related to the representation and analysis of relational and temporal data continue to be pertinent within the field of Biology, mainly due to the characteristics of such datasets. In this regard, we are also particularly interested in biological datasets that fit our target research objectives, such as PPI networks and gene expression time-series data, whose analysis still represents a challenge in molecular biology [9], [10]. The analysis and identification of different types of temporal behaviors and patterns may allow for a deeper understanding of such datasets and foster new knowledge.

Our contributions in this paper focus on the representation and analysis of multivariate datasets, in particular time-series data, through interactive visualizations and coordinated multiple views. While CroP has been featured in previous publications [11]–[13], this paper presents the cumulative result of our research and development from which we can highlight the following contributions:

A. MODULAR WORKSPACE

CroP utilizes a modular workspace where each visualization model and respective interface functionalities are contained within a flexible panel that can be resized and moved, allowing users to adapt the workspace to the current dataset. Moreover, the workspace is divided by a grid and automatically adjusts the size of panels and prevents overlaps as to help maintain its organization. This modular framework accommodates the integration of new visualization models and functionalities, and allow for multiple datasets to be simultaneously visualized and compared through juxtaposition and difference views.

B. MULTIVARIATE VISUALIZATION

The implemented visualization models employ various functionalities to explore and organize data, including different network layouts, clustering, an integrated biological database, and a timeline for navigating through time-series data. Multiple variables can also be visualized through visualizations that either employ dimensionality reduction techniques or force-directed layouts, spatially positioning attributes according to their similarity in order to highlight patterns in their relationships. For instance, the time curve visualization utilizes a dynamic layout that bends timelines to reveal how the data behaves over time. These visualization models are presented through the representation of multiple biological datasets, and validated through model tests performed by users from relevant fields of study.

C. MULTIDIMENSIONAL ANALYSIS

In parallel to the CroP's visualization models, we also present methods that further explore large time-series datasets functionally and aesthetically. Firstly, we complemented the implementation of a dynamic Time Curve model with temporal glyphs, a supporting timeline graph and a lens-based approach, directed at aiding in the interactive discovery and analysis of temporal patterns across complex datasets. Secondly, we introduced Time Paths, a force-directed and parameter-based layout that can dynamically transform a time curve visualization to not only smoothen the visual elements and transitions between time points, but also reduce visual noise in favor of overall patterns. These functionalities are presented in the discovery and analysis of events and patterns in datasets with diverse characteristics, including gene expression time-series data, and employed by users from relevant fields of study in multiple interface tests.

This paper is structured as follows: we begin by presenting a summary of related work, including biological visualization tools and methods for analyzing relational and temporal datasets; we then present an overview of CroP, its visualization models and functionalities, followed by a description of these methods applied to pattern discovery and analysis; the implemented models are further showcased through the visualization of multiple diverse datasets, throughout which we employ data analysis methods to highlight patterns in the data, which are then discussed; finally, we present the validation of the visualization tool through multiple usability tests and model surveys, complemented with a discussion of the results and obtained feedback.

II. RELATED WORK

Visualization has been shown to be an important tool in knowledge discovery, being used alongside data analysis to identify and highlight patterns, trends and outliers and aid users in decision-making [14]. The need for analyzing unstructured and increasingly larger datasets has led to the continued emergence of visualization tools that seek to provide methods that facilitate the exploration and analysis of such datasets. For instance, datasets that are often the target of modern research in the field of computational biology may be classified as complex due to containing large volumes of multivariate data with a wide range of values and patterns [15], [16]. In this regard, it is necessary to not only understand the range of visualization models and tools that are available, but also the modern data analysis methods that can be used cooperatively to explore and analyze such datasets.

A. DATA ANALYSIS

In order to explore visualization methods directed at large and complex datasets, it is necessary to overview some of the supporting data analysis methods that are able to extract knowledge from the data, such as dimensionality reduction, feature selection and clustering. Dimensionality reduction is used to map data to a lower dimensional space, reducing the number of variables while minimizing the loss of information, extracting relationships between multiple variables and highlighting batch effects or outliers [17]. In biological datasets with high dimensionality, they have been used to study molecular pathways in cells and their role in diseases [18], as well as in pattern analysis in gene expression data, which is often characterized as containing a significant amount of noise [19], [20]. Such methods include singular value decomposition (SVD) and principal component analysis (PCA), which search for patterns and linear combinations across complex and noisy data, minimizing redundancy and grouping elements with similar patterns [21], [22]. Moreover, there is t-Distributed Stochastic Neighbor Embedding (t-SNE), a non-linear dimensionality reduction algorithm that uses machine learning to attribute a position on a two-dimensional plane to every data point based on their proprieties and the implicit structure of the dataset.

Clustering can also play an important role in data analysis by organizing large volumes of data into a discrete set of groups of points with similar proprieties [23]. While there is not one single clustering algorithm that can be effectively applied to every problem, multiple algorithms have been developed to answer the needs for different types of dataset and analysis [24]. For instance, hierarchical agglomerative clustering algorithms apply a bottom-up strategy that successively groups the closest clusters until only a single cluster remains, which creates a hierarchical tree that represents the nested grouping of patterns [25]. Despite its higher computational cost, it only needs to be calculated once before it can be used to create any number of clusters, having been applied to gene expression datasets to group genes that exhibit similar expression patterns over time or over diverse experimental conditions [26], [27]. Alternatively, k-means is an unsupervised machine learning algorithm which requires the number of clusters to be pre-assigned, creating random centroids and then iteratively adjusting their position to the closest data points until they stabilize which determines their respective cluster [28]. Its fast execution time makes it suitable for discovering temporal patterns in complex datasets [29], and has also resulted in the development of additional algorithms, such as the bisecting k-means algorithm which is able to recognize clusters of any shape and size [30].

We can also highlight DBSCAN (Density-Based Spatial Clustering of Applications with Noise), a clustering algorithm that finds clusters of points based on the density, grouping points that have many neighbors while marking points from areas with low-density as noise [31]. While this algorithm requires parameters specific to each dataset, it is proficient at discovering cohesive clusters of various shapes while excluding less relevant points. Furthermore, the same principles are applied by OPTICS (Ordering Points to Identify the Clustering Structure), which produces an augmented ordering of the database that represents its clustering structure based on its density, from which clusters can then be extracted [32]. However, unlike DBSCAN, sorts data points based on their neighbors, creating an hierarchical order for each cluster and also takes into consideration an additional distance measure to further filter noise, making it better suited for larger datasets.

B. VISUAL ENCODING

Despite the wide range of possible representations for varied types of data, there are works that have established guidelines for visual encoding, such as Bertin's Semiology of Graphics [33] which evaluates the effectiveness of each visual propriety in conveying different information, and the Gestalt laws [34] which explain how graphical elements are perceived based on their relative position, color, shape and orientation. These guidelines serve to not only better convey different variables, but also represent and highlight patterns of relationships between them, which includes the creation and visual categorization of groups of similar elements [35]. The development of clear and precise data visualization is further supported by the principles for 'graphical excellency' established by Tufte [36], which emphasize the importance of accuracy over aesthetics, as well as the taxonomy proposed by Dasgupta *et al.* [37] that classifies and reviews cases of uncertainty resulting from limitations of the canvas, missing values in the data, and illegible relationships caused by cluttered visuals.

One of the main challenges in the comprehensive representation of complex datasets is visual scalability, as the simultaneous representation of large volumes of multivariate data points naturally contribute towards visual noise that obfuscates potentially significant details and patterns [38], [39]. Visualizations can be simplified through aggregation, where data is grouped using data mining methods into groups of similar elements, which can be represented by a lower number of visual elements. Moreover, such elements can use compounds of visual proprieties and other elements to represent multiple proprieties simultaneously, such as glyphs. An early example of this is Chernoff faces, which represented the living conditions in Los Angeles by using faces where variables are mapped to different eyes, mouths, faces and colors [40]. However, the same concept can be applied with higher levels of abstraction, such as the symbols proposed by Dunne and Shneiderman which represent common sub-structures in graphs through different shapes and colors, in order to simplify complex networks [41].

Abstraction can also be applied to the visualization as a whole in order to highlight patterns and significant points of data through visual transformations. For instance, the Time Curves model, presented by Bach *et al.*, utilizes multidimensional scaling to position time points in low-dimensional space, bending timelines with a force-directed layout so that the relative distance between each point represents the similarity between their attributes [8]. The shape of the resulting layouts is capable of visually representing how the data behaves, such as periods of stagnancy, cycles and moments marking significant events. A similar concept was presented by Elzen *et al.* which utilizes bending timelines to represent the structural changes in a network over time, creating visualizations that show the overall behaviors of complex systems [42].

Abstraction inherently increases the complexity of visualizations, as visual elements that have been subjected to transformations may not reflect the proprieties of the data as accurately. For instance, aggregation methods may reduce the granularity of the data and hide significant individual elements. However, while abstraction can be employed to distort visual elements, interactive systems can provide the means to dig-down on sections that were highlighted by these methods, such as switching between different levels of details or even manipulate the level of abstraction. For instance, VisANT [43] and AVOCADO [44] are two visualization tools that utilize aggregation in networks to conceal child nodes, allowing these to be accessed individually by selecting the respective parent nodes, while iHAT [45] applies the same concept to heatmaps, where users can aggregate rows of gene expression data.

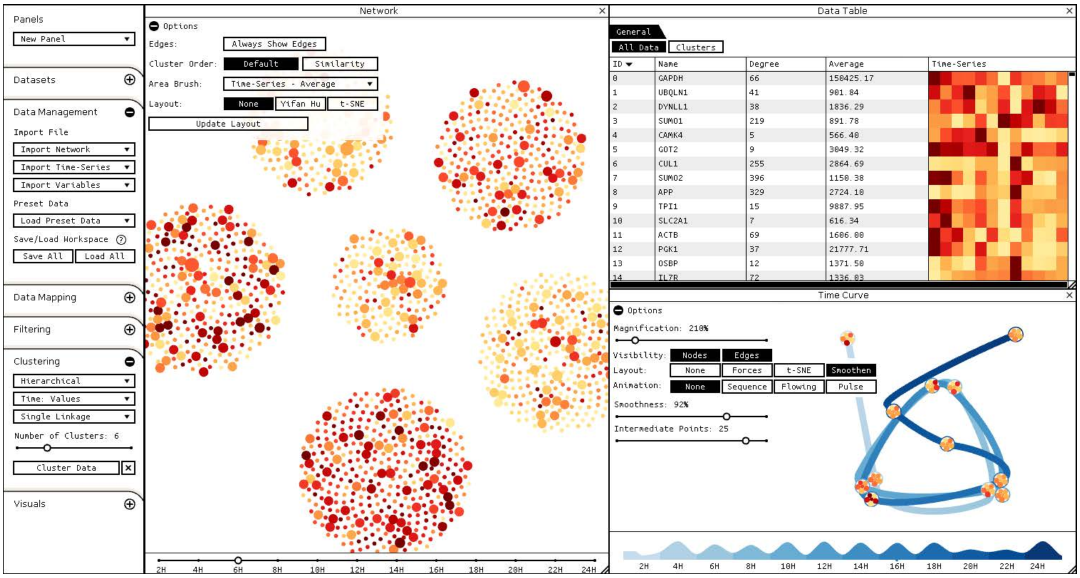
C. COORDINATED MULTIPLE VIEWS

The exploration of multivariate datasets can be supported by an environment that enables multiple visualization models, as different models focus on the representation and analysis of different data proprieties. It is in this respect that visualization tools integrate Coordinated Multiple Views (CMV) frameworks, where datasets can be represented simultaneously through varied visualization models that can be coordinated with each other to facilitate their exploration and discovery of meaningful new information [46].

The most common composition of multiple views is through juxtaposition, where each view is assigned to its own space in the work environment, either by diving the workspace [47] or by containing them within individual windows [48]. This allows users to compare visualizations that specialize in different proprieties and identify relationships and patterns through common graphical properties. Views can also be arranged in favor of navigation by providing an overview of the dataset in one view and using another to focus on specific sections [49], [50], allowing users to more easily explore visualizations with multiple levels of detail. For instance, MizBee utilizes four different scales simultaneously to enable the comparison and analysis of sets of chromosomes [51]. TimeLineCurator utilizes multiple windows to not only represent a timeline of encoded events, but also provide detailed views of these events and a control panel to manage them [52]. Moreover, juxtaposition can also be employed through small multiples, where visualizations using the same model are displayed in a sequence or grid in order to either compare between different states of a dataset, such as simultaneously visualizing the temporal profiles of multiple clusters [53], or to compare between datasets with similar structures, such as the expression data of different experiments performed over the same PPI network [54].

In addition to juxtaposition, data visualizations can also be compared through composition mechanisms that involve direct integration. For instance, views can be superimposed by overlaying multiple visualization models, which may highlight both correlations and differences between them [55]. Views can also be nested within others, where individual data elements are represented with visualization models instead of simple graphical elements, similarly to glyphs, such as replacing the nodes of a network with linear graphs [43], [56]. Additionally, similarities between views can also be computed and encoded into new visualizations that highlight significant elements [57], such as encoding changes over time by animating transitions between a sequence of states in the dataset.

Visualization tools with CMV frameworks can employ multiple composition mechanisms simultaneously, supported by interactive functionalities coordinated across different views, which facilitates navigation between distinct visualization structures [58], [59]. Saraiya *et al.* created a tool that encodes time-series in graphs through heatmaps and line charts, using multiple views and coordinated brushing to explore groups of time-series. Pathline [60] and MulteeSum [61] represent time-series gene expression profiles through small multiples of area plots displayed in a grid, known as a curvemap, where the plots from each row and column are superimposed to create representations of their average values. Additionally, Pathline encodes genes and metabolites into a pathway visualization which are added to the curvemap when brushed, while MulteeSum maps



Clustering: Hierarchical; Time: Values; Single Linkage

Number of Clusters: 6

Edges: Always Show Edges

Cluster Order: Default | Similarity

Area Brush: Time-Series - Average

Layout: None | Yifan Hu | t-SNE

ID	Name	Degree	Average	Time-Series
0	GAPDH	66	150425.17	
1	UBQLN1	41	901.84	
2	DYNLL1	38	1836.29	
3	SUMO1	219	891.78	
4	CAMK4	5	566.40	
5	GOT2	9	3049.32	
6	CUL1	255	2864.69	
7	SUMO2	396	1150.38	
8	APP	329	2724.10	
9	TPI1	15	9887.95	
10	SLC2A1	7	616.34	
11	ACTB	69	1606.00	
12	PGK1	37	21777.71	
13	OSBP	12	1371.50	
14	IL7R	72	1336.03	

Magnification: 210%

Visibility: Nodes | Edges

Layout: None | Forces | t-SNE | Smoothen

Animation: None | Sequence | Flowing | Pulse

Smoothness: 92%

Intermediate Points: 25

FIGURE 1. CroP's user interface with a loaded temporal dataset, showcasing the options sidebar (left) and three different panels in its workspace: a network panel (middle), a data table panel (top right) and a time curve panel (bottom right).

the spatial position of the cells corresponding to the curve map's expression profiles onto a plot visualization. Similarly, Cerebral is a Cytoscape plugin that uses small multiples of plots to present time-series profiles of clustered genes, highlighting the respective groups on a network view through selections [62]. Moreover, Cerebral also represents different temporal states of that network through small multiples of networks, which coordinate panning and zooming to always focus on the same region across all views.

III. COORDINATED PANELS VISUALIZATION

CroP is a data visualization tool developed in Java using the Processing library [63], designed to represent and analyze multivariate data, in particular relational and temporal data. While it is able to process generic datasets, there is additional support for biological datasets, such as the integration of an external database for cross-referencing gene proprieties, allowing it to be used to explore PPI networks and gene expression time-series.

CroP uses a CMV framework, where loaded data is represented through visualization models within flexible panels that can be arranged according to each user's objectives and queries. Data can also be sorted, clustered and filtered, which is reflected through its dynamic visualization models. Additionally, multiple datasets can also be loaded and visualized simultaneously, then compared through multiple panels and a differences view. In this section, we will provide an overview of CroP and its functionalities, presenting its capabilities for data analysis, the representation of multiple variables, the layouts provided by each type of panel, and its ability to handle multiple datasets.

A. USER INTERFACE

CroP's user interface is divided into an options sidebar and a workspace (Figure 1). In the options sidebar, users can import data files and manage datasets, while the workspace consists of a modular environment where panels containing the visualization models are set on a grid-based layout, which adapt to any changes made to the size of CroP's window. The options within the sidebar are categorized into different groups and their visibility can be toggled based on the needs of the user. Circular icons with question marks are located next to options with complex functionalities in order to provide users with instructions or context regarding that function. This information is contained within a text prompt that appears when the icon is hovered with the mouse.

At any point, users can select the "Save All" option to generate a file containing the current state of CroP, which includes the entire dataset as parsed by the application, clustering, panel positions, settings and parameters. This file can be loaded at any point through the "Load All" option in order to restore the workspace to its previously saved state with minimal loading times, as it bypasses the need to recalculate layouts or clustering.

B. DATA MANAGEMENT

CroP can receive relational data – which describes the edges of a network, containing all of the direct relationships between existing data points –, time-series data – ordered lists of values, describing how a propriety of each point varies over time –, and multivariate data – a set of independent quantitative variables, which can be used define general attributes for each data point without a defined order. Loaded files will be parsed and the user will be alerted any detected errors, along with the line numbers in which they occurred so that these can be more easily located and corrected. Multiple files can also be loaded and either merged or filtered. If any values in time-series and multivariate files are left blank, the application will interpret this as nodes being "inactive" at those time points or for those variables, and they will be represented in the visualization models accordingly. In support of

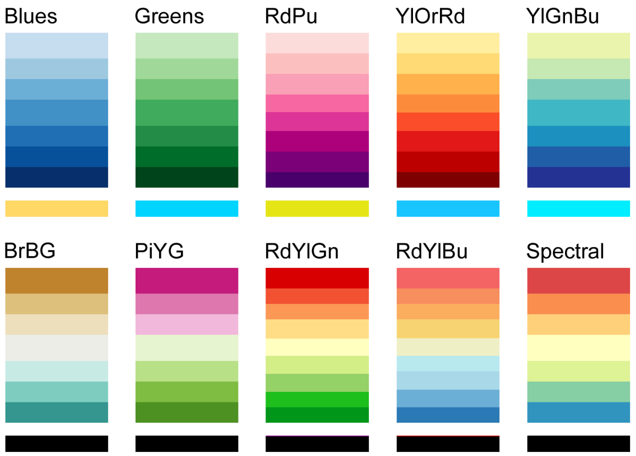


FIGURE 2. List of the color palettes available in CroP; the final color in each palette is used to represent inactivity.

biological data analysis, the Gene Ontology (GO) databases were integrated into the application in order to provide additional information and the ability to compare biological elements. Data points with names that correspond to those in the database will be associated with the biological processes of their corresponding proteins.

After a dataset has been loaded, data can be clustered into groups of nodes with similar proprieties using the options in the sidebar located under "Clustering". Users can select the type of clustering, the attribute being clustered, and the merging criteria. Data elements can be clustered by their position in the network panel, by temporal attributes, or by the values of variables. Data points and clusters selected in the visualization panels can also be removed or copied using the options provided in the sidebar under "Filtering".

C. COLOR PALETTES

In order to consistently represent the same types of values across different visualization models, we established a set of ten color palettes which were chosen based on their ability to represent various ranges of values across different types of datasets (Figure 2). In the options sidebar, different palettes can be chosen to map general numerical values, temporal progression, and differences between multiple datasets.

These palettes were based on those from ColorBrewer [64], a work in which such color schemes have been shown to distinctively represent different ranges of values while also being appropriate for users with any of the common types of colorblindness. The first five color palettes are sequential, representing low values with light tones and high values with dark tones: "Blues" and "Greens" vary between brightness within the similar hues, while "RdPu", "YlOrRd" and "YlGnBl" present variation in both their brightness and hue. While sequential palettes will generally emphasize higher values, those with higher variation in hue (in particular "YlGnBl") allow users to differentiate between values more effectively. The remaining five color palettes are divergent, representing either two very distinct colors at each of their extremes with light colors in the middle, as is the case for "BrBG", "PiYG", "RdYlGn" and "RdYlBl", or representing a larger gamut of colors, as does "Spectral". Furthermore, as CroP is able to receive and process *null* values as to explore patterns of inactivity in datasets, we established a color for each palette that highlights inactive nodes. While each sequential palette was assigned a bright color that was able to contrast against any other color mapped between its values, divergent palettes simply represent inactive nodes with black, as to contrast with their generally large variation in hue.

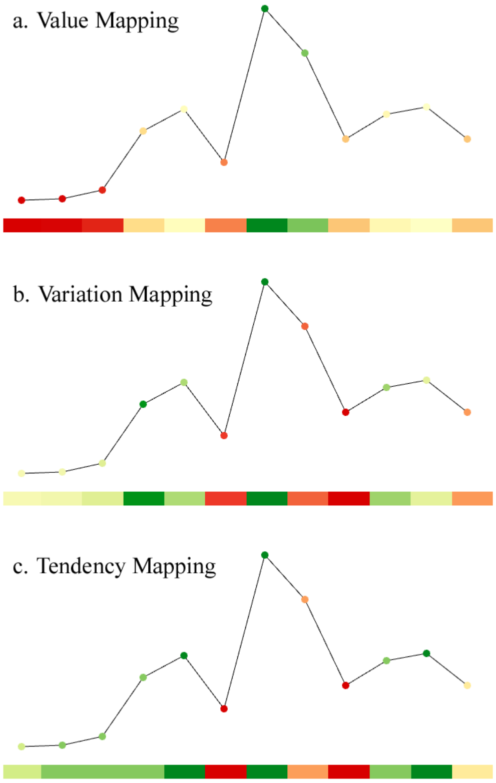
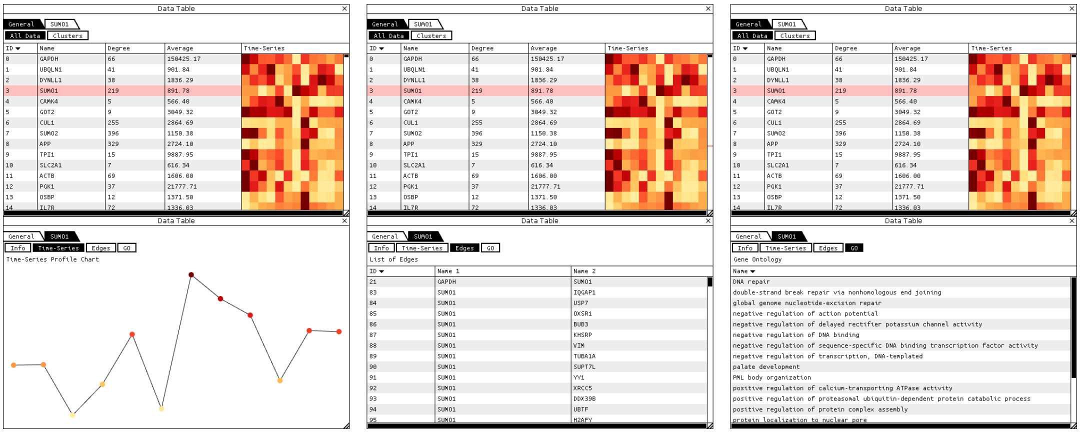


FIGURE 3. Examples of mapping colors to values (a.), variation (b.), and tendency (c.), using the "RdYlGn" color palette.

D. DATA MAPPING

To explain how values loaded from time-series or multivariate data files are mapped to a color palette, we will use the "RdYlGn" palette as a reference as it better distinguishes between extreme and middle values. CroP normalizes values by default, mapping colors between the minimum and maximum values of each time-series or variable, from red to green respectively (Figure 3.a). However, values can be unnormalized in the option sidebar, in which case colors will be mapped between the minimum and maximum values across the entire dataset.

As time-series data describes a list of ordered values, this allows for color to represent how the values change over time, using either its variation or tendency. We define variation as



Data Table

General | SUMO1

All Data | Clusters

ID ▼	Name	Degree	Average	Time-Series
0	GAPDH	66	150425.17	
1	UBQLN1	41	901.84	
2	DYNLL1	38	1836.29	
3	SUMO1	219	891.78	
4	CAMK4	5	566.40	
5	GOT2	9	3049.32	
6	CUL1	255	2864.69	
7	SUMO2	396	1150.38	
8	APP	329	2724.10	
9	TPI1	15	9887.95	
10	SLC2A1	7	616.34	
11	ACTB	69	1606.00	
12	PGK1	37	21777.71	
13	OSBP	12	1371.50	
14	IL7R	72	1336.03	

Data Table

General | SUMO1

All Data | Clusters

ID ▼	Name	Degree	Average	Time-Series
0	GAPDH	66	150425.17	
1	UBQLN1	41	901.84	
2	DYNLL1	38	1836.29	
3	SUMO1	219	891.78	
4	CAMK4	5	566.40	
5	GOT2	9	3049.32	
6	CUL1	255	2864.69	
7	SUMO2	396	1150.38	
8	APP	329	2724.10	
9	TPI1	15	9887.95	
10	SLC2A1	7	616.34	
11	ACTB	69	1606.00	
12	PGK1	37	21777.71	
13	OSBP	12	1371.50	
14	IL7R	72	1336.03	

Data Table

General | SUMO1

All Data | Clusters

ID ▼	Name	Degree	Average	Time-Series
0	GAPDH	66	150425.17	
1	UBQLN1	41	901.84	
2	DYNLL1	38	1836.29	
3	SUMO1	219	891.78	
4	CAMK4	5	566.40	
5	GOT2	9	3049.32	
6	CUL1	255	2864.69	
7	SUMO2	396	1150.38	
8	APP	329	2724.10	
9	TPI1	15	9887.95	
10	SLC2A1	7	616.34	
11	ACTB	69	1606.00	
12	PGK1	37	21777.71	
13	OSBP	12	1371.50	
14	IL7R	72	1336.03	

Data Table

General | SUMO1

Info | Time-Series | Edges | GO

Time-Series Profile Chart

Data Table

General | SUMO1

Info | Time-Series | Edges | GO

List of Edges

ID ▼	Name 1	Name 2
21	GAPDH	SUMO1
83	SUMO1	IQGAP1
84	SUMO1	USP7
85	SUMO1	OXSR1
86	SUMO1	BUB3
87	SUMO1	KHSRP
88	SUMO1	VIM
89	SUMO1	TUBA1A
90	SUMO1	SUPT7L
91	SUMO1	YY1
92	SUMO1	XRCC5
93	SUMO1	DDX39B
94	SUMO1	UBTF
95	SUMO1	H2AFY

Data Table

General | SUMO1

Info | Time-Series | Edges | GO

Gene Ontology

Name ▼
DNA repair
double-strand break repair via nonhomologous end joining
global genome nucleotide-excision repair
negative regulation of action potential
negative regulation of delayed rectifier potassium channel activity
negative regulation of DNA binding
negative regulation of sequence-specific DNA binding transcription factor activity
negative regulation of transcription, DNA-templated
palate development
PML body organization
positive regulation of calcium-transporting ATPase activity
positive regulation of proteasomal ubiquitin-dependent protein catabolic process
positive regulation of protein complex assembly
protein localization to nuclear pore

FIGURE 4. Sets of table panels showing an element being selected (top) and various information about that element (bottom): a temporal profile (left), a list of its edges (middle), and a list of its Gene Ontology proprieties (right).

the difference between the current value and that of the previous time point, as to represent the variation of values over time. Color is then mapped to the intensity of the variation and whether it is negative or positive, approaching either red and green respectively (Figure 3.b). When values have not changed significantly from the previous time point, points will approach with middle tones, in this case yellow.

Tendency consists of a simpler approach that does not take into account values, only how the data shifts between the previous and next time points. Each of the color palette's extremes corresponds to a shift in variation, where green represents a peak of values and red represents a valley, while middle colors represent other behaviors, such as light green representing increasing values, orange representing decreasing values, and yellow representing values that did not change (Figure 3.c). Tendency mapping is aimed at the analysis of datasets where shifts between positive and negative variation mark significant moments in the data, such as gene expression time-series where peaks of expression represent when proteins have become over-expressed.

E. VISUALIZATION MODELS

Visualization panels contain models and layouts that represent loaded datasets, each one dedicated to analyzing a different type of data. Every panel contains specific consistent elements that allow users to organize them: the top bar can be dragged to move the panel, and the corner of the panel can be dragged to resize it. Additionally, the bar contains a button to close the panel and a dropdown that selects the dataset being visualized when there exist multiple datasets.

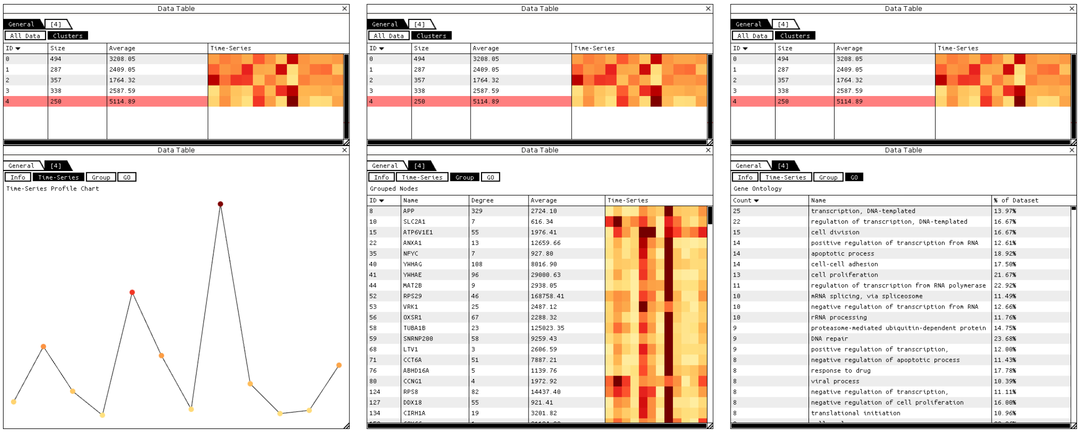
When a panel is moved or resized, its corners will always snap to the closet point on the grid of the workspace. This grid layout ensures that the organization of the workspace is maintained, as panels can be sorted and adjusted to make use of the available space. This, for instance, allows users to easily place panels next to each other and resize them to consistent sizes when comparing between multiple similar visualizations. Overlapping panels are handled automatically, where the overlapped panels are resized or moved to accommodate the new changes. There are four types of visualization panels: Data Table, Network, Time Curve, and Multivariate View. In this section, we will primarily utilize the ''YlOrRd'' color palette for values and the ''Blues'' color palette for time.

1) DATA TABLE

The data table panel shows every data point at its lowest level, listing them in a table that can be sorted by any of its columns (Figure 4). Each selected row will create a new tab on the top of the table that describes the proprieties of that point, including a line chart of its temporal profile, a bar chart depicting the values of multivariate data, a list of its edges between other nodes, and a table of corresponding Gene Ontology terms, depending on the existing data. Alternatively, selecting a cluster will also create a tab with detailed proprieties of that group, including aggregated profiles of its data, a list of its nodes and a table of aggregated Gene Ontology terms (Figure 5). Selected points are marked on the scroll bar with colored bars, and are coordinated across visualization models, highlighting these points in all network and table visualizations of the same dataset. Moreover, the ''CTRL'' key can be used to select multiple rows and create individual tabs, while the ''SHIFT'' key will toggle start/stop points that create a group containing every row between every two selections. Due to the high number of rows that could selected simultaneously, the latter type of selection the latter creates a single tab called ''Highlighted'' that contains the aggregate data on every data point selected.

2) NETWORK PANEL

The network panel consists of a dynamic node-link graph that represents data points and their relationships in two-dimensional space. The position of nodes can reflect the relationships between their attributes through multiple layouts, which allows the network panel to be used even when visualizing data that does not possess relational attributes. The resulting visualization can be panned by clicking and



Data Table

General [4]

All Data Clusters

ID ▾	Size	Average	Time-Series
0	494	3208.05	
1	287	2409.05	
2	357	1764.32	
3	338	2587.59	
4	250	5114.89	

Data Table

General [4]

Info Time-Series Group GO

Time-Series Profile Chart

Data Table

General [4]

Info Time-Series Group GO

Grouped Nodes

ID ▾	Name	Degree	Average	Time-Series
8	APP	329	2724.10	
10	SLC2A1	7	616.34	
15	ATP6V1E1	55	1976.41	
22	ANXA1	13	12659.66	
35	NFYC	7	927.80	
40	YWHAG	108	8016.90	
41	YWHAE	96	29000.63	
44	MAT2B	9	2938.05	
52	RPS29	46	168758.41	
53	VRK1	25	2487.12	
56	OXSR1	67	2288.32	
58	TUBA1B	23	125023.35	
59	SNRNP200	58	9259.43	
68	LTV1	3	2606.59	
71	CCT6A	51	7887.21	
76	ABHD16A	5	1139.76	
80	CCNG1	4	1972.92	
124	RPS8	82	14437.40	
127	DDX18	55	921.41	
134	CIRH1A	19	3201.82	

Data Table

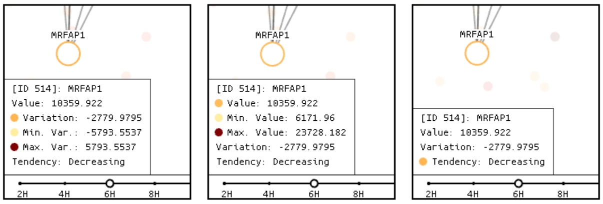
General [4]

Info Time-Series Group GO

Gene Ontology

Count ▾	Name	% of Dataset
25	transcription, DNA-templated	13.97%
22	regulation of transcription, DNA-templated	16.67%
15	cell division	16.67%
14	positive regulation of transcription from RNA	12.61%
14	apoptotic process	18.92%
14	cell-cell adhesion	17.50%
13	cell proliferation	21.67%
11	regulation of transcription from RNA polymerase	22.92%
10	mRNA splicing, via spliceosome	11.49%
10	negative regulation of transcription from RNA	12.66%
10	rRNA processing	11.76%
9	proteasome-mediated ubiquitin-dependent protein	14.75%
9	DNA repair	23.68%
9	positive regulation of transcription,	12.00%
8	negative regulation of apoptotic process	11.43%
8	response to drug	17.78%
8	viral process	10.39%
8	negative regulation of transcription,	11.11%
8	negative regulation of cell proliferation	16.00%
8	translational initiation	10.96%

FIGURE 5. Sets of table panels showing a list of clusters with one being selected (top) and various information about that cluster (bottom): an aggregated temporal profile (left), a list of its elements (middle), and a list of the Gene Ontology proprieties within the cluster (right).



[ID 514]: MRFAP1
Value: 10359.922
● Variation: -2779.9795
● Min. Var.: -5793.5537
● Max. Var.: 5793.5537
Tendency: Decreasing

[ID 514]: MRFAP1
● Value: 10359.922
● Min. Value: 6171.96
● Max. Value: 23728.182
Variation: -2779.9795
Tendency: Decreasing

[ID 514]: MRFAP1
Value: 10359.922
Variation: -2779.9795
● Tendency: Decreasing

FIGURE 6. Hovering a node will display information on its current value in relation to the current value mapping and selected time point or variable. The image displays temporal information shown for the same node when colors are mapped by value (left), variation (middle) and tendency (right).

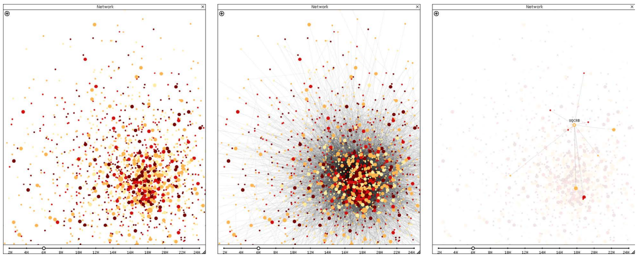
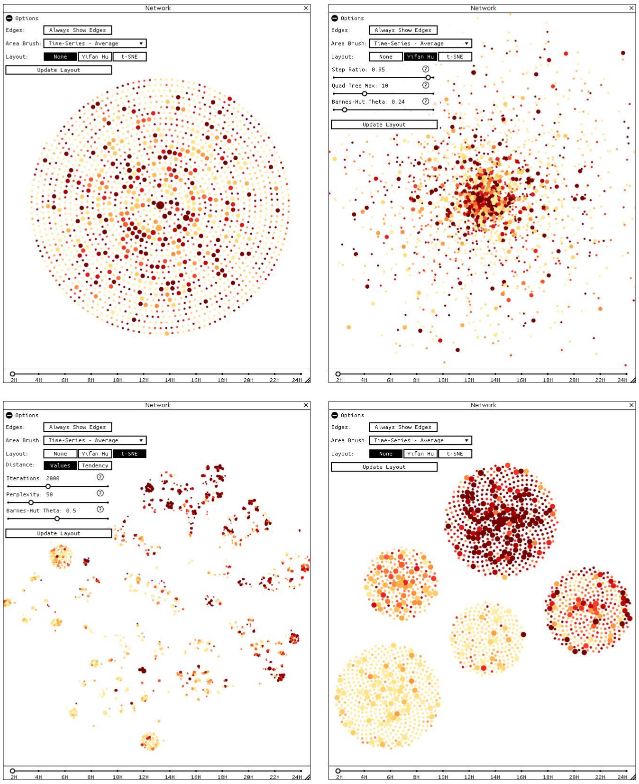


FIGURE 7. In high amounts, edges are hidden by default (left), but can be shown by selecting the "Always Show Edges" button (middle). Selecting a node will only highlight that node's edges (right).



Network

Options
Edges: Always Show Edges
Area Brush: Time-Series - Average ▾
Layout: None Yifan Hu t-SNE
Update Layout

Network

Options
Edges: Always Show Edges
Area Brush: Time-Series - Average ▾
Layout: None Yifan Hu t-SNE
Step Ratio: 0.95
Quad Tree Max: 10
Barnes-Hut Theta: 0.24
Update Layout

Network

Options
Edges: Always Show Edges
Area Brush: Time-Series - Average ▾
Layout: None Yifan Hu t-SNE
Distance: Values Tendency
Iterations: 2000
Perplexity: 50
Barnes-Hut Theta: 0.5
Update Layout

Network

Options
Edges: Always Show Edges
Area Brush: Time-Series - Average ▾
Layout: None Yifan Hu t-SNE
Update Layout

FIGURE 8. Network panels with different layouts: Sunflower (top-left), Yifan Hu (top-right), t-SNE (bottom-left), and force-directed clusters (bottom-right).

dragging the mouse, or zoomed in/out on the current mouse position using the mouse wheel, while nodes can be hovered and selected to not only highlight their names and edges, but also display a small information window with their proprieties (Figure 6).

If either time-series data or multivariate data have been loaded, nodes will be colored and sized according to the type of color mapping selected. This will also create a slider at the bottom of the panel that will either display a timeline of the time-series data, or a list of all the variables, smoothly transitioning between the colors and sizes at each step when the slider is dragged. In regards to edge representation, a common issue with the visualization of complex networks as graphs is that overlapping edges may create visual noise that provides little information and obscures other visual elements. To prevent this, we map the transparency of edges to their quantity, fading them out as their number increases until they are no longer drawn, only showing those from selected nodes. However, this can also be reverted in the options menu (Figure 7).

Nodes are initially displayed in a sunflower spiral layout, which utilizes the order loaded from the original file. The options menu provides two additional layouts for the network that can sort nodes based their attributes: the Yifan Hu layout [65] and the t-SNE layout (Figure 8). The Yifan Hu layout sorts nodes based on their edges, positioning them so that related nodes will be closer to each other, while the t-SNE

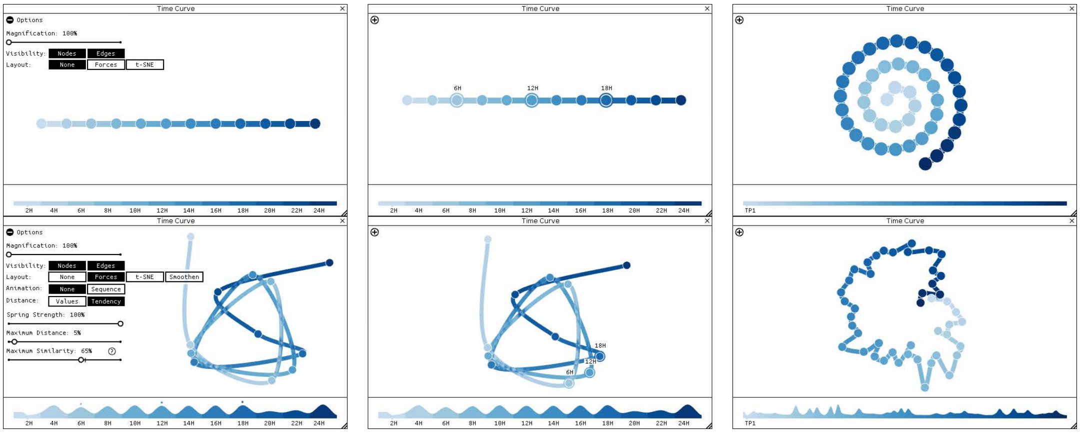


FIGURE 9. Time curve panels displaying initial timeline layouts (top) and corresponding time curves (bottom). The first timeline is sequential due to a low amount of time points (left), and when transformed into a timeline, time points representing similar states are pulled together (middle). When the number of time nodes would exceed the available space for a timeline, they are placed in a spiral layout (right).

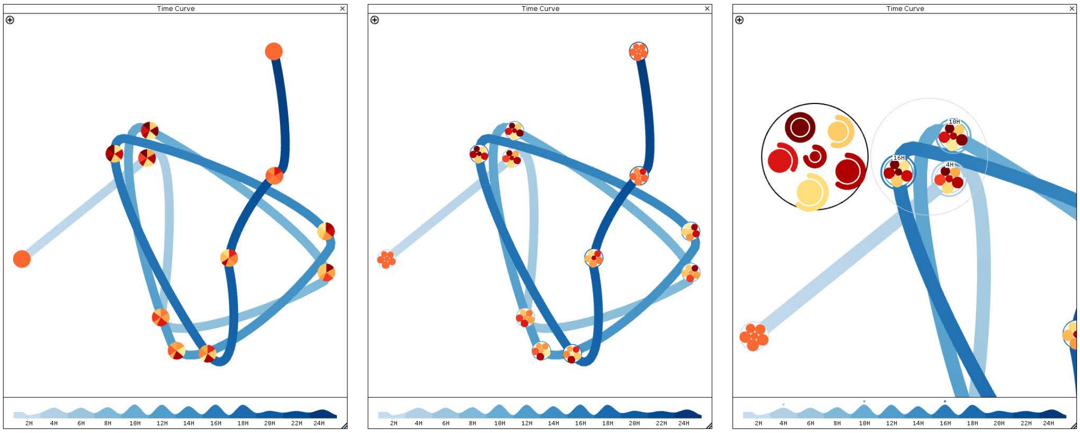


FIGURE 10. Time curve with pie chart glyphs (left) and miniature network glyphs (middle). The data lens is being used to select three time points (right), creating an aggregated visualization of their average values that depicts the similarity between them.

layout will positions nodes based on the similarity between their time-series values or multivariate data. These layouts have parameters that are mapped to sliders in the options menu which can be adjusted to balance the calculation time of the layout against its accuracy in relatively positioning nodes. Moreover, clustering the data will apply a force-directed layout over the nodes, grouping them into circular areas and sorting them into sunflower spirals that are ordered by the clustering algorithm, meaning that neighboring nodes may also be more similar to each other. The relative position of the clusters also reflects the relationships between their nodes, as clusters with nodes that have edges between them will be placed closer to each other.

3) TIME CURVE

The time curve panel focuses on the representation and analysis of time-series data through a timeline and layouts that distort in order to represent the general behaviors of the dataset over time, such as significant changes in values, regressions and cyclical shifts. The mouse can be dragged to pan the visualization, while the mouse wheel will zoom in/out of the current mouse position. Nodes can also be selected, which will update the current time step on network panels. The top left of the panel contains a list of options that offer control over visual elements, such as animations that show the flow of time, and parameters that control the current layout.

Each time point is initially converted into a node and displayed sequentially as a timeline, either in a horizontal line or as a spiral, with the latter being used when the width of the former surpasses the amount of available space (Figure 9). The timeline can then be distorted into a time curve, based on the layout proposed by Bach *et al.* [8], by positioning time points relatively to the similarity of the dataset at those times. This is achieved by using either the force-directed layout – adjustable attraction and repulsion forces that dynamically pull similar nodes closer –, or the t-SNE layout – a static, non-deterministic layout that is calculated with parameters that offer a balance between quality and processing time. To differentiate between the data states at each time point, glyphs are used to represent clustered data (Figure 10). Moreover, the resulting visualization can then be further adjusted through Time Paths, a layout that creates segmented edges

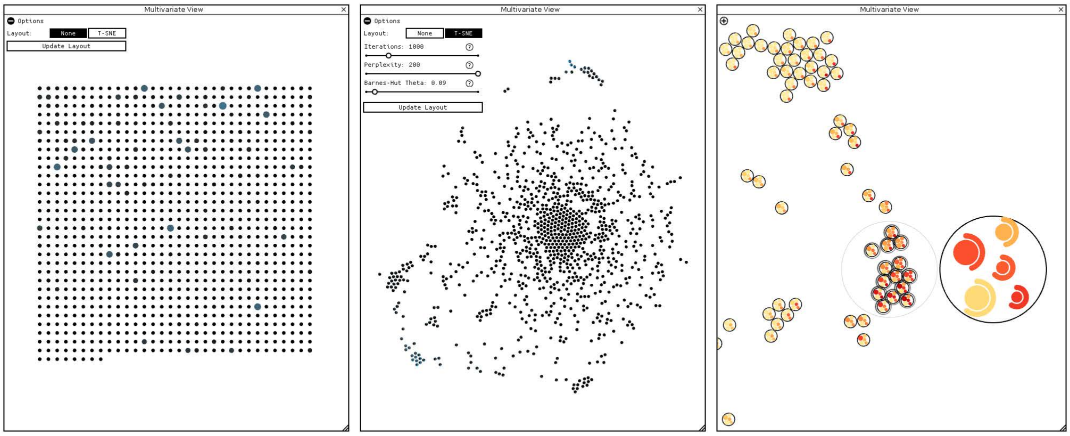
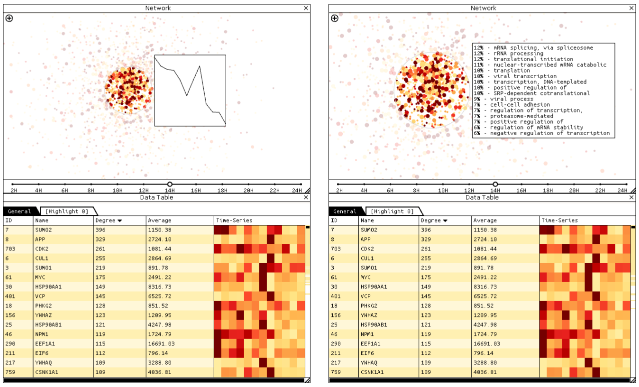


FIGURE 11. Visualization of multivariate data in a grid layout (left) and sorted by the t-SNE layout (middle). Network glyphs and the mouse lens are also available in this visualization panel when the data is clustered (right).



12% - mRNA splicing, via spliceosome
12% - rRNA processing
12% - translational initiation
11% - nuclear-transcribed mRNA catabolic
10% - translation
10% - viral transcription
10% - transcription, DNA-templated
10% - positive regulation of
10% - SRP-dependent cotranslational
9% - viral process
7% - cell-cell adhesion
7% - regulation of transcription,
7% - proteasome-mediated
7% - positive regulation of
6% - regulation of mRNA stability
6% - negative regulation of transcription

ID	Name	Degree ▾	Average	Time-Series
7	SUMO2	396	1150.38	
8	APP	329	2724.10	
703	CDK2	261	1081.44	
6	CUL1	255	2864.69	
3	SUMO1	219	891.78	
61	MYC	175	2491.22	
30	HSP90AA1	149	8316.73	
401	VCP	145	6525.72	
18	PHKG2	128	851.52	
156	YWHAZ	123	1209.95	
25	HSP90AB1	121	4247.98	
46	NPM1	119	1724.79	
290	EEF1A1	115	16691.03	
211	EIF6	112	796.14	
217	YWHAQ	109	3288.80	
759	CSNK1A1	109	4036.81	

FIGURE 12. Brushing a network with the mouse lens (top) to view the average temporal profile (left) or a count of Gene Ontology proprieties of the selected nodes. Selected nodes are also highlighted in the data table panel (bottom).

with adjustable proprieties that allows us to smoothen time curves by controlling their trajectory, curvature and transitions between colors and sizes. These parameters control the level of detail of a time curve, either smoothing it to remove visual noise in favor of portraying overall behaviors, or emphasizing small variations by distorting the timeline further.

To help better navigate and understand the time curve visualization, the bottom of the panel contains a timeline slider where time points are displayed in sequence. Dragging the slider will highlight selected time points and create an animated transition across the sequence of time points that the user brushed. Additionally, when the timeline model is distorted by a layout, the timeline slider (at the bottom of the panel) will display a supporting wave graph that represents the distance between sequential time points. As distance is mapped to similarity, large waves portray moments when significant changes in the data occurred, while flat sections represent periods of low changes in the data.

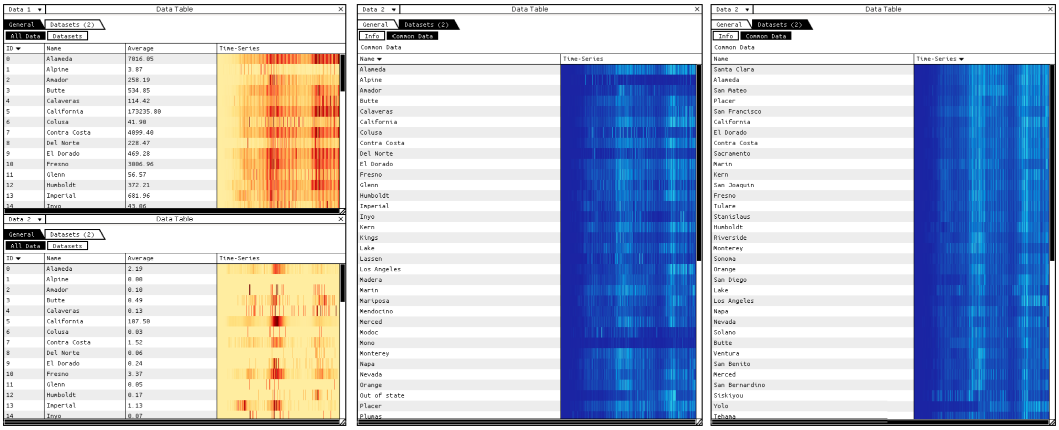
4) MULTIVARIATE VIEW

The multivariate view panel represents independent, quantitative variables as points in two-dimensional space, providing tools to analyze relationships and discover patterns of correlation between their values, similarly to the time curve panel. The mouse can be dragged to pan the visualization and the mouse wheel will zoom in/out of the current mouse position. Nodes can also be selected, which will update the current variable focus on network panels, and the top left of the panel contains an options menu with proprieties for the existing layouts. Nodes are initially displayed in a grid layout (Figure 11), which helps list individual variables while distinguishing this panel from others, and through the t-SNE layout they can be positioned relatively to their values, grouping similar nodes.

F. VALUE AGGREGATION

If time-series or multivariate data have been loaded, or if the names of biological nodes correspond with those of the Gene Ontology database, then an additional analysis tool will be available: the mouse lens. Right-clicking anywhere on the network panel will create a circle around the mouse which will act as a lens, following the mouse and selecting every node inside of it. The aggregated data of every node that is selected in this manner will be displayed in a small visualization next to the lens (Figure 12). For time-series and multivariate data, the visualization will be of a line chart that depicts an average of all the values for every node selected. For biological nodes, the lens will show the percentage of each gene ontology propriety that exists within the selected group. The size of the lens can be increased or decreased with the mouse wheel, and it may be used to select large areas of nodes.

To better discern between different states of the data without having to solely rely on other views, we created glyphs that portray the dataset at each time step or for each variable. However, representing every data point through a simple glyph may be unfeasible for significantly large datasets, so instead we represent the groups of similar points created through clustering. As such, applying clustering will convert the nodes in the time curve and multivariate view panels into glyphs (Figure 10), which can take the form of either a



ID ▾	Name	Average	Time-Series
0	Alameda	7016.05	
1	Alpine	3.87	
2	Amador	258.19	
3	Butte	534.85	
4	Calaveras	114.42	
5	California	173235.80	
6	Colusa	41.90	
7	Contra Costa	4099.40	
8	Del Norte	228.47	
9	El Dorado	469.28	
10	Fresno	3006.96	
11	Glenn	56.57	
12	Humboldt	372.21	
13	Imperial	681.96	
14	Inyo	43.06	

ID ▾	Name	Average	Time-Series
0	Alameda	2.19	
1	Alpine	0.00	
2	Amador	0.10	
3	Butte	0.49	
4	Calaveras	0.13	
5	California	107.50	
6	Colusa	0.03	
7	Contra Costa	1.52	
8	Del Norte	0.06	
9	El Dorado	0.24	
10	Fresno	3.37	
11	Glenn	0.05	
12	Humboldt	0.17	
13	Imperial	1.13	
14	Inyo	0.07	

Common Data

Name ▾	Time-Series
Alameda	
Alpine	
Amador	
Butte	
Calaveras	
California	
Colusa	
Contra Costa	
Del Norte	
El Dorado	
Fresno	
Glenn	
Humboldt	
Imperial	
Inyo	
Kern	
Kings	
Lake	
Lassen	
Los Angeles	
Madera	
Marin	
Mariposa	
Mendocino	
Merced	
Modoc	
Mono	
Monterey	
Napa	
Nevada	
Orange	
Out of state	
Placer	
Plumas	

Common Data

Name	Time-Series ▾
Santa Clara	
Alameda	
San Mateo	
Placer	
San Francisco	
California	
El Dorado	
Contra Costa	
Sacramento	
Marin	
Kern	
San Joaquin	
Fresno	
Tulare	
Stanislaus	
Humboldt	
Riverside	
Monterey	
Sonoma	
Orange	
San Diego	
Lake	
Los Angeles	
Napa	
Nevada	
Solano	
Butte	
Ventura	
San Benito	
Merced	
San Bernardino	
Siskiyou	
Yolo	
Tehama	

FIGURE 13. **The original data of the two selected datasets is represented in data tables on the left, while their average differences are depicted in the table in the middle. This table can be sorted by its differences (right).**

miniature network (circles which represent the clusters in the network panel) or a pie chart (a space-filling layout where each slice represents a cluster and its proprieties), which is provides additional visibility at smaller sizes.

The mouse lens can be used to select these glyphs, which will create a larger version of the miniature network next to the lens where the color of each circle now represents the average values of each cluster between the highlighted glyphs (Figure 10). More importantly, arcs are also drawn around every circle, indicating the percentage of similar behaviors exhibited by data points within that cluster at the selected glyphs. In other words, if around half the nodes within a cluster are exhibiting the same behaviors in the nodes hovered by the mouse lens, then the arc around that cluster's respective circle would be drawn as a semi-circle.

G. MANAGING MULTIPLE DATASETS

While multiple files can be loaded and merged to combine different sets of data points and attributes, CroP also allows datasets to be stored, accessed and managed individually. Individual datasets are stored within tabs, which can be accessed and managed in the options sidebar. Sections of interest can be copied into new tabs, allowing users to focus on specific groups of data without altering the dataset.

Multiple datasets can be visualized simultaneously through juxtaposition by using multiple panels, or compared through a differences view which can highlight patterns of similarity or dissimilarity. Loaded datasets will be listed in the data table panel, and selecting two or more dataset rows will create a table tab containing a small visualization that shows the average similarity between every node, in addition to a differences view table. This differences view consists of a list of every common data point shared between the selected datasets, where a column depicts a color matrix visualization of the time-series or multivariate data from each data point that represents the difference between the values across all selected datasets (Figure 13). The color difference is represented by a separate palette that can be changed in the options sidebar, and it is depicted by default with the YlGrBu palette, which emphasizes extreme and middle values, where dark blue represents high similarity and light yellow represents high differences. Selecting this column will order all the data points by the sum of their differences, allowing users to either sort by those that are the most similar or the most different.

When multiple datasets are selected in the data table panel, the color of nodes in network panels will also be mapped to their differences. Moreover, points that do not exist across all selected datasets will not be shown in the differences view and will be represented in the network panels using transparency.

IV. PATTERN ANALYSIS

Through its coordinated multiple views framework, CroP's functionalities can be used in conjunction in order to different types of relationships, such as analyzing potential patterns between multiple proprieties across large sets of data points, in particular relational, temporal and certain biological proprieties. As described previously, some of these functionalities are also common between visualization models, although their purpose and outputs may vary according to the type of data. In this section, we will describe how the implemented models and functions can be used individually or cooperatively to explore different datasets and discover meaningful patterns.

A. VISUALIZATION LAYOUTS

CroP integrates a variety of visualization models and layouts with the objective of allowing users to visualize, explore and analyze different types of data. For instance, the data table panel represents low level data through lists and linear visualizations, supporting other panels through coordination. By loading only necessary rows, these lists are able to contain the entire dataset with minimal processing cost, providing tools to order and brush between large sections, creating aggregate data visualizations.

The remaining visualization panels make use of different types of layouts to position data relatively to their relationships. While the initial positions are determined by

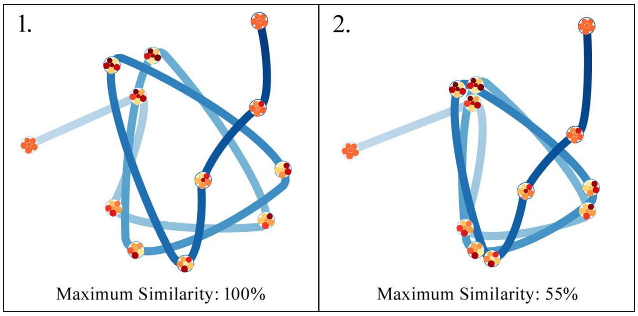


FIGURE 14. Time curve visualizations of the HIV-1 virus gene expression time-series dataset (7589 data points) showing how maximum similarity is mapped to the minimum distance between time points, reflecting their percentage of similarity.

simple space-filling layouts that aim to display all nodes without overlaps, users are then provided with other layout options specific to each panel with adjustable parameters. For instance, the Yifan Hu and t-SNE layouts use parameters that balance the quality of their results and the processing time required for their calculation, allowing users to choose between quality and speed. While the Yifan Hu layout is specific to sorting networks based on edge data, the t-SNE layout can be used to both sort data points based on their attributes (network panel) or sort attributes based on the whole dataset (time curve and multivariate view panels).

Alternatively, there is also a dynamic force-directed layout that is used not only to create clusters in the network panel, but also to bend timelines through adjustable parameters in the time curve panel. These parameters control the strength of the forces, the size of the layout, and the maximum similarity that is mapped to the distance between nodes. In other words, the maximum similarity slider defines the maximum percentage of similarity between two time points that will be mapped to their minimum distance from each other. For instance, at a maximum similarity of 55%, time points placed together will represent states where the dataset is at least 55% as similar, as illustrated in Figure 14. By adjusting this parameter, the layout can be adjusted to datasets that may have patterns between only a small percentage of data points. Moreover, to more quickly identify such behaviors in any dataset, the highest value of maximum similarity between any time point in the current dataset is marked on the slider.

In comparison to the t-SNE layout, the force-directed layout generally has more difficulties in dealing with larger datasets, as the convergence of large quantities of nodes may be very slow. However, it is comparatively more effective on smaller datasets, displaying consistent results with more accuracy.

B. CLUSTERING ALGORITHMS

In addition to the visualization models, clustering plays an important role in the analysis of both simple and complex datasets. If a dataset is represented in a network panel, points can be clustered spatially, while if time-series data has been loaded, data points can be clustered by their values, variation, tendency, or patterns of inactivity, if the dataset contains null values. As clustering can be an important step in the discovery of new knowledge through the identification of meaningful data patterns, CroP provides a range of clustering options for analyzing different types of datasets: Hierarchical, K-means, Bisecting K-means, DBSCAN and OPTICS.

Hierarchical clustering may result in long processing speeds but allows the user to select between any number of clusters after being calculated only once [25]. K-means has a fast execution time, but requires the user to specify the number of desired clusters beforehand [28]. These proprieties are shared by the bisecting k-means algorithm, which is also able to recognize clusters of any shape and size, iterating over each bisection step multiple times to improve its results [30]. DBSCAN and OPTICS are also effective at discovering clusters of different shapes and sizes [31], [32], but require very specific search parameters, as points that do not meet the established requirements are sectioned off as noise. In particular, OPTICS orders points hierarchically and utilizes an additional parameter to further filter noise.

As such, hierarchical clustering may be best suited for smaller datasets, allowing users to quickly explore the diversity of existing value profiles based on how clusters form and divide using just default parameters. Meanwhile, large datasets may be more quickly clustered through k-means and even further refined through bisecting k-means if the user is willing to search for an optimal number of clusters. On the other hand, DBSCAN and OPTICS are able to quickly determine very cohesive clusters in large and complex datasets with the correct parameters, but these are also much more sensitive than those of other algorithms and may require more effort to reach desirable results.

The implemented hierarchical clustering is based on Michael Anderberg's approach [66] described by Müllner [67], while the remaining algorithms were implemented through the SPMF open-source data mining library [68]. When clustered, nodes in the network will be grouped into circular areas representing each of the calculated clusters. In order to organize nodes within each cluster and prevent overlapping, these are sorted with a sunflower spiral layout which is ordered by the chosen clustering algorithm, meaning that neighboring nodes may also be more similar. Multiple data tables can be juxtaposed to compare between multiple individual or aggregated profile charts, as well as to quickly navigate through attribute information, as panels focused on attribute tabs will update dynamically with the user's selections.

C. CLUSTER ANALYSIS

The mouse lens is a circular brush for selecting large areas of nodes and analyzing their combined proprieties through small data visualizations. While this lens can also be used to quickly view information from individual nodes, such as line graph of their value profiles or a table of Gene Ontology proprieties, it is best used to visualize the predominant proprieties of nodes positioned by visualization layouts or grouped through the clustering algorithms. Regarding the latter, clustering the

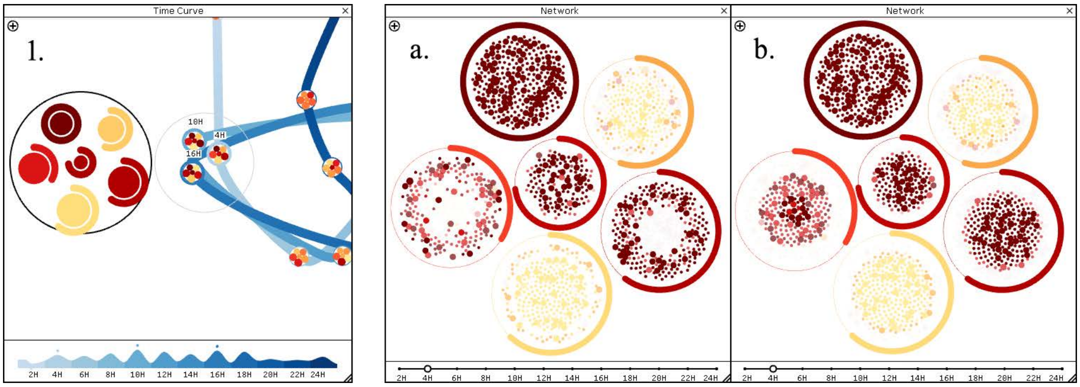

Time Curve
1.
10H
4H
16H
2H 4H 6H 8H 10H 12H 14H 16H 18H 20H 22H 24H
Network
a.
2H 4H 6H 8H 10H 12H 14H 16H 18H 20H 22H 24H
Network
b.
2H 4H 6H 8H 10H 12H 14H 16H 18H 20H 22H 24H

FIGURE 15. Time curve with three time points hovered by the mouse lens (1.) creating a larger visualization that measures the similarity of each node through their surrounding arc. The similarity between time points is mirrored in the network panel through the same arcs and through node transparency (a.), where nodes with consistent behaviors are less transparent. Network clusters can be orders by this similarity (b.), where the nodes with the highest similarity between the current selection are closer to the center.

dataset will convert nodes in the time curve and multivariate view panels into glyphs that

These glyphs represent the average values of the clustered dataset for each time point or variable, respectively, using either a miniature network or a pie chart. The miniature network glyph is a simplified representation of the visualization in the network panel, converting every cluster into a circle whose size and position is mapped to that of the cluster and colored based on the average values of the cluster. Furthermore, this also allows the glyph to be created without clusters that have been classified as ‘‘noise’’ by the DBSCAN or OPTICS clustering algorithms, allowing the remaining clusters to be more visible and easier to analyze. However, the miniature network becomes more difficult to be read when the glyph is too zoomed out, mainly due to the white space between cluster nodes, in which case the pie chart glyphs are used. Each slice of the pie chart glyph represents a cluster, where the width of its arc represents the number of nodes in the cluster and the color corresponds to the average properties of its data points. The pie chart slices are sorted relatively to the positions of the clusters on the network visualization, allowing users to more easily match each slice to its corresponding cluster.

Brushing these glyphs with the mouse lens will create a large version of the network glyph, mapping the color of each cluster to the average values between those that are brushed. Furthermore, this visualization acts as a differences view, where each cluster node is drawn with a surrounding arc whose diameter is mapped to their similarity, a percentage that is calculated using the sum of differences of each data point between all the selected glyph nodes. This is demonstrated in Figure 15.1, where three time points are selected with the mouse lens and the resulting visualization shows arcs of different lengths surrounding each cluster: the top cluster has a full circle, indicating that every data point in that cluster has consistent behaviors across all three time points, while the bottom-most cluster shows that only two-thirds of its data points are behaving similarly.

While glyphs and the mouse lens provide the means to identify patterns of values between attributes without relying on additional panels, these functionalities are also coordinated with the network panel to help dig-down into the discovered patterns. When nodes in the time curve and multivariate view panels are selected through the mouse lens, the arcs drawn around the clusters on the lens will also be drawn around their respective clusters on the network panel. Moreover, the transparency and saturation of nodes in the network will be mapped to their similarity between the selected nodes (Figure 15). As such, the more consistent each data point's values are across the selected time points or variables, the less transparent their corresponding network nodes will be, highlighting them over data points with inconsistent values.

To help better identify individual nodes with high similarity in each group, the cluster can be ordered by the similarity of each node, resulting in the most similar nodes being ordered from the center to the outside (Figure 15.b). This allows users to more easily select the most similar nodes by brushing the center of the cluster with the data lens and isolate them if needed. However, this requires the order of each cluster to be recalculated every time that the user changes the nodes brushed by the mouse lens, which may be visually overwhelming if the user is actively using the lens, as every affected node must be re-positioned. Due to this, the user can switch between ordering clusters by similarity, or to maintain their original order in the network panel's options menu.

D. EXPLORING TIME-SERIES

In what specifically concerns the analysis of temporal data, the existence of a sequential order of values allows for the representation and analysis of their shifts between time points. To this end, data points can have their color mapped to either their values, variations over time, or tendencies. This allows visualizations to portray and highlight different types of temporal behaviors, such as positive or negative trends, as well as the moments when these tendencies shift, marked with peaks and valleys of values. Such events are particularly significant in specific datasets, such as gene expression time-series data where peaks mark the moments when proteins are over-expressed.

The general temporal behaviors of the entire dataset are reflected in the time curve visualization, which depicts the similarity of values, variation and tendency between time points. For instance, large distances between two sequential nodes will indicate a notable shift in values, clusters of sequential nodes portray periods when the data did not change significantly, and cycles or regressions are represented by the curve travelling multiple times between two or more groups of non-sequential nodes. However, constant erratic changes in values may result in complex time curve visualizations with visual noise caused by significant amounts of overlapping edges. Moreover, while the Processing library provides several options to manage the proprieties of the edges, its methods offer limited control over visual attributes.

In order to not only to surpass the limitations of the basic edge drawing tools in the Processing library, but also to help create more comprehensible time curve visualizations, we developed Time Paths. Time Paths is a layout that redraws time curve visualizations through a brush controlled by parameter-based attraction forces, creating segments that allow for the creation of better transitions between colors, opacity and line weight. The brush consists of a moving point which is first placed at the initial time node on the original time curve and it is then pulled towards the following time node using a spring, calculated using Hooke's law [69] and a fixed attraction strength. The brush's route is mapped by intermediate points that are left behind as it moves between time nodes. However, the transition between nodes is not instantaneous as we apply momentum: a percentage value that defines how quickly the attraction force from the previous time node is converted into the attraction force to the next node. We defined two variables that can be controlled through sliders which update the layout dynamically:

Intermediate Points — Defines the number of points that make up the edges drawn between time points, controlling the visual definition of each curve; extreme values will cause distortions.

Smoothness — Controls the speed of forces converging between time points, previously described as momentum, where lower values create sharp turns between points, and higher values result in wider loops.

The resulting timeline is defined by the sets of intermediate points that were left in its path, which allow for increased control over its visual representation as we can define gradual transitions of visual proprieties between any time node. The increased control over edge representation allowed for addition of two animations that convey the flow of time: a pulse created from increasing and decreasing the weight of each segment in sequence and arrow particles that move across the time curve. Both of these animations convey the intensity of variation between sequential time points, where the size of pulses and speed of arrows both increase proportionally to the difference of similarity between two time points.

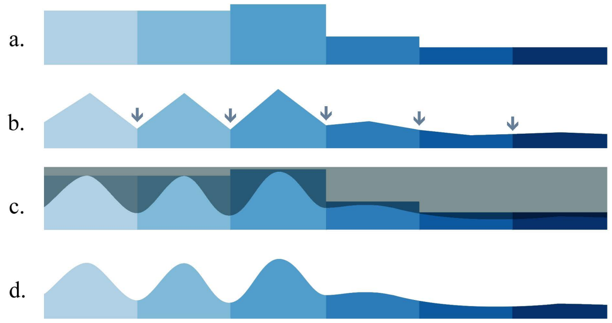


FIGURE 16. Illustration of a segment from timeline graph that represents high variation for the first three time steps and then a period of minimal changes; First it is drawn as bar chart (a), then the middle points are calculated to highlight moments of high variation (b); These points are used to create a shape that masks the initial bar chart (c), smoothing the spike shapes (d).

The calculation of a Time Path only needs to be performed once for each set of parameters, as all of the intermediate points are saved along with their properties. Moreover, it should be addressed that the layout naturally distorts the position of the time nodes from the original time curve, in which their position best reflected their similarity. To diminish this distortion, after the time path has been calculated, we move time nodes along the new path to a point that is closest to their original position on the time curve.

The analysis of temporal patterns is further supported by the timeline graph, a wave graph located on the timeline slider at the bottom of the panel. This graph shows how data shifts over time by mapping the height of each wave to the distance between that time point and the previous one. As such, large waves highlight moments of intense changes in the data, while periods of stagnation can be identified by flat segments of the graph, allowing users to identify significant events or periods in complex time curve visualizations. While a bar chart could also represent these changes in the data, waves make it easier to differentiate between concurrent intense shifts of values, representing such periods with matching visual fluctuations. Waves are created by adding points between intense shifts in the data, which are then used to calculate a shape that smoothens the graph (Figure 16).

V. EXPERIMENTATION

In this section, we present the visualizations created by our models and discuss not only their performance in representing behaviors over time in diverse datasets, but also the role they play in the discovery of significant data points or temporal events. The first experiments are produced from simple datasets comprised of single time-series which can be easily compared with the resulting visualizations, followed by experiments with biological datasets containing thousands of data points with individual time-series, in addition to a multivariate dataset. Throughout these experiments, we employ the developed methods to explore each dataset, identify patterns, and analyze their composition, sources and

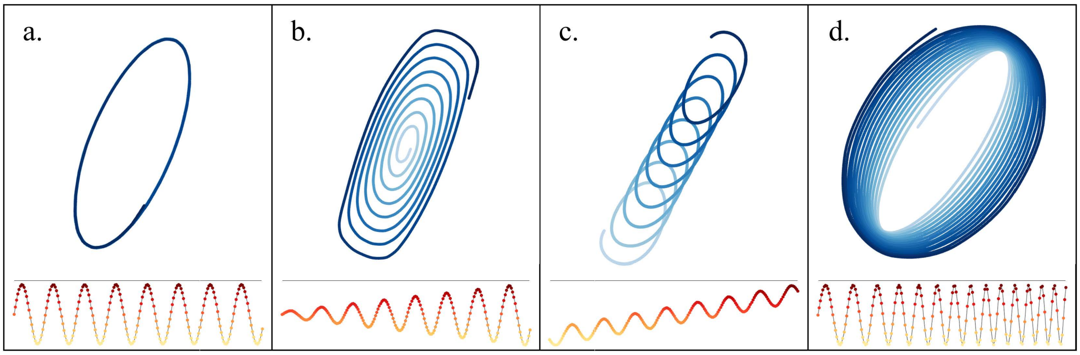


FIGURE 17. Time curve visualization of a sine wave dataset (a), followed by time curves of the same dataset with different gradual transformations: an increase of its amplitude (b), an overall increase of values (c), and an increase of its frequency (d). A linear representation of each dataset is displayed below their respective visualizations.

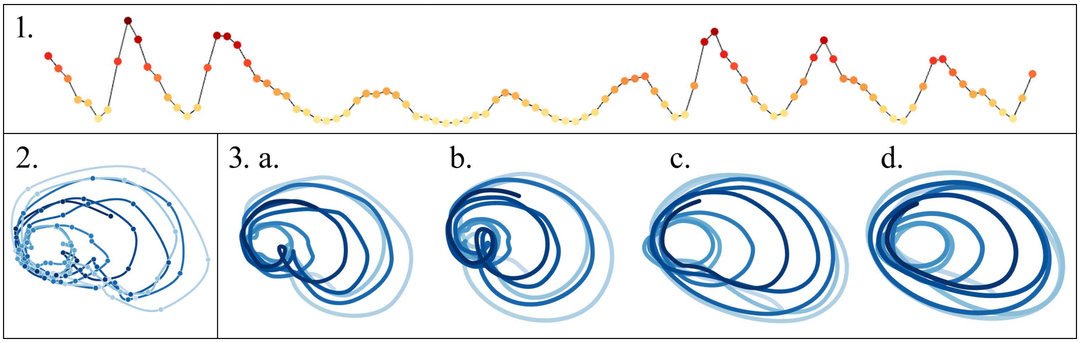


FIGURE 18. Visualizations of the "Wolfer's Sunspot Numbers" time-series, depicted as a line chart (1.), a Time Curve (2.), and then transformed through Time Paths with different parameters (3.) where the level of smoothing is increased (a to d).

impact. Regarding representation, we will primarily utilize the ''YlOrRd'' color palette for values and the ''Blues'' color palette for time, unless specified otherwise.

A. BENDING TIME

The time curve layout aims to represent changes in values across entire datasets through the distortion of a timeline. In order to test how our implementation of this layout visually translates different types of temporal behaviors, we utilized single time-series datasets that describe simple and consistent behaviors without noisy data. For this test, we chose datasets that describe sine waves across 500 time points, depicting cyclical increases and decreases of values with additional behaviors. The resulting time curves and respective time-series are presented in Figure 17. The time curve representing the initial sine wave dataset is described as an oval with overlapping loops that match both the shifts in variation over time and the number of cycles in the dataset (Figure 17.a). The second dataset resulted from a gradual increase to both the minimum and maximum values, which resulted in a matching transformation where the oval was increased to match the extreme values of each cycle (Figure 17.b). The following dataset presented a gradual increase to every value of the sine cycle, which was visually translated into each cycle shifting in position in the same direction (Figure 17.c). The last dataset featured a gradual increase to the sine wave's frequency, meaning that the variation between values increased over time, which resulted in each subsequent cycle being represented with wider loops (Figure 17.d).

These tests were followed by the representation of two time-series from real events that also depict cycles but with varying characteristics and some noise, which also serve to demonstrate how the Time Paths layout can utilize different parameters to smoothen curves, reduce visual clutter and highlight general trends. The first dataset is ''Wolfer's Sunspot Numbers'', a yearly measurement of sunspots, which are dark areas on the surface of the sun caused by concentrations of the magnetic field flux, from 1770 to 1869 [70]. This time-series is described by periodic value increases of varying intensity, which resulted in a time curve containing loops of varying sizes with minor distortions that match the inconsistent shifts in values (Figure 18). The second dataset depicts ''Monthly Milk Production'', measuring pounds per cow monthly from January of 1962 to December of 1975 [71]. The dataset is described by a yearly production cycle with consistent increasing trend which stabilizes in the last five years, and it was represented in the resulting time curve through loops consistently shifting in one direction and only overlapping during the period of stabilization (Figure 19).

The Time Paths layout was then applied over each of these time curves using four distinct sets of parameters. The first results were produced using default parameters (3.a), smoothing both of the original time curves while still

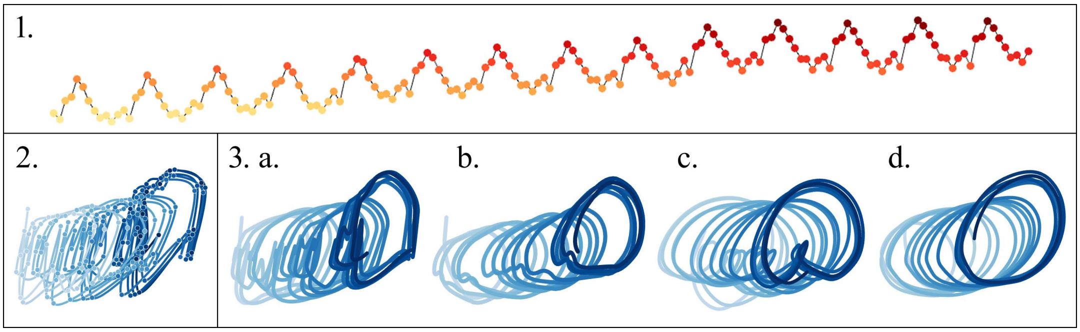


FIGURE 19. Visualizations of the "Monthly Milk Production" time-series, depicted as a line chart (1.), a Time Curve (2.), and then transformed through Time Paths with different parameters (3.) where the level of smoothing is increased (a to d).

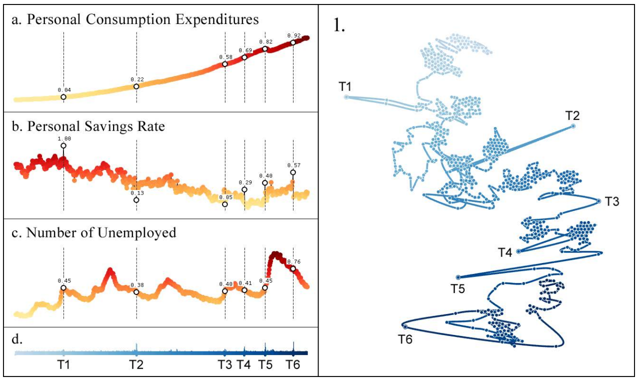


FIGURE 20. Time-series visualizations of personal consumption expenditures (a.), personal savings rate (b.) and number of unemployed (c.) in the U.S. between 1967 and 2015, along with their timeline (d.) and time curve (1.) visualizations. Time points marking significant changes in the data are highlighted (T1 through T5).

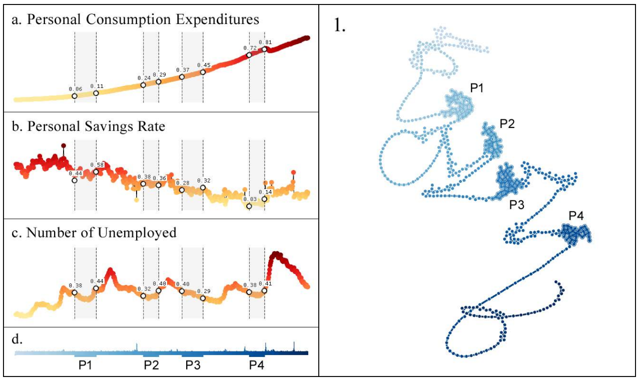


FIGURE 21. Time-series visualizations U.S. economic time-series (a., b. & c.), and respective timeline (d.) and time curve (1.) visualizations. Time curve is smoothed by the time paths layout, and four clusters of data points are highlighted, indicating periods of low variation in the data.

representing some of the smaller variations in the values. With the increase of momentum and decrease of intermediate points, the following Time Path visualizations progressively lose the details that represent minor variations in favor of visually exaggerating overall behaviors. As such, the last set of Time Path visualizations (3.d) consist primarily of the loops that describe the cycles in each dataset, but also highlight some of the largest shifts in the data.

Finally, to demonstrate how the models handle simultaneous time-series, we visualized a United States economic dataset that lists personal consumption expenditures (in billions of dollars), the personal savings rate and the number of unemployed (in thousands), from July of 1967 to April of 2015. When viewing each of these time-series through the data table panels, we can observe that expenditures have had a consistent increasing trend, personal savings had an overall decreasing trend with some outliers, and unemployment numbers are characterized by slow changes with various peaks.

Figure 20 shows these time-series along with the resulting time curve visualization, highlighting six time points that signal moments of significant changes in the data (T1 through T6). Some of these appear to be a result from outliers in personal savings, although some can be observed to match events on other time-series. For instance, T1 marks the highest point in the rate of personal savings as well as one of the initial peaks of unemployment numbers. More notably, T5 marks the moment where consumption expenditures broke its consistent raising trend, followed by a significant increase of unemployment. In the time curve, this period of time is represented with wider distances between time points, indicating stronger shifts in values and setting it apart from the rest of the visualization. This period matches the deep recession of 2007 and 2008, caused by the collapse of the housing bubble. Additionally, the time curve was also smoothed by time paths in order to decrease some of its visual noise in favor of portraying the general behaviors observed across its time-series, as shown in Figure 21. The resulting visualization presents an overall direction which reflects the main tendencies observed in expenditures and savings, with several loops and clusters that match the peaks of unemployment number and their stabilization, respectively. In this regard, four clusters of time points have been highlighted, which match periods of low changes both in unemployment numbers and rate of savings.

B. HIV-1 VIRUS

While general tendencies and outliers can be more easily identified in single time-series, gene expression time-series

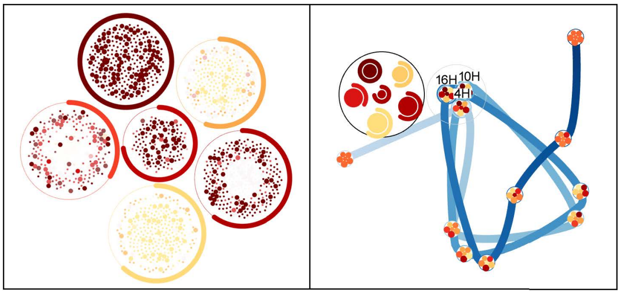


FIGURE 22. Network (left) and time curve (right) representations of the HIV-1 virus gene expression time-series dataset. Data has been clustered into 6 groups through the bisecting k-means clustering algorithm, and the 4H, 10H and 16H time points are selected with the mouse lens.

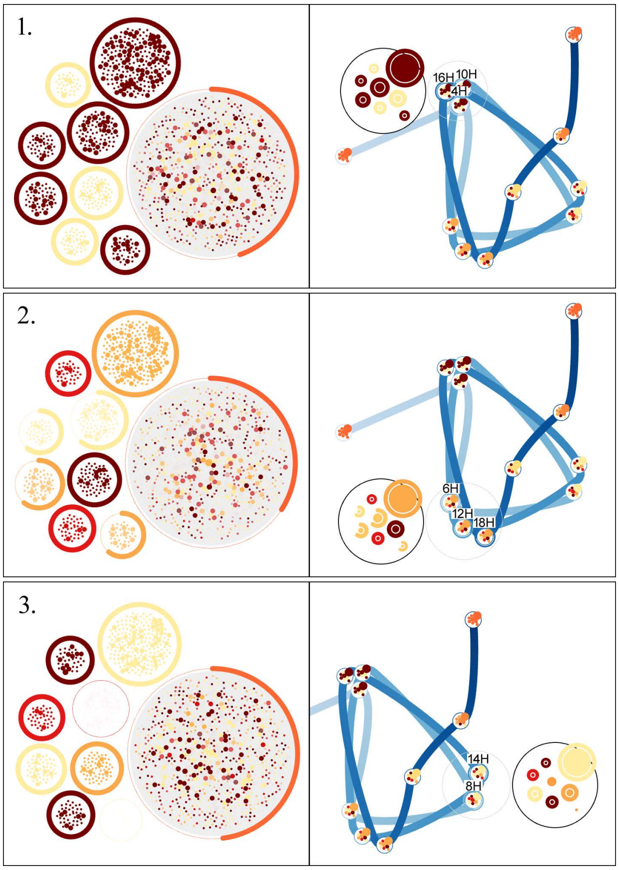


FIGURE 23. Visualizations of the HIV-1 virus gene expression time-series dataset clustered using the DBSCAN algorithm, as shown in the network panel (left). The cluster with a grey background represents genes classified as noise and it is not represented in the time curve's glyphs (right). Each visualization reflects the behaviors across each group of time points selected with the mouse lens (1, 2 & 3).

datasets contain thousands of data points with individual and varied expression profiles. To this end, we approach the analysis of these datasets through a combination of the implemented visualization and data analysis approaches, not only with the objective of identifying hidden temporal patterns, but also to better understand their origin and characteristics.

The first gene expression time-series dataset that we visualized shows human proteins reacting to the HIV-1 infection. This dataset was obtained from Mohammadi *et al.* [72], which measured gene expression every 2 hours for 24 hours after transfection with HIV-1 in Sup-T1 cell line. Expression was profiled using SAGE-Seq and normalization was done using DESeq [73]. The network dataset is comprised of a human PPI network with 7589 proteins, after being filtered by CroP to exclude proteins that do not contain time-series data. The dataset was initially clustered by tendency into 6 groups through bisecting k-means, and its time curve visualization revealed cyclical patterns consisting of the same groups of proteins behaving similarly at non-sequential points in time (Figure 22). Each of these clusters consists of proteins that at first glance are not related, but in fact may be considered as co-expressed. We are able to discern at least two cycles through three distinct groups of non-sequential time points, where at least half of the dataset presented very similar behaviors (as maximum similarity has been set at 60%). By brushing these groups of time points with the mouse lens, it is possibly to identify the clusters of proteins exhibiting the same behaviors at these points. In Figure 22, we show the lens being used on the group of time points representing 4, 10 and 16 hours, revealing that there is a significant percentage of proteins in each cluster that present similar behaviors across the three time points. Furthermore, one dark red cluster presents full similarity, meaning that every one of its proteins presents a peak of expression at these points in time, while most others appear to show either increasing tendencies or valleys of values.

To further analyze these behaviors, we clustered the dataset using DBSCAN, which resulted in about half the proteins getting filtered as noise, but creating clusters with more consistent behaviors (Figure 23). When comparing the previous group of time points with the mouse lens again, we can clearly identify that the resulting clusters have full similarity across all time points, where 5 are exhibiting peaks of expression and the remaining show valleys of values (Figure 23.1). The other two groups of time points were also examined using this method, revealing many of the same clusters to also present consistent behaviors, although more varied between them (Figure 23.2,3). In particular, we can discern 2 clusters with minimum similarity between the 8 and 14 hour time points (Figure 23.3). Through such exploration, these types of clusters can be identified, selected and either isolated to be studied further or filtered out of the dataset.

C. MALARIA VIRUS

We analyzed a time-series of the gene expression for the intraerythrocytic developmental cycle of Plasmodium Falciparum, the agent responsible for human malaria, whose dataset contains 5080 genes with expression values measured every hour over a 48-hour period. The time curve visualization of this dataset shows a general continuous behavior throughout, where each time node remains close to the one that follows without loops or overlapping, as shown

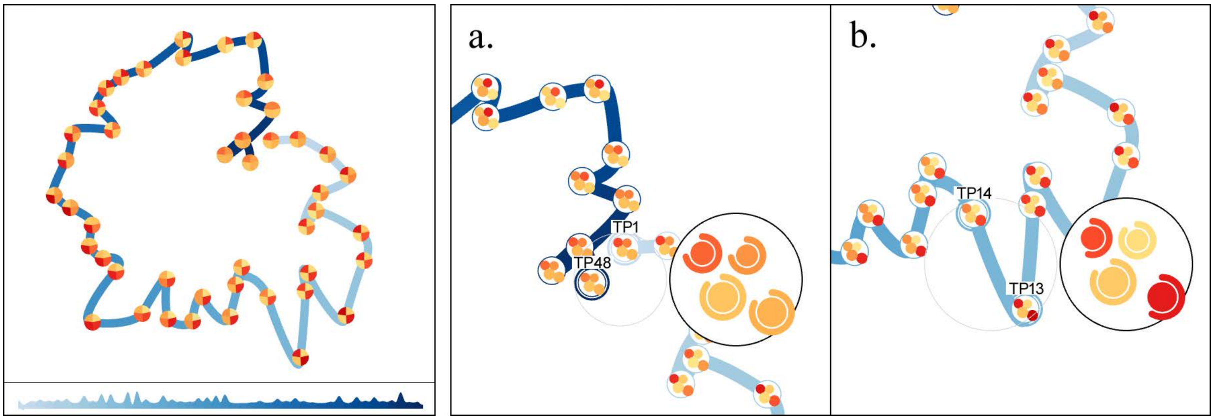


FIGURE 24. Time curve visualization of the Plasmodium Falciparum dataset with supporting timeline graph below it (left), along with close-ups of an analysis of two sets of time points (right): TP1 and TP48, which show overall similarities (a.), and TP13 and TP14, which present two clusters with a higher degree of differences (b.).

in Figure 24.a. Additionally, the time curve has a near 90% maximum similarity, meaning that the genes across the entire dataset present very similar behaviors overall. These characteristics match an existing study of this agent by Bozdech *et al.* [74] that refers to the behavior of the genes as a cascade of continuous expression that lacks sharp transitions. Furthermore, the dataset appears to return to a state similar to that of its initial time point, indicating a cycle. This is shown in Figure 24.b, where the mouse lens is used to compare between the first and last time points (TP1 and TP48), revealing that a large percentage of the dataset presents the same behaviors at those times.

However, while expression values do not appear to shift drastically, through both the time curve and the supporting timeline graph, we can identify periods of stable variation and moments of larger shifts in the data. By using the lens, we can analyze one of these shifts to identify the responsible data points. For instance, by comparing TP13 and TP14 (Figure 24.b), we can identify that the two clusters of nodes with higher values (dark colors) were responsible for that spike in the time curve, as they contain the data points with the least consistent values between the two selected time points. Furthermore, through the glyph colors, it is possible to observe that this was the result of a relatively significant increase in values between these time points.

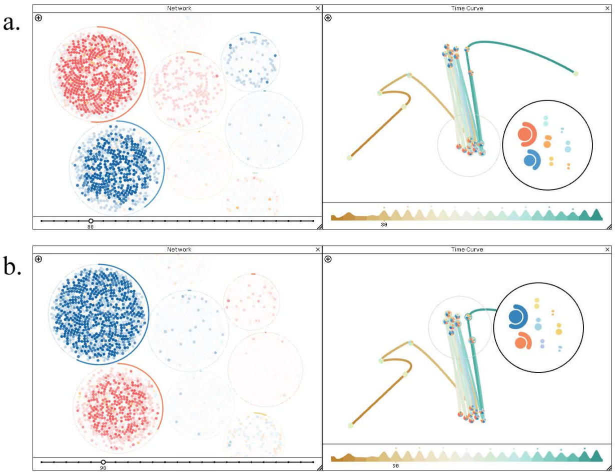
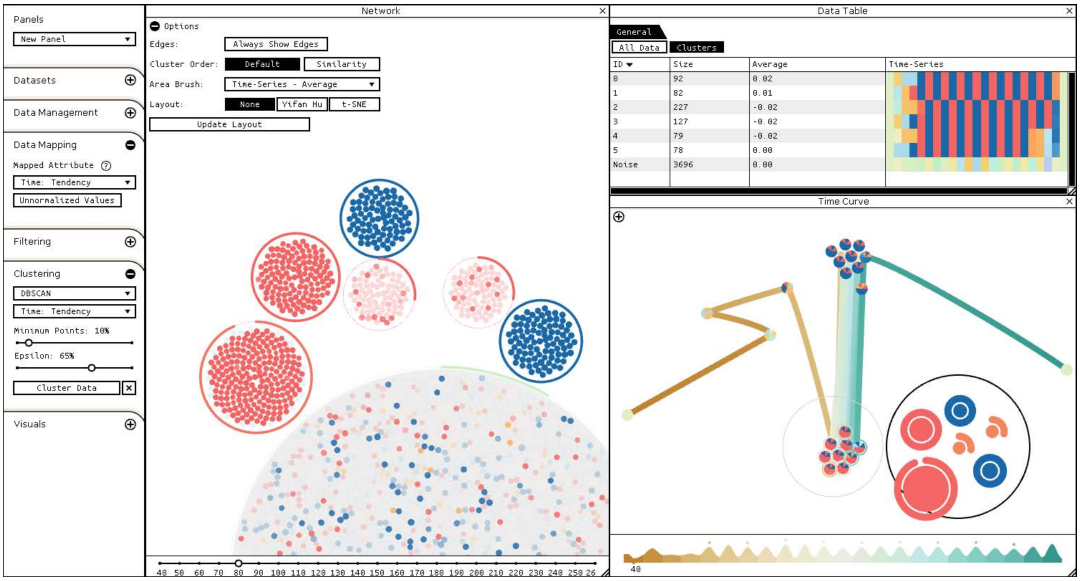


FIGURE 25. Network (left) and time curve (right) visualizations of the Saccharomyces cerevisiae dataset, where the data alternates between two states throughout most of its time-series (a. & b.). The time points corresponding to these two states are selected by the mouse lens, highlighting the network's data points that present similar behaviors in each state.

D. YEAST CELL CYCLE

We visualized gene expression data measured in Saccharomyces cerevisiae cell cultures, a species of yeast, which have been synchronized at different points of the cell cycle through a temperature-sensitive mutation (CDC15) that arrests cells late in mitosis. The dataset contains 4816 cells with expression values measured every 5 minutes for 2 hours. The dataset was first clustered into 7 groups using the hierarchical clustering algorithm, while the time curve visualization was created by positioning time points by tendency, using a maximum similarity of 50% (Figure 25). To better highlight both extremes of values, we have chosen the "RdYlBu" color palette, while time is mapped across the "BrBG" palette. The resulting time curve shows that most of dataset is initially comprised values increasing or decreasing with little correlation, changing between unique states during the first four time points. However, this followed by a behavior that is repeated throughout the remaining time points: the dataset alternates between two states where two clusters alternate oppositely between peaks and valleys of expression. Towards the end of this consistent behavior, it is possible to discern one time point that is located relatively farther from the top group. This may indicate the occurrence of an event that resulted in a break of the cycle.

Additionally, we clustered the dataset using the DBSCAN algorithm. While a significant portion of the dataset was



Mapped Attribute: Time: Tendency
Unnormalized Values

Clustering: DBSCAN
Time: Tendency
Minimum Points: 10%
Epsilon: 65%
Cluster Data

Edges: Always Show Edges
Cluster Order: Default | Similarity
Area Brush: Time-Series - Average
Layout: None | Yifan Hu | t-SNE

ID	Size	Average	Time-Series
0	92	0.02	
1	82	0.01	
2	227	-0.02	
3	127	-0.02	
4	79	-0.02	
5	78	0.00	
Noise	3696	0.00	

FIGURE 26. Screenshot of CroP visualizing the Saccharomyces cerevisiae dataset. The clusters created by the DBSCAN algorithm are represented in the network panel and listed in the data table panel. The time curve panel shows the data alternating between two states, one being selected with the mouse lens to highlight consistent peaks and valleys of values across several clusters.

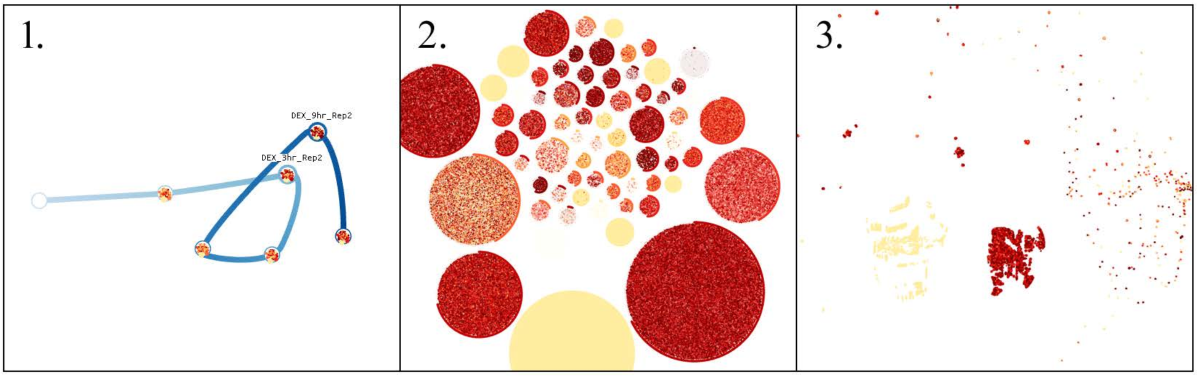


FIGURE 27. Lung cancer dataset represented through a time curve visualization (1.), a network clustered by the OPTICS algorithm (2.), and a network sorted using t-SNE (3.). While OPTICS discovered a large diversity of temporal patterns, the t-SNE layout divided most cells into two groups.

classified as noise, likely due to a high amount of variation in temporal patterns, the algorithm was capable of grouping the primary genes responsible for the previously discussed behaviors. This can be observed in Figure 26, where using the mouse lens to select the bottom group of nine time nodes shows that 4 out of the 6 clusters consist almost entirely of nodes with the same behaviors. Furthermore, the inconsistencies within the remaining two clusters appear to be caused by genes that stop behaving consistently towards the end of the timeline, which was also noted previously.

E. LUNG CANCER

To demonstrate CroP's ability to process larger datasets, we represented a dataset that explores human gene expression responses to glucocorticoids [75]. This dataset contains 119y208 cells of the human lung adenocarcinoma exposed the synthetic glucocorticoid dexamethasone, and describes their changes in gene expression every 2 hours for 6 time points. The time curve revealed a simple circular pattern with significant similarities between the data at the 3 hour and 9 hour time points (Figure 27.1). To better compare the dataset between these time points, we clustered the data using the OPTICS clustering algorithm (Figure 27.2), as it was one of the fastest available algorithms for a dataset with these characteristics. Due to the size of the dataset, the amount of distinct temporal patterns resulted in the creation of a high number of clusters. However, from these we are able to discern some particularly large clusters, one characterized by having minimal values throughout the dataset, while the remainder showed significantly high values at the 3 hour and 9 hour time points.

To further analyze these potential patterns, the data was spatially sorted using the t-SNE layout (Figure 27.3). Although the layout does not create cluster objects, it was capable of effectively sorting the points into visually distinct groups: two large groups in the center, surrounded by some small groups and a "cloud" of scattered points on the right. In comparison to the clusters created with OPTICS, the t-SNE layout more clearly divided the data points that contain consistently low values into the left group and those with significantly high values into the right group, while also separating data points with temporal patterns that do not fit any particular group. We were able to conclude that the large group of data points with high values was primarily responsible for the pattern observed in time curve: a large increase that led to peaks of values at the 3 hour and 9 hour time points, followed by a slow decrease. While there is a limited amount of time points, this may describe a potential cyclical pattern of expression that would continue happening across this group of points.

F. CORONAVIRUS DISEASE

In addition to the previous temporal datasets, we also visualized a multivariate dataset detailing the effects of the COVID-19 pandemic on the population of the state of California in the United States of America. This data was obtained from the California Health and Human Services Open Data Portal [76] and describes the number of tests, cases and deaths across every county in California between February of 2020 and January of 2022. In order to better compare data between counties, we divided the total number of tests, cases and deaths by the population as to obtain these values per capita. Due to the vast differences in values between these variables, it may be cumbersome to identify the relative significance of each variable across the whole dataset. As such, we utilize normalized variables to represent the intensity of each value in relation to every other county, as shown in Figure 28. For instance, the bars for total tests, positive tests and cases in Lassen are completely filled, indicating that this county had the highest number of these per capita in comparison to all other counties, but this does not mean that each bar represents the same value (which is written above each bar).

As with previous datasets, we clustered the data into a small number of groups in order to quickly identify any patterns in the distribution of values (Figure 29). The four network clusters, obtained through bisecting k-means, presented distinct profiles where middle cluster appears to contain all the counties with the highest values per capita across the dataset. Additionally, each variable is depicted as a node in the multivariate view, which have been positioned by the t-SNE layout. The two variables with the most similar distribution of values are "Positive Tests" and "Cases" (Figure 29.1), which is expected as a positive test would be an indication of the COVID-19 infection, unless it was a false positive. There is also a correlation between "Cases" and "Deaths" (Figure 29.2), although it is more inconsistent, possibly due to the variation in factors related to the ability of each county to handle the pandemic. Finally, the largest difference in distribution appears to be between "Total Tests" and "Deaths". The glyphs in the multivariate view show that the number of total tests per capital is relatively lower than their other variables, which the exception of the counties in the right-most cluster, where this trend appears to be inverted. In order to better understand this, we can look at the graphs in Figure 28 where we see that "Total Tests" and "Death" are often inversely proportional across these counties. It is possible that such a correlation could be attributed to prevention measures, as a higher number of tests per capita would lead to infections being detected and treated earlier, lowering death rates (and vice versa). However, such conclusions would also have taken into consideration additional factors throughout the counties such as hospital availability, number of people with health insurance, and other preconditions that may have affected these values.

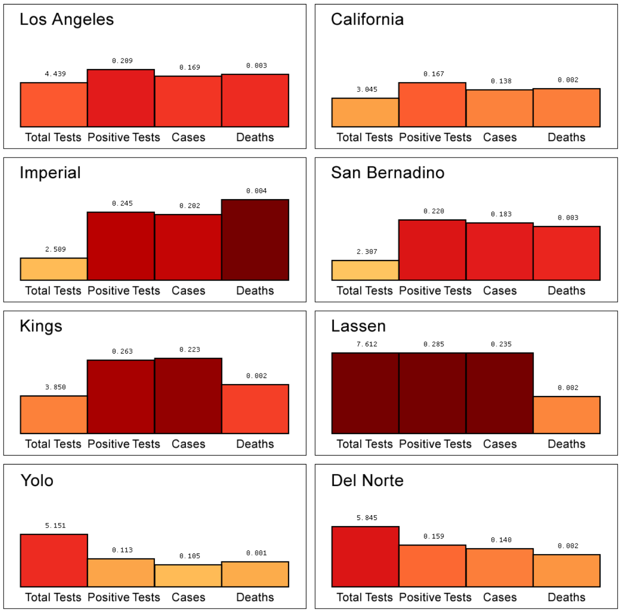


FIGURE 28. **Bar charts depicting the normalized values of total tests, positive tests, cases and deaths per capita registered during the COVID-19 pandemic for several counties in state of California.**

G. DISCUSSION

Throughout the preformed experiments, we were able to visualize and explore various types of datasets, starting with low-dimensional datasets that were used to test basic representation features, up to high-dimensional datasets which exhibited diverse behaviors across thousands of data points.

Due to the inherent complexity in creating comprehensible abstract visualizations, the initial experiments focused on the time curve visualization as its ability to represent various behaviors results from the distortion of a timeline. Through simple datasets, we were able to more easily compare the created time curve visualizations with the original time-series, allowing us to match its visuals with any observed behaviors, such as periods of stagnation and moments with intense shifts of values, as well as regressions and cycles. Moreover,

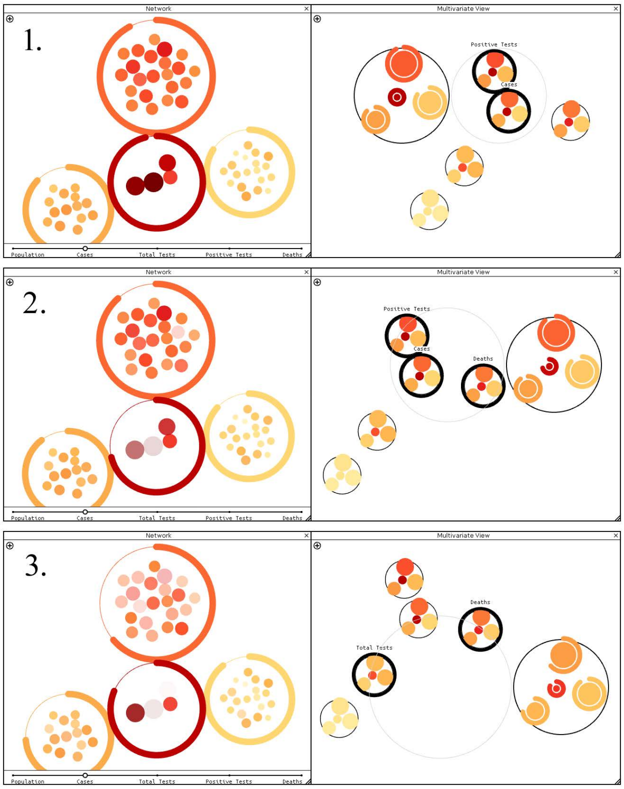


FIGURE 29. Data visualizations of the COVID-19 pandemic dataset, where each county is represented as a node in the network (left) that has been clustered by its variables, which are represented as nodes in the multivariate view (right). Using the mouse lens, several variables are being compared across all counties: the number of positive tests and cases (1.); positive tests, cases and deaths (2.); total tests and deaths (3.).

in seeking to answer how visual abstractions can be used to manage visual complexity, we developed the time paths layout to reduce visual noise in favor of representing prominent behaviors, but this comes at the cost of data fidelity. This was reflected in the preformed experiments, where manipulating the layout's parameters showed that increasing the level of smoothing highlighted overall tendencies by removing not only smaller variations, but also some outliers which could have marked significant events in these datasets. Additionally, high parameters resulted in exaggerated deformations of simple behaviors, reinforcing the need for balance between accuracy and abstraction when seeking to achieve readability. However, such exaggerations could be considered for artistic representations of datasets.

In addition to the visualization models, CroP's functionalities support the exploration of more complex datasets to facilitate their analysis and discovery of patterns of information. As shown throughout the performed experiments, the different types of clustering allowed for varying degrees of precision in the creation of groups of data points containing similar patterns. While the hierarchical and k-means clustering require fewer parameters, they were able to reveal the diversity in patterns across multiple datasets and provide a better understanding of the patterns revealed by the time curve visualizations. Additionally, the DBCSAN and OPTICS clustering algorithms were able to define more uniform groups while isolating independent patterns as noise. The composition of clusters can be explored through the data table and network panels by using juxtaposed views and coordinated brushing, while the time curve and multivariate view panels uses these clusters in glyphs to represent the state of the dataset at different instances so they can be compared. The mouse lens then allows for further exploration into the patterns revealed by the layouts of these panels, facilitating the identification of groups of nodes that are responsible for unique behaviors and relationships. For instance, while a general cyclical tendency was identified in the HIV-1 dataset it was only through the time lens that we identified the nodes that followed this behavior, despite the existence of multiple groups with different temporal profiles that followed the same cyclical pattern. Similar analysis was performed for other behaviors, such as the large shifts of values and regressions identified in the Malaria Virus and Yeast Cell Cycle datasets, whose responsible cells were highlighted by the differences represented in the data lens.

VI. VALIDATION

For the design and development of the visualization tool, we adopted Ben Fry's methodology [77], which consists of several steps that establish a path from the collection of raw data to its representation and interaction with users. This is a flexible methodology where various steps can be iterated through successive refinements and validation through user evaluation, as to progressively improve user interaction, visual encoding and data analysis.

Due to the inherent complexity of the abstract visualizations created by time curves model, some preliminary tests were performed early in the development to determine the viability and effectiveness of the model in the context of analyzing temporal patterns and discovering significant moments. With the development of a functional prototype, a round of interface tests was performed with a small group of users with a low level of experience with visualization tools, as to detect any immediate usability problems. With the progressive improvement of the tool's visuals and interaction, a more comprehensive round of tests was then performed with a wider and more varied group of users. In these tests, users were asked to not only use CroP to complete a series of tasks, but also evaluate the visualization models and provide general feedback. In this section, we will describe the performed surveys and tests, as well as discuss the results and obtained feedback.

A. PRELIMINARY MODEL SURVEY

As the time curve depicts temporal behaviors through the abstraction of a timeline, one predominant concern was the comprehensibility of the model by different types of users, in particular those minimal knowledge of data visualizations. To this end, an early survey was performed to test how users from different fields would fare in decoding time curve visualizations and in identifying significant moments or periods, in addition to an inquiry for personal feedback. This study was conducted in person with university students from various fields of study: out of the 25 participants, 8 had an Information Visualization background, 4 had a Computational Creativity background, 6 had a Computer Science background, and 7 had a Biomedical Science background.

The survey presented users with a series of time curve visualizations with increasing complexity, each with a set of questions related to the represented behaviors. Participants were first shown the time curve of the sine wave dataset (shown in Figure 17.a) as it represents a simple and consistent cycle, and were asked to associate the visualization with a behavior from a list of multiple choices. In this question, all of the participants were able to identify that the predominant behavior being represented was cyclical. This was followed by three time curves of the sine wave dataset representing gradual shifts in values and intensity (shown in Figure 17.b,c,d), where participants had to choose from a set of four distinct time-series and choose the one that was best represented by the time curve. Out of these four options, 48% of the participants correctly matched the increasing shift in variation, 86% correctly matched the consistent increase of values, and 68% were able to match the dataset representing increasing intensity.

The survey then presented time curve visualizations from the ''Monthly Milk Production'' (Figure 19) and the ''Wolfer's Sunspot Numbers'' datasets (Figure 18), asking participants to identify trends, cycle periodicity, and specific events or behaviors. Over 90% of participants were able to correctly identify both increasing trends and periods of stabilization, as well as moments with significant changes, but only 54% were capable of identifying a specific outlier in the sunspot time curve. For each of the previous two datasets, the survey presented a set of six time path transformations of each time curve, each with increasing smoothness (similar to the time paths presented in Figures 18,19). Participants were then asked to choose the one that they thought best conveyed the original dataset, the one they preferred visually, and their overall choice. We were able to observe that participants generally preferred rounder curves with less variation details, as long as the visualization was still able convey the overall behaviors of the original dataset. For instance, while participants visually preferred time curves with elevated smoothing, the Sunspot dataset visualizations with high smoothing parameters (Figure 18.d) were overall unpopular due to not representing minor variations that were considered significant.

Finally, participants were asked to rate the presented visualizations from 1 (very poor) to 5 (very good), first based on their ability to describe information or highlight behaviors, and then based their ability to draw interest or to be visually appealing. The average score given to the model's ability to represent behaviors was 4.1 out of 5, accompanied by feedback that the model should be useful tool in data analysis after the learning curve is surpassed. However, the absence of interaction in these tests did highlight some limitations on the static models, such as overlapping lines causing visual noise. Regarding the aesthetic presentation of the visualizations, participants gave an average score of 3.9 out of 5, commenting that they were generally more visually appealing than linear visualization models, but that their application is very context-sensitive.

B. INITIAL INTERFACE TESTS

The initial interface tests were conceived with the purpose of detecting overall usability problems and to better understand how users with minimal knowledge of data visualization would perform in solving their assigned tasks. Due to the limited availability of participants that would fit within our expected target audience, the tests were performed with a group comprised of 9 college students from a Biochemistry degree who had a low level of experience with visualization tools. User tests were performed in person, using datasets from our previous experiments: a human PPI network paired with the gene expression time-series dataset depicting the HIV-1 infection (depicted in Figure 1). We chose these datasets due to the large number of proteins but small number of time points, giving users a large dataset to explore with a low temporal complexity, including its time curve visualization which simply portrays two loops.

Participants were initially asked to navigate the tool and import the PPI network file (T1), followed by the gene expression time-series file, while filtering out data that was not present in both datasets when prompted (T2). The first tasks that involved the visualization panels focused on the data table panel, as it primarily contains tables and linear visualizations, tasking participants with the discovery and selection of the protein with the highest average expression values (T3) and then opening its time-series profile (T4). This was followed by prompting users to apply clustering (T5) and then select the cluster of nodes containing the protein from the previous task. Finally, participants were asked to apply a layout in the time curve panel (T6) and observe the resulting visualization. Task completion times are represented in Figure 30.

During T1 and T2, 3 of the 9 participants selected the wrong type of loading in one of the two initial tasks. At this stage of testing, the prototype utilized a single dropdown for loading datasets from which users could choose between all the supported types of data which may have contributed to these user errors, and this was taken into consideration when updating the user interface. While there were no significant difficulties in solving T3 and T4, two participants took a significant amount of time in exploring the tool and its different panels. In T5 and T6, all of the participants were able to use the menus to apply clustering and apply layouts on the time curve, and the interaction with the latter model allowed for the detection of specific usability problems when using the timeline slider to pinpoint specific time points. We identified this issue as deriving from a lack of visual feedback, as the

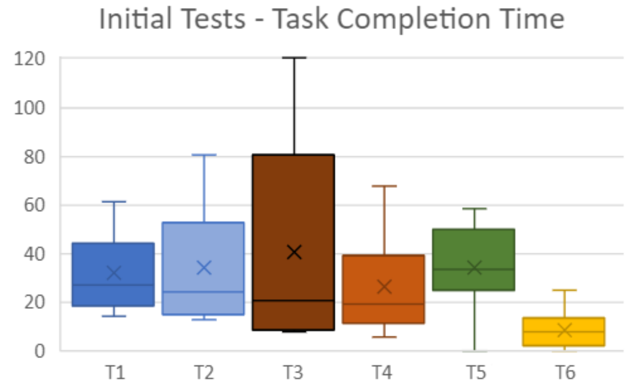


FIGURE 30. Box plot of the time taken (in seconds) by all participants to complete each of the interface tasks of the initial tests.

hovered nodes in the timeline slider were not highlighted in the time curve.

After completing the final interface task, participants were asked to associate the time curve to a temporal behavior from these options: Stagnant, Erratic, Cyclical, No pattern, or Other. To further understand the level of confidence in the user's previous answer, we asked them how clearly they were able to perceive the previous pattern, if any, on a scale from 1 (Confusing) to 7 (Clear). 7 out of the 9 participants were able to identify the pattern of the time curve as representing multiple cycles, rating the clarity of the representation with an average score of 5.6 out of 7. The remaining two users did not think the visualization represented neither continuous nor cyclical behaviors.

Finally, participants were asked to select how much they agreed with several affirmations from a 1 (Strongly Disagree) to 5 (Strongly Agree) scale, namely if they thought CroP was easy to use or unnecessarily complex, if they needed more time to learn how to use it, and if they could see themselves using it in the future. The average distribution of the given scores is represented in Figure 31, where participants generally agreed that CroP was accessible with 3.9, generally disagreed that the tool was complex with 1.6, generally agreed that they required more time to use CroP properly with 3.9, and finally generally agreed that they could see themselves using this tool in the future with 4.1.

C. EXTENDED INTERFACE TESTS

Following the feedback obtained from the previous tests, we refined the interface and extended the tests to include additional interface tasks and further model testing. These tests were conducted with 26 college students, where 16 were from the field of computational biology and 11 from information science, with varying degrees of data visualization knowledge and experience. Each user was initially introduced to CroP through a video that presented an overview of its functionalities, with minimal details, as well as a description of the data and how it is represented. The test consisted of 16 tasks divided into 5 categories (T1 through T5), followed by a set of five feedback questions, and ending with 10 questions regarding the visualization models, which are divided into 3 categories.

For these tests we once again used the gene expression time-series dataset of HIV-1 infection, not only so that these tests could be potentially compared to those done previously, but also due to the characteristics of this dataset continuing to be favorable for user testing. Similarly to the previous tests, participants began by loading the two datasets into the tool. However, upon the data being loaded into the visualization panels, users were encouraged to navigate the tool freely to reduce the potential fear of interacting with a new system. Participants were then directed to the data table panel and asked to search for the protein with the highest degree value (T2.1) and explore its proprieties to identify its highest expression value (T2.2), then select the three proteins with the highest degree values (T2.3) and filter that group off the dataset (T2.4), and finally, to encourage the use of group selection tools, they were asked to select a group of 16 points and then simply deselect them (T2.6). The next set of tasks involved a simple data analysis task directed at identifying a particular subset of the data. First, participants were tasked with clustering the dataset into five groups (T3.1) to then identify and select the cluster of proteins with the lowest expression values at 16 hours (T3.2). Afterwards, they were asked to simply deselect the cluster (T3.3).

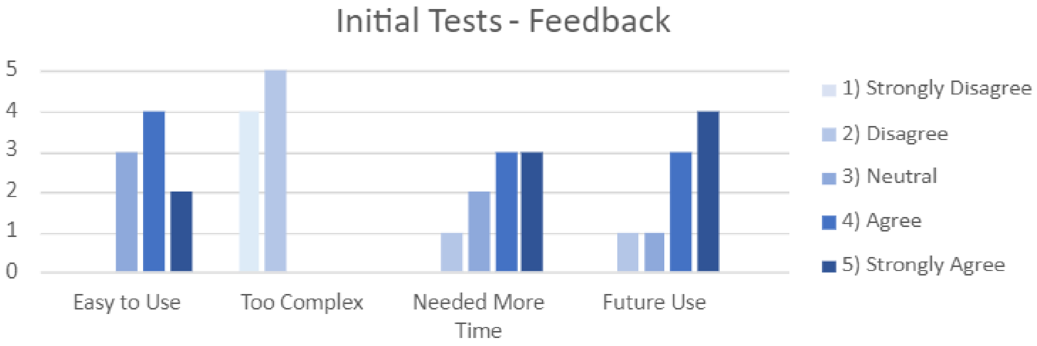


FIGURE 31. Score total given by participants for each of the affirmations in the feedback section of the initial interface tests.

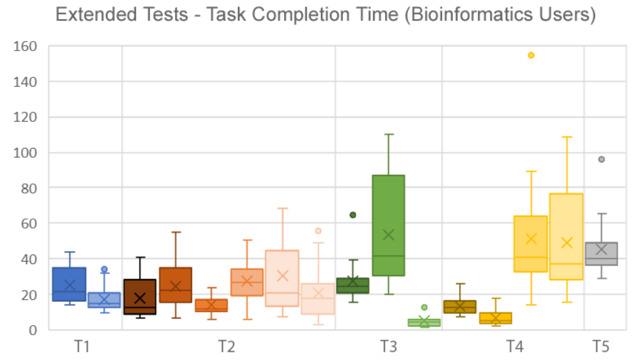


FIGURE 32. Box plot of the time taken by participants with a biological background to complete each of the interface tasks of the initial tests. Groups of tasks that are similar with those of the initial tests, in Figures 30, have matching colors.

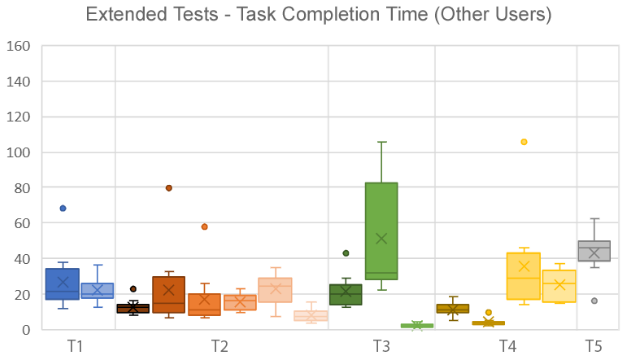


FIGURE 33. Box plot of the time taken by participants without a biological background to complete each of the interface tasks of the initial tests. Groups of tasks that are similar with those of the initial tests, in Figures 30, have matching colors.

In order to test the tools for analyzing temporal data, participants were first tasked with changing the "Data Mapping" to tendency (T4.1) and then apply the "Forces" layout in the time curve panel (T4.2). Given the initial complexity of the time curves model for new users, we encouraged participants explore the time curve panel during this time, including changing the color scheme or the sliders that control the parameters of the layout. Then they were tasked with observing the time curve to identifying time points with similar tendency shifts (T4.3) and time point that mark the largest variations in expression values (T4.4). The latter task could also be solved by looking at the timeline graph, which would highlight the time points where significant value shifts had occurred. The final task focused on the user interface, where participants were asked to create a new "Data Table" panel, then organize the workspace by moving and resizing panels (T5). As CroP allows panels to be moved and resized within a fixed grid, this task was meant to not only gauge the difficulties of managing the workspace, but also detect potential issues with CroP's automatic panel adjustments, such as overlap detection and resolution. In general, the only observed issues being the location of the option to create new panels and the initial learning curve of managing the panels, although either of these issues were minor or rare.

Task completion time is represented for those with a biological background in Figure 32, and those without in Figure 33. Overall, participants with a bioinformatics background took an average of 24% longer to resolve tasks than those with a visualization background, when excluding outliers. However, performing tests with this diverse group of users helped us detect and correct not only prominent usability problems, but also consider new actions. For instance, when interacting with a minimized section in the options sidebar, many of the users first tried to click the title of a section to open it before using the plus button. As this would allow users to more easily access or hide sections of the interface due to having a wider clickable area, we changed titles to also toggle sections open.

Participants were then asked to select how much they agreed with four affirmations from a 1 (Strongly Disagree) to 5 (Strongly Agree) scale, which related how easily they used CroP, whether the data visualizations were easy to interpret, whether they needed more time to learn how to use the tool, and if they could see themselves using the tool to analyze relational or temporal data. The distribution of answers is depicted in Figure 34. Participants were also provided with an open-ended question where they could write down any difficulties they felt when using the tool. In general, participants thought the tool was easy to use with an average score of 4.4, while the ease of interpreting the data representations got an average score of 3.8. These difficulties in interpreting some of the visualization models was also notable when participants were asked if they needed more time to learn how to use the tool, which got an average of 3.0. Based the obtained feedback, CroP was considered to be generally intuitive and easy to pick up despite the learning curve, having received positive interest in regards to potential future use with an average score of 4.4.

D. EXTENDED MODEL SURVEY

The model survey was performed immediately after the interface tests with the same participants, and served to not only continue our previous study on the efficacy of the time curves model, but also to obtain feedback on our new visualization models, namely the cluster glyphs. Regardless of their performance within the interface tests, participants were provided a quick introduction to the dataset, clustering, and time curve layout with an example. The first set of questions presented three time curves (Figure 35), each one accompanied with multiple options for behaviors where participants were asked to choose those that applied to each visualization. The first time curve represented the HIV-1 infection dataset and over 88% of the participants were able to associate the circular pattern to a cyclical tendency, and about 73% correctly interpreted these variations as being strong. However, 50% also discerned small variations between time points, but this may have been due to the close proximity of non-sequential

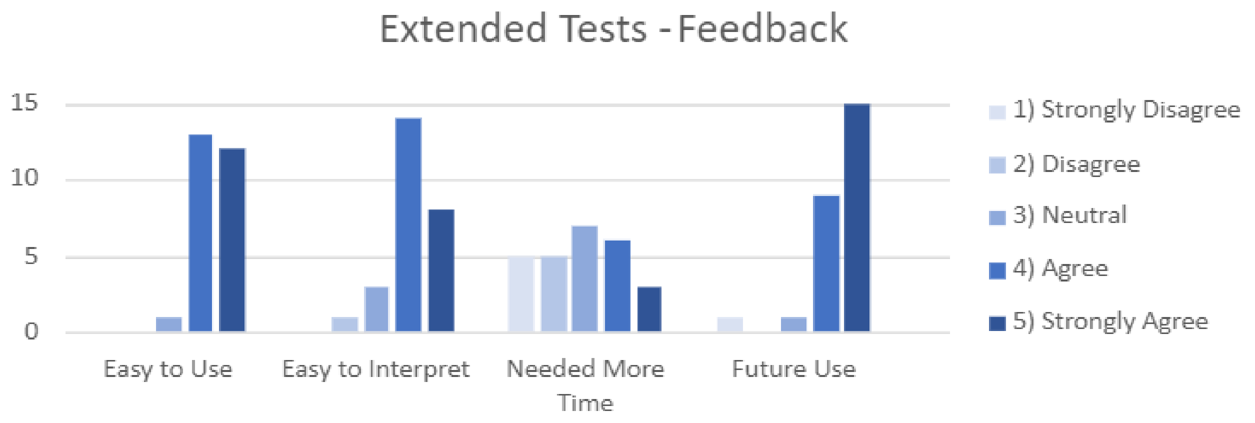


FIGURE 34. Score total given by participants for each of the affirmations in the feedback section of the extended tests.

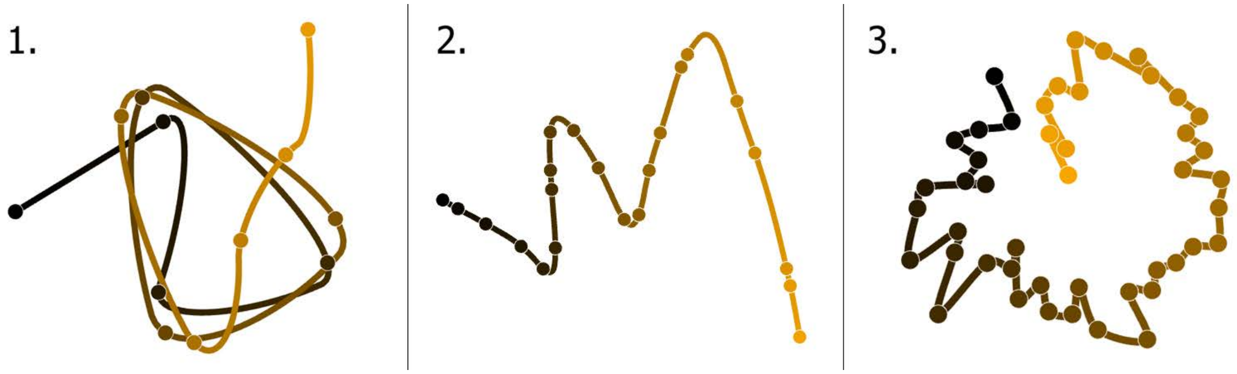


FIGURE 35. Three time curves shown to the participants of the third round of user testing, where they were asked to choose the behaviors that were represented in each visualization.

time points, according to feedback. The second time curve consisted of a single time-series without cycles but with inconsistent variations. About 23% of the participants interpreted these shifts in variation as a cycle, likely due to the time curve going up and down over time, and only 53% of the participants discerned the existence of both moments of strong and low variations of values. Finally, the third time curve depicted the malaria virus dataset which represents a single cycle with low expression variations. While 96% of participants were able to identify the small shifts in variation, only about 35% interpreted the overall shape of the time curve as representing a cycle. The distribution of these answers is displayed in Figure 36.

In order to obtain feedback on our cluster glyphs, we then presented participants with a network visualization being represented by three different glyphs (Figure 37). The first glyph is miniature representation of a clustered network, reducing each cluster into a circle whose size and color represents the average proprieties of that cluster. The second glyph abstracts the clustered network into a circular graph by converting each cluster into a slice, where its color represents the average values, its size represents the size of the cluster, and its relative position reflects the position the cluster in the network. The third glyph consists of a bar chart, where each bar represents a cluster in color and size, while its order reflects the horizontal position of clusters on the network. After being presented with the glyphs, participants were shown each one in a larger size and asked to rate the comprehensibility of the glyph, from 1 (Strongly Disagree) to 5 (Strongly Agree), first on reading value distribution and then whether they required more time to understand the

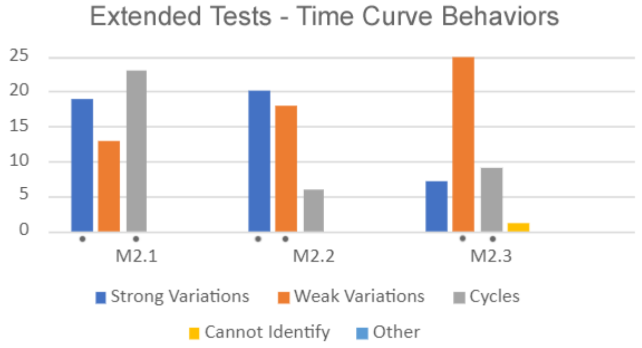


FIGURE 36. Total behaviors perceived by all participants for each of the time curves in the model questions of the extended tests; the most prominent behaviors that are exhibited by each of the curves are marked below their respective bar.

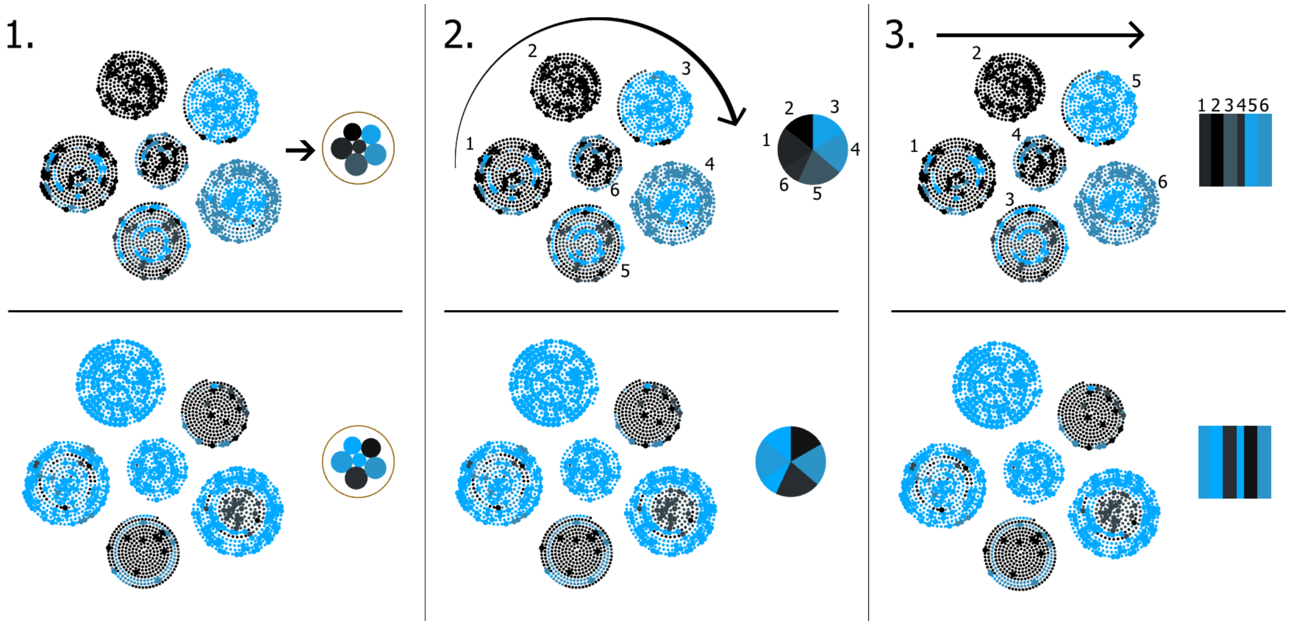


FIGURE 37. Three glyph representations of a network (1: miniature, 2: circular, 3: bars) at two different time points (top and bottom) used in the model tests performed during the third round of user testing.

glyph. Then, the glyphs were presented in a smaller size, which better portrayed how the glyphs are viewed on a time curve, and participants were asked to rate their readability, first on their legibility at a small size, and then on how easy multiple glyphs could be compared. When presented with each of the large glyphs, participants were provided an explanation of what each visual variable represented, and then they were shown statements regarding that glyph's comprehensibility. For the small glyphs, the statements reflected their legibility and how well they could be compared to each other. For each statement, participants could choose whether they agreed or disagreed using a scale from 1 (Strongly Disagree) to 5 (Strongly Agree) and the totals for the scores given are displayed in Figure 38. While results showed that the majority of participants agreed that the miniature network was the most intuitive glyph of those presented, the remaining scores did not show a clear preference, regardless of whether the glyph was presented in a small or large size. We can, however, note that the range of scores was higher for the evaluations of the bar graph glyph in comparison to the circular graph glyph.

Additionally, the test concluded with another open-ended question, inquiring participants on any particular difficulties, their preferred glyphs, and any additional feedback. Here, 54% of the participants expressed a preference for the miniature network as a large glyph, followed by 23% preferring the circular graph glyph, while the remaining either preferred the bar chart or had no preference. Regarding the glyphs in smaller size, the support for all three glyphs was once again balanced, although the bar chart was overall the least favorite among the participants that provided feedback.

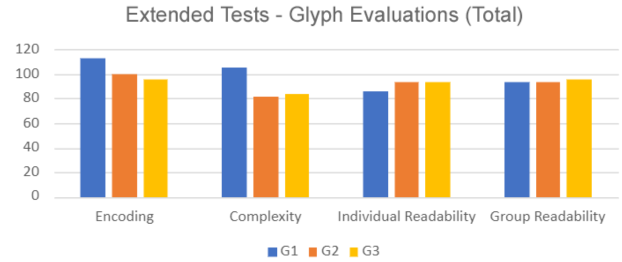


FIGURE 38. Sum of score values given by all participants for each of the attributes of each of the three glyphs (G1: miniature, G2: circular, G3: bars).

E. DISCUSSION

While in the experimentation chapter we were able to test the ability of CroP as a tool for representing and analyzing different types of data, it is only through user validation that we can evaluate its usability and the comprehensibility of the created visualizations. Our conclusions regarding the interpretation of the time curves visualizations were generally consistent throughout all the tests, where we must acknowledge the existence of a learning curve that is inherent to a model that creates abstracted data visualizations. For instance, in the preliminary survey, nearly half the participants misinterpreted the first time curve despite having correctly identified the cyclical behavior previously. However, over 80% of participants were able to identify the behavior that followed it, and most participants correctly answered all of the questions through the use of time curves, with half of these problems being answered correctly by over 90% of the participants.

In the final model tests, some participants also misinterpreted weak variations in the first time curve (Figure 35.1) and the existence of cycles in the second time curve (Figure 35.2), which may have been a result of trying to compare these visualizations to common linear graphs. Additionally, only a quarter of the participants identified a cycle in the third time curve (Figure 35.3), although a contributing factor may have been that the circle was not closed. However, despite these difficulties, even participants with little experience with data visualization were able to discern prominent behaviors, significant events and trends. As these visualizations were created from datasets containing thousands of data points, these results show that it is possible for CroP to create visualizations that comprehensibly represent complex datasets and promote the discovery of meaningful patterns. Regarding the cluster glyphs, we were able to take the following conclusions: while the miniature network was the easiest to comprehend for participants, it appeared to be more difficult to comprehend at a smaller size unless the graphical elements were enhanced; regarding the other glyphs, the bar chart glyph was considered to be easier to follow and compare due to its order of elements, while other participants preferred the circular chart glyph due to its simplicity and circular shape that matches the original nodes, unlike the former.

In what regards to testing CroP's usability, the interface tests that were performed contributed towards understanding how coordinated multiple views facilitate the exploration of multivariate datasets and whether visualization and data analysis approaches promote the discovery of meaningful relationships and patterns. As the initial interface tests were performed by a small group of participants with low experience with visualization tools and data analysis, there existed exceptional difficulties with concepts such as clustering. However, despite their inexperience, most users were able to navigate the tool and perform the indicated tasks, including loading data, brushing nodes, applying filters and analyzing data from their visual elements. These tasks were adjusted and expanded in the extended tests, further testing the ability of users to use CroP to analyze data, now with participants from various fields of study. While those with low experience with visualization tools had on average the longest task completion times, all of them were eventually able to solve all of the data analysis tasks involving the data panel and only 4 out of the 26 participants presented any significant difficulties with the tasks involving network clustering and the time curves. This showed how different types of users were able to utilize the available tools to identify elements or groups with specific proprieties, as well as analyze of one dataset across multiple visualization panels to discover different types of relationships.

Lastly, we can overview the results of the validation tests in relation to the considerations needed to be taken to accommodate users with varying levels of experience into CroP. Many of the preemptive measures that were taken with regard to usability were based on the fluid interaction principles, with particular regard to error prevention: failing to load a dataset will return an appropriate error message and list of the lines containing errors when appropriate; buttons and sliders are clearly labeled, and uncommon features are accompanied a help icon that describes the functionality; actions performed on either the visualization models or the user interface give immediate visual feedback. However, it was through validation with a wide variety of individuals that we were able to obtain new insight into the development of CroP and resolve issues that were not initially anticipated. This includes a revision of not only interface elements, but also interaction to be more intuitive, such as the addition of common keyboard shortcuts and data table selections. Additional visual feedback was added to certain hovered elements, including the addition of contextual information on the brushed data. Moreover, the addition of varied color palettes options was in response to feedback relative to some readability issues and concerns with accessibility to potential colorblind users.

VII. CONCLUSION

Interactive visualization can be a powerful tool in data analysis, providing the means to represent high-dimensional datasets comprehensibly and an environment to explore these datasets, discover new meaningful information and extract new knowledge. In this paper, we focused on the representation and analysis of temporal and relational data, in particular those from biological fields of study as they are often characterized as complex, due to their volume and high-dimensionality. It is in this context that we presented CroP, a new visualization tool that utilizes a coordinated multiple views framework to provide a modular environment where visualization panels can be placed and resized within a grid, building a workspace that best suits the dataset being analyzed. These panels can be used to visualize relational, temporal and multivariate datasets at different levels of detail, providing users with several types of layouts and tools to sort data points and variables in order to discover patterns of relationships.

Regarding the visualization of time-series in particular, we presented our implementation of the time curves layout and demonstrated its ability to represent different types of behaviors in time-series datasets. We complemented this model with Time Paths, a parameter-based layout that dynamically transforms time curve visualizations to represent temporal behaviors with varying levels of sensitivity to shifts in the data. By increasing the level of smoothing, the layout can not only reduce visual clutter but also promote the representation of predominant behaviors. Additionally, we can more easily control the visual proprieties of edges, which allows for smoother transitions between time points that more clearly represent the flow of time, including the creation of animated edges. The tool also features supporting visualization elements aimed at facilitating the identification of specific moments and behaviors in the timeline, particularly when dealing with complex time curve visualizations. Specifically, we implemented glyphs that represent the dataset at each stage, a lens-based area brush that can be used to search across groups of nodes and highlight those with similar proprieties, and the timeline graph, which supports the exploration of complex time curve visualizations while providing a simple visualization of the general shifts of data.

These functions and visualization models were demonstrated through the representation and analysis of multiple datasets with various levels of complexity. As discussed, the network and data table panels were able to present the datasets with different levels of detail, while providing the means to create groups of data points with similar proprieties and explore the composition of these groups. In this respect, the implemented layouts and clustering algorithms helped in understanding the structure of each dataset, highlighting the diversity of patterns of variables while isolating potential noise. Moreover, the models developed for the time curve panel were capable of representing different types of behaviors over time, across both single time-series and large datasets, highlighting periods of stagnation and cycles, as well as events that mark significant shifts of values. Using the glyphs, mouse lens and timeline graphs, we were able to dig-down into these patterns and identify nature of their respective behaviors and the nodes at their origin.

CroP and its visualization models were also subjected to multiple tests with a wide range of participants from different fields of study, involving the navigation of the tool, exploring a dataset and performing several tasks of varying complexity. As expected, participants with less experience with visualization tools presented longer times in completing tasks, particularly with those involving the identification of patterns in clusters and the time curve visualization. However, despite their limitations, most participants were able to complete most tasks without issues, showing the ability to navigate the tool's multiple views, identify data points and behaviors, as well as managing the workspace. In general, the tests showed that a majority of users were able to use CroP, regardless of their background and in spite of the amount of time spent with the tool, with feedback being overall positive in regards to the tool's usability and models. Moreover, feedback was employed in the tool's iterative development and we were able to solve most of the detected issues, including that of visualization tests served which was used in the development of glyphs, visual feedback and addition of color palettes.

REFERENCES

[1] S. Nusrat, T. Harbig, and N. Gehlenborg, "Tasks, techniques, and tools for genomic data visualization," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 781–805, 2019.

[2] N. Kerracher, J. Kennedy, and K. Chalmers, "The design space of temporal graph visualisation," in *Proc. EuroVis—Short Papers*, N. Elmqvist, M. Hlawitschka, and J. Kennedy, Eds. Swansea, U.K.: The Eurographics Association, 2014.

[3] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, and L. Yang, "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[4] J. F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, and N. Klitgord, "Towards a proteome-scale map of the human protein–protein interaction network," *Nature*, vol. 437, pp. 1173–1178, Sep. 2005.

[5] A. Nikitin, S. Egorov, N. Daraselia, and I. Mazo, "Pathway studio—The analysis and navigation of molecular networks," *Bioinformatics*, vol. 19, no. 16, pp. 2155–2157, Nov. 2003.

[6] S. Jain, J. Arrais, N. J. Venkatachari, V. Ayyavoo, and Z. Bar-Joseph, "Reconstructing the temporal progression of HIV-1 immune response pathways," *Bioinformatics*, vol. 32, no. 12, pp. 253–261, Jun. 2016.

[7] M. Baitaluk, M. Sedova, A. Ray, and A. Gupta, "BiologicalNetworks: Visualization and analysis tool for systems biology," *Nucleic Acids Res.*, vol. 34, pp. 466–471, Jul. 2006.

[8] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, "Time curves: Folding time to visualize patterns of temporal evolution in data," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 559–568, Jan. 2016.

[9] M. Secrier and R. Schneider, "Visualizing time-related data in biology, A review," *Briefings Bioinf.*, vol. 15, no. 5, pp. 771–782, Sep. 2014.

[10] E. D. Coelho, J. P. Arrais, and J. L. Oliveira, "From protein-protein interactions to rational drug design: Are computational methods up to the challenge?" *Current Topics Medicinal Chem.*, vol. 13, no. 5, pp. 602–618, Apr. 2013.

[11] A. Cruz, P. Machado, and J. P. Arrais, "CroP—Coordinated panel visualization for biological networks analysis," *Bioinformatics*, vol. 36, no. 4, pp. 1298–1299, 2010, doi: 10.1093/bioinformatics/btz688.

[12] A. Cruz, J. P. Arrais, and P. Machado, "Interactive network visualization of gene expression time-series data," in *Proc. 22nd Int. Conf. Inf. Visualisation (IV)*, Jul. 2018, pp. 574–580.

[13] A. Cruz, J. P. Arrais, and P. Machado, "Force-directed timelines: Visualizing & exploring temporal patterns," *Big Data Res.*, vol. 27, Feb. 2022, Art. no. 100291, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214579621001088

[14] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," *Queue*, vol. 8, no. 5, p. 20, May 2010, doi: 10.1145/1794514.1805128.

[15] A. Kerren, K. Kucher, Y.-F. Li, and F. Schreiber, "MDS-based visual survey of biological data visualization techniques," in *Proc. EuroVis*, A. P. Puig and T. Isenberg, Eds. Barcelona, Spain: The Eurographics Association, 2017, pp. 85–87.

[16] A. C. Greene, K. A. Giffin, C. S. Greene, and J. H. Moore, "Adapting bioinformatics curricula for big data," *Briefings Bioinf.*, vol. 17, no. 1, pp. 43–50, Jan. 2016, doi: 10.1093/bib/bbv018.

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, May 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[18] C. Meng, O. A. Zeleznik, G. G. Thallinger, B. Kuster, A. M. Gholami, and A. C. Culhane, "Dimension reduction techniques for the integrative analysis of multi-omics data," *Briefings Bioinf.*, vol. 17, no. 4, pp. 628–641, Jul. 2016, doi: 10.1093/bib/bbv108.

[19] J. Venna and S. Kaski, "Comparison of visualization methods for an atlas of gene expression data sets," *Inf. Visualizat.*, vol. 6, no. 2, pp. 139–154, Jan. 2007, doi: 10.1057/palgrave.ivs.9500153.

[20] L. Bijnens, P. Lewi, H. Goehlmann, G. Molenberghs, and L. Wouters, "Spectral map analysis-a method to analyze gene expression data," in *Proc. Amer. Stat. Assoc., Biopharmaceutical Sect.* Alexandria, Egypt: American Statistical Association, 2004, pp. 553–559.

[21] I. T. Jolliffe, *Principal Component Analysis and Factor Analysis*. Berlin, Germany: Springer, 1986, pp. 115–128.

[22] G. A. Pavlopoulos, D. Malliarakis, N. Papanikolaou, T. Theodosiou, A. J. Enright, and I. Iliopoulos, "Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future," *GigaScience*, vol. 4, no. 1, p. 38, Dec. 2015.

[23] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 267–279, Sep. 2014.

[24] S. Rani and G. Sikka, "Recent techniques of clustering of time series data: A survey," *Int. J. Comput. Appl.*, vol. 52, no. 15, pp. 1–9, Aug. 2012.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999, doi: 10.1145/331499.331504.

[26] A. Vogogias, J. Kennedy, D. Archambault, V. A. Smith, and H. Currant, "MlCut: Exploring multi-level cuts in dendrograms for biological data," in *Proc. Comput. Graph. Vis. Comput. Conf. (CGVC)*. Goslar, Germany: Eurographics Association, 2016, pp. 1–8, doi: 10.2312/cgvc.20161288.

[27] Qlucore. (2017). *Omics Explorer*. [Online]. Available: http://www.qlucore.com

[28] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, Nov. 1967, vol. 1, no. 233, pp. 281–297.

[29] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, ''Time-series clustering—A decade review,'' *Inf. Syst.*, vol. 53, pp. 16–38, Oct. 2015.

[30] M. S. G. Karypis, V. Kumar, and M. Steinbach, ''A comparison of document clustering techniques,'' in *Proc. TextMining Workshop KDD*, May 2000, pp. 428–439.

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, ''A density-based algorithm for discovering clusters in large spatial databases with noise,'' in *Proc. KDD*, vol. 96, Jan. 1996, pp. 226–231.

[32] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, ''OPTICS: Ordering points to identify the clustering structure,'' *ACM SIGMOD. Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999.

[33] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*. Redlands, CA, USA: Esri Press, 2010.

[34] L. Graham, ''Gestalt theory in interactive media design,'' *J. Humanities Social Sci.*, vol. 2, no. 1, pp. 1–12, 2008.

[35] C. Vehlow, F. Beck, and D. Weiskopf, ''The state of the art in visualizing group structures in graphs,'' in *Proc. Eurographics Conf. Vis. (EuroVis)—STARs*, R. Borgo, F. Ganovelli, and I. Viola, Eds. Cagliari, Italy: The Eurographics Association, 2015, pp. 21–40.

[36] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT, USA: Graphics Press, 1986.

[37] A. Dasgupta, M. Chen, and R. Kosara, ''Conceptualizing visual uncertainty in parallel coordinates,'' in *Computer Graphics Forum*, vol. 31. Hoboken, NJ, USA: Wiley, 2012, pp. 1015–1024.

[38] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, ''A taxonomy and survey of dynamic graph visualization,'' *Comput. Graph. Forum*, vol. 36, no. 1, pp. 133–159, Jan. 2017, doi: 10.1111/cgf.12791.

[39] C. Wang and J. Tao, ''Graphs in scientific visualization: A survey,'' *Comput. Graph. Forum*, vol. 36, no. 1, pp. 263–287, Jan. 2017, doi: 10.1111/cgf.12800.

[40] H. Chernoff, ''The use of faces to represent points in K-dimensional space graphically,'' *J. Amer. Statist. Assoc.*, vol. 68, no. 342, pp. 361–368, 1973, [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1973.10482434

[41] C. Dunne and B. Shneiderman, ''Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs,'' in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, Apr. 2013, pp. 3247–3256, doi: 10.1145/2470654.2466444.

[42] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, ''Reducing snapshots to points: A visual analytics approach to dynamic network exploration,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 1–10, Jan. 2016.

[43] Z. Hu, J. Mellor, J. Wu, T. Yamada, D. Holloway, and C. DeLisi, ''VisANT: Data-integrating visual framework for biological networks and modules,'' *Nucleic Acids Res.*, vol. 33, pp. 352–357, Jul. 2005.

[44] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg, ''AVOCADO: Visualization of workflow–derived data provenance for reproducible biomedical research,'' *Comput. Graph. Forum*, vol. 35, no. 3, pp. 481–490, Jun. 2016, doi: 10.1111/cgf.12924.

[45] J. Heinrich, C. Vehlow, F. Battke, G. Jäger, D. Weiskopf, and K. Nieselt, ''IHAT: Interactive hierarchical aggregation table for genetic association data,'' *BMC Bioinf.*, vol. 13, no. 8, pp. 1–12, Dec. 2012, doi: 10.1186/1471-2105-13-S8-S2.

[46] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, ''Guidelines for using multiple views in information visualization,'' in *Proc. Work. Conf. Adv. Vis. Interfaces*, New York, NY, USA, 2000, pp. 110–119, doi: 10.1145/345513.345271.

[47] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg, ''StratomeX: Visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization,'' *Comput. Graph. Forum*, vol. 31, pp. 1175–1184, Jun. 2012, doi: 10.1111/j.1467-8659.2012.03110.x.

[48] A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi, and H. Kitano, ''CellDesigner 3.5: A versatile modeling tool for biochemical networks,'' *Proc. IEEE*, vol. 96, no. 8, pp. 1254–1265, Aug. 2008.

[49] H. Hochheiser, E. H. Baehrecke, S. M. Mount, and B. Shneiderman, ''Dynamic querying for pattern identification in microarray and genomic data,'' in *Proc. Int. Conf. Multimedia Expo. (ICME)*, 2003, p. 453.

[50] H. Ding, C. Wang, K. Huang, and R. Machiraju, ''IGPSe: A visual analytic system for integrative genomic based cancer patient stratification,'' *BMC Bioinf.*, vol. 15, no. 1, p. 203, Dec. 2014, doi: 10.1186/1471-2105-15-203.

[51] M. Meyer, T. Munzner, and H. Pfister, ''MizBee: A multiscale synteny browser,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 897–904, Nov. 2009.

[52] J. Fulda, M. Brehmer, and T. Munzner, ''TimeLineCurator: Interactive authoring of visual timelines from unstructured text,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 300–309, Jan. 2016.

[53] J. Ernst and Z. Bar-Joseph, ''STEM: A tool for the analysis of short time series gene expression data,'' *BMC Bioinf.*, vol. 7, no. 1, p. 191, Dec. 2006, doi: 10.1186/1471-2105-7-191.

[54] A. Lex, H.-J. Schulz, M. Streit, C. Partl, and D. Schmalstieg, ''VisBricks: Multiform visualization of large, inhomogeneous data,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2291–2300, Dec. 2011.

[55] P. Craig, A. Cannon, R. Kukla, and J. Kennedy, ''MaTSE: The microarray time-series explorer,'' in *Proc. IEEE Symp. Biol. Data Vis.*, Oct. 2012, pp. 41–48.

[56] A. Gerasch, D. Faber, J. Küntzer, P. Niermann, O. Kohlbacher, H.-P. Lenhof, and M. Kaufmann, ''BiNA: A visual analytics tool for biological network data,'' *PLoS ONE*, vol. 9, no. 2, 2014, Art. no. e87397, doi: 10.1371/journal.pone.0087397.

[57] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, ''Visual comparison for information visualization,'' *Inf. Vis.*, vol. 10, no. 4, pp. 289–309, Oct. 2011, doi: 10.1177/1473871611416549.

[58] M. H. Shimabukuro, E. F. Flores, M. C. F. de Oliveira, and H. Levkowitz, ''Coordinated views to assist exploration of spatio-temporal data: A case study,'' in *Proc. 2nd Int. Conf. Coordinated Multiple Views Explor. Vis.*, 2004, pp. 107–117.

[59] M. Lawrence, E.-K. Lee, D. Cook, H. Hofmann, and E. Wurtele, ''ExploRase: Exploratory data analysis of systems biology data,'' in *Proc. 4th Int. Conf. Coordinated Multiple Views Explor. Vis.*, Jul. 2006, pp. 14–20.

[60] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister, ''Pathline: A tool for comparative functional genomics,'' *Comput. Graph. Forum*, vol. 29, no. 3, pp. 1043–1052, Aug. 2010, doi: 10.1111/j.1467-8659.2009.01710.x.

[61] M. Meyer, T. Munzner, A. DePace, and H. Pfister, ''MulteeSum: A tool for comparative spatial and temporal gene expression data,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 908–917, Nov. 2010.

[62] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid, ''Cerebral: Visualizing multiple experimental conditions on a graph with biological context,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1253–1260, Nov. 2008, doi: 10.1109/TVCG.2008.117.

[63] C. Reas and B. Fry, ''Processing: Programming for the media arts,'' *AI Soc.*, vol. 20, no. 4, pp. 526–538, Sep. 2006, doi: 10.1007/s00146-006-0050-9.

[64] M. Harrower and C. A. Brewer, ''ColorBrewer.Org: An online tool for selecting colour schemes for maps,'' *Cartographic J.*, vol. 40, no. 1, pp. 27–37, 2003, doi: 10.1179/000870403235002042.

[65] Y. Hu, ''Efficient, high-quality force-directed graph drawing,'' *Math. J.*, vol. 10, no. 1, pp. 37–71, 2005.

[66] M. R. Anderberg, *Cluster Analysis for Applications* (Monographs and Textbooks on Probability and Mathematical Statistics). New York, NY, USA: Academic Press, 1973.

[67] D. Müllner, ''Modern hierarchical, agglomerative clustering algorithms,'' 2011, *arXiv:1109.2378*.

[68] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, ''The SPMF open-source data mining library version 2,'' in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2016, pp. 36–40.

[69] J. Rychlewski, ''On hooke's law,'' *J. Appl. Math. Mech.*, vol. 48, no. 3, pp. 303–314, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021892884901370

[70] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. Berlin, Germany: Springer, 1986.

[71] J. Minichino. (2017). *Recurrent Neural Networks Course Project: Time Series Prediction and Text Generation*. Accessed: Dec. 2, 2019. [Online]. Available: https://github.com/techfort/aind2-rnn

[72] P. Mohammadi, S. Desfarges, I. Bartha, B. Joos, N. Zangger, M. Muñoz, H. F. Günthard, N. Beerenwinkel, A. Telenti, and A. Ciuffi, ''24 hours in the life of HIV-1 in a T cell line,'' *PLOS Pathogens*, vol. 9, no. 1, 2013, Art. no. e1003161, doi: 10.1371/journal.ppat.1003161.

[73] S. Anders and W. Huber, *Differential Expression of RNA-Seq Data at the Gene Level—The DESeq Package*, vol. 10. Heidelberg, Germany: European Molecular Biology Laboratory, 2012.

[74] Z. Bozdech, M. Llinás, B. L. Pulliam, E. D. Wong, J. Zhu, and J. L. DeRisi, ''The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum,'' *PLoS Biol.*, vol. 1, no. 1, p. e5, Aug. 2003.

[75] I. C. McDowell, D. Manandhar, C. M. Vockley, A. K. Schmid, T. E. Reddy, and B. E. Engelhardt, "Clustering gene expression time series data using an infinite Gaussian process mixture model," *PLOS Comput. Biol.*, vol. 14, no. 1, 2018, Art. no. e1005896, doi: 10.1371/journal.pcbi.1005896.

[76] C. Health and H. S. O. D. Portal. (2022). *Statewide COVID-19 Cases Deaths Tests*. accessed: Mar. 1, 2022. [Online]. Available: https://data.chhs.ca.gov/dataset/covid-19-time-series-metrics-by-county-and-state/resource/046cdd2b-31e5-4d34-9ed3-b48cdbc4be7a

[77] B. Fry, *Visualizing Data: Exploring and Explaining Data With the Processing Environment*. Sebastopol, CA, USA: O'Reilly Media, 2008.

ANTÓNIO CRUZ is currently pursuing the Ph.D. degree with the Faculty of Sciences and Technology, University of Coimbra, through the Doctoral Program for Information Science and Technology. His early visualization work was later published in IEEE COMPUTER GRAPHICS AND APPLICATIONS. During his M.Sc. degree, he has developed a visualization tool for genetic algorithms, which was published at *GECCO*. He is developing interactive visualization models directed at exploring large and complex datasets within the field of computational biology, including gene expression time-series, biological pathways, and protein-protein interaction networks. His research interests include graphic design and programming applied to information visualization, in particular the development of dynamic data visualization methods and tools.

JOEL P. ARRAIS received the M.Sc. and Ph.D. degrees in computer science from the University of Aveiro, in 2004 and 2010, respectively. He has been an Assistant Professor at the Department of Informatics Engineering, University of Coimbra, since 2012. In the recent past, he has been involved on several studies that focused on the analysis and knowledge extraction from gene-expression data. He has authored about 100 works, including journals and conference papers, including participation on reference conferences (ISMB/ECCB, BIBE, ICANN, and ESANN) and journals (*Bioinformatics* (Oxford), IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and *Journal of Cheminformatics*). He has been involved in several national and FP7/HORIZON2020 European funded projects. His particular research interests include using pattern recognition and machine learning methods applied to biological problems.

PENOUSAL MACHADO is currently an Associate Professor at the Department of Informatics Engineering, University of Coimbra. He is a Coordinator of the Cognitive and Media Systems Group, CISUC. He is the author of more than 200 refereed journals and conference papers. His research interests include artificial intelligence, evolutionary computation, computational creativity, and information visualization. He is the President of SPECIES—the Society for the Promotion of Evolutionary Computation in Europe and its Surroundings. He is a member of the Steering Committee of EuroGP, EvoMUSART, and Evostar. He is a member of the Executive Board of SPECIES. He was a recipient of several scientific awards, including the prestigious EvoStar Award for Outstanding Contribution to Evolutionary Computation in Europe. His publications have been awarded as best papers multiple times. He has been a keynote speaker at several major international conferences. His work was presented in venues, such as the National Museum of Contemporary Art and MoMA, NY, USA. He has chaired major events, including ICCC, PPSN, EvoStar, EuroGP, and EvoMusart.