Exploring Time-Series Through Force-Directed Timelines

António Cruz
University of Coimbra, CISUC,
Department of Informatics Engineering
Coimbra, Portugal
Email: antonioc@dei.uc.pt

Joel P. Arrais
University of Coimbra, CISUC,
Department of Informatics Engineering
Coimbra, Portugal
Email: jpa@dei.uc.pt

Penousal Machado
University of Coimbra, CISUC,
Department of Informatics Engineering
Coimbra, Portugal
Email: machado@dei.uc.pt

***Abstract*—Temporal datasets are a product of many scientific disciplines and analyzing the events that they describe may help provide valuable insight into their respective research subjects and help move towards solutions to existing problems. Time-series analysis is still an open problem which prompts new solutions, particularly the discovery of patterns across complex temporal networks. Visualization has proven to be a valuable tool in the analysis of such datasets, with the emergence of new models such as Time Curves, which distorts timelines to position time points based on their similarity, creating visualizations that highlight behavior patterns. In this paper, we further explore time-series functionally and aesthetically by revising the dynamic Time Curves models in CroP, a visualization tool with coordinated multiple views. Firstly, we propose the additional of new visual elements and interactive functions, coordinated with a network visualization to help discover and understand temporal patterns across complex datasets. Secondly, we visually explore time-series through Time Paths, a parameter-based force-directed layout that can dynamically transform the original model to either highlight small data variations or reduce visual noise in favor of overall patterns.**

***Keywords*-Data visualization, Time series analysis, Interactive systems**

I. Introduction

The analysis of temporal data has long been a topic of interest in the field of data analysis, as multiple disciplines must contend with the study of subjects which contain a temporal component, such as the evolution of physical organisms or the observation of processes affecting single entities or complex networks [1], [2]. Visualization can be a powerful tool in data analysis, providing users with the means to navigate, brush and filter information through simple representations, or even provide new ways to visualize the data that highlight patterns and significant moments [3], [4]. Through this, it may be possible to obtain a deeper understanding of the source of the observed behaviors, as well as gain the ability to predict future events in the data.

In this paper, we propose new visual and interactive approaches to explore and analyze time-series based on the Time Curves layout [5], where timelines are bent using multi-dimensional scaling to position time points relatively to their similarity. These functionalities were implemented into CroP, an interactive data visualization tool capable of representing network and time-series datasets through multiple coordinated views [6].

Our first contribution is a revision of CroP's Time Curve model, providing users with additional tools to dig-down and analyze temporal behaviors and patterns. Specifically, we present a lens-based approach that allows users to brush across groups of time points displaying general time patterns in order to identify the and analyze the data points at their source. This analysis is assisted through the addition of glyphs, which are able to represent instances of complex datasets at each time point, and through coordination with other views.

As our second contribution, we introduce Time Paths, a force-directed layout that smoothens timelines according to variable parameters, creating visualizations that can either be more or less sensitive to variations in the data. While increasing the level of smoothing further increases data abstraction, this can be used to not only reduce visual clutter and promote the representation of overall behaviors, but also highlight significant events or outliers. Additionally, Time Paths allow for additional control over the visual proprieties of edges, which we explore by animating the flow of time.

II. Related Work

A. Time Representation

Timelines have been used to describe sequences of events across many disciplines for centuries, having had multiple types of representations to help portray and analyze time-series within various contexts [7]. M. Brehmer et al. surveyed existing timeline visualizations in the context of storytelling, in the sense of using these as a means to portray information coherently and in such a way that it engages the viewer [8], where linear representations were found to be the most common, mapping the evolution of values across one axis representing the flow of time.

While timelines can easily encode a small number of variables through shape, size and color, this may not be feasible in accurately representing complex systems changing over time, such as large networks where each time point represents a different state for each data point. While animation can be employed as a natural way to convey changes over time, where each state of the entire system can

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00061

be represented in successive frames, it is limited by human perception capabilities, as people are more likely to only focus on significant changes [9]. Alternatively, complex glyphs or small multiples can be used to show a representation of the state of the data for each time point [10], while three-dimensional graphs can map thousands of time steps [11]. However, showing every state simultaneously may significantly increase the complexity of the visualization and the excess of information could overwhelm viewers.

B. Time Curves

To manage visual complexity, visualizations may seek to reduce the size of the information space, condensing visualizations, or even aggregating groups into singular visual elements. This often entails a loss of information in favor of reducing visual noise and highlighting general patterns. Bach et al. presented Time Curves [5], a visualization model which utilizes multidimensional scaling to position time points in low-dimensional space in such a way that their relative distance reflects the similarity between their attributes. This is achieved by calculating the similarity between time points using data-specific metrics and then applying a force-directed layout on the timeline to attract time points based on their similarity, resulting in a bent timeline whose shape reflects the behaviors of the data, such as significant events, cyclical patterns, regressions and outliers.

Elzen et al. [12] presented a similar concept which further showed how this layout is able to represent the overall behaviors of complex systems over time. In their work, the properties of a network at each point in time are abstracted into a point on a two-dimensional plane to build a Time Curve, which will then portray the structural changes in the network over time. Through the Time Curve it is possible to identify the periods in which the network remained with a specific structure, the moments when this structure changed, the intensity of these changes, and when the network returned to similar, previous structures.

C. Interactive Visualizations

Visualization tools can employ interactive functions that provide users with the ability to navigate and filter the data, giving them control over the amount of information displayed on screen. Regarding navigation, visualization tools can employ a details-on-demand approach by allowing users control over the timeline to switch between time points and visualize additional information through supporting visualizations [13], [14]. Switching between levels of detail can also be achieved through semantic zooming, where the amount of information displayed regarding each time point increases as the user zooms in on a section of the timeline [15]. Alternatively, analyzing different levels of detail simultaneously has also been explored. VAST [16] is web-based application that analyzes spatiotemporal datasets by providing a high-level overview of the underlying structure of the data at each level of detail, allowing users to identify patterns across multiple levels of detail.

Filtering or highlighting information can be achieved through queries. These can be performed indirectly through user interface elements, such as search bars to find specific values and sliders that set threshold values to filter less relevant data, or directly through brushing, where data is selected by interacting with its graphical representations. For instance, TimeSearcher [17] and MaTSE [18] allow users to select sections of time series data visualizations in order to find temporal patterns that are similar to the one within the selected area. Additionally, lens-based approaches can be used as both a semantic zooming tool and as area brushes, as they are controlled by users and can change the elements and information located within their radius [19].

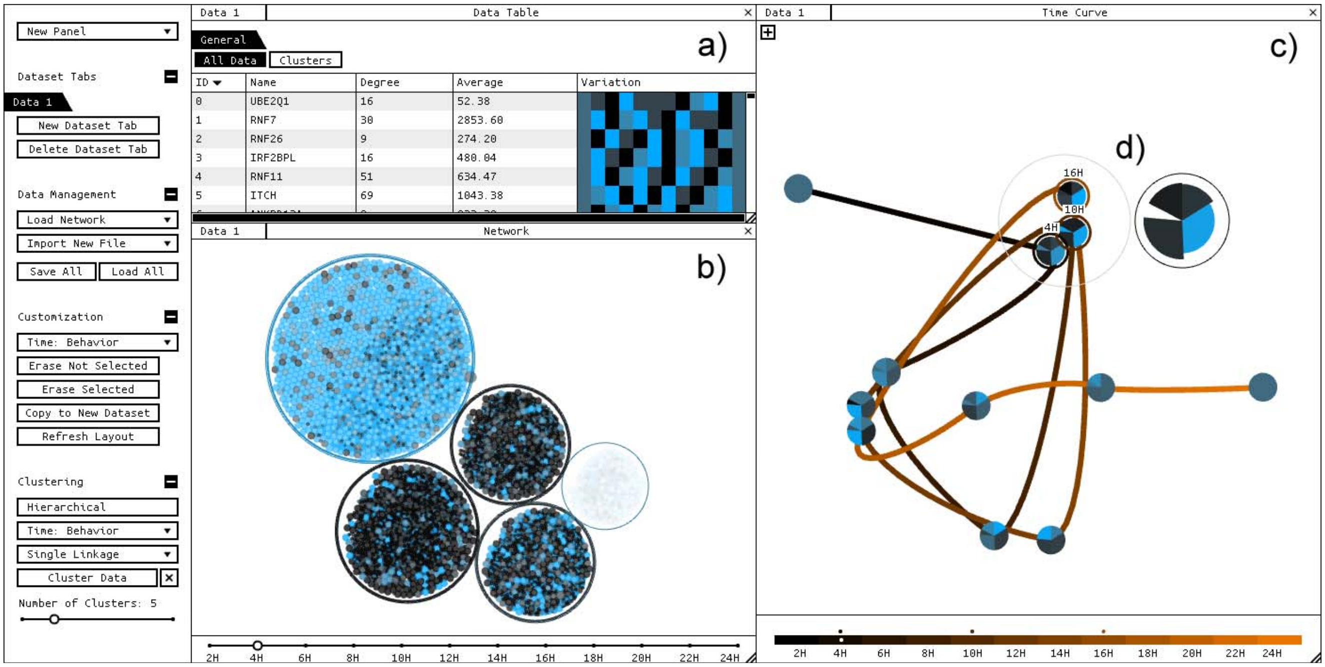
III. Framework

The methods presented in this paper were integrated into CroP [6], a visualization tool which employs a multiple coordinated views layout to visualize user-provided datasets at different levels of detail. It provides multiple visualization models, including relational networks, tabular visualizations, linear graphs, and an implementation of time curves. These are contained within panels that users can move and resize to tailor the work environment to different problems (Figure 1).

While CroP was designed for the analysis of biological datasets, it is also able to process generic relational and temporal datasets. Loaded networks can be integrated with time-series data and then visualized in the network panel (Figure 1.b), which allows users to explore how they change over time through an interactive timeline slider. Time-series data can also be clustered to create groups of nodes that present similar temporal patterns. This is achieved through a hierarchical clustering algorithm [20], which allows users to set and dynamically switch between the number of clusters.

Data can either be clustered by values at each time point, by the variation of these values between time points, or by behaviors, a simplistic approach which simply looks at whether values are increasing, decreasing, and when these tendencies change. The most appropriate approach will depend on the data and the type of relationships being studied. These properties are portrayed using color, mapping increasing values and variation from black to blue. When color is used to represent behaviors, shades of blue represent increasing values, with the strongest blue representing value peaks, and dark tones represent value decreases, with black showing valleys.

In this paper, our focus is on the Time Curve panel (Figure 1.c), which allows users to discover the relationship between time points through a visualization model based on the layout presented by Bach et al. [5]. The following sections describe our revisions to the model and the imple-



ID ▾	Name	Degree	Average	Variation
0	UBE2Q1	16	52.38	
1	RNF7	30	2853.60	
2	RNF26	9	274.20	
3	IRF2BPL	16	480.04	
4	RNF11	51	634.47	
5	ITCH	69	1043.38	

Number of Clusters: 5

Figure 1. Screenshot of CroP, representing a dataset through its multiple views: a data table (a) listing every data point, a network panel (b) showing the data clustered into five groups, and a Time Curve panel (c), where the timeline has been bent to show groups of time points. One of these groups is being brushed with the mouse lens (d), creating a pie chart that reveals the data clusters that present the most similar behaviors during the three selected temporal instances, and also highlighting the corresponding clusters in the network visualization.

mented functionalities for visualizing and exploring time-series.

IV. Time Curve Panel

When temporal data is loaded into CroP, each time point is converted into a node and displayed sequentially as a timeline in the Time Curve panel, represented either as a straight horizontal line or as a spiral, the latter being used when the length of the former surpasses the width of the window due to a large quantity of time points. The default color scheme represents temporal progression from the initial time point to the last using a gradient from black to orange. Alternatively, the color can also portray similarity, where nodes that are spatially closer will also be closer chromatically. In regard to navigation, the mouse can be used to pan, zoom and select time points. Furthermore, on the bottom of the panel there is a slider that can switch between time points while highlighting them in the top visualization. Moving the slider will hide every edge except those between the time points that the slider crossed. This results in an animated transition that portrays the sequence of time without the need for additional visual elements.

A. Bending Time

After the initial layout is loaded, the timeline can be distorted through a force-directed layout comprised of springs between every time node. The formula used to calculate the force applied to each spring is based on Hooke's law [21] and each spring's ideal stretching length is determined using a similarity matrix, which was updated in regard to the previous implementation. The calculation of the similarity between two time points is now determined using mean of the difference between their values and the difference between their variation. We define variation as the difference between the current value and that of the previous and next time points. As such, the position of time points is based not only on their current values, but also depends on how they vary between each time step.

As a baseline, this model was tested using datasets containing single time-series that depicted consistent behaviors. For these experiments, we chose a time-series of 500 time points depicting a sine wave and altered it to depict various transformations over time. The resulting visualizations are shown in Figure 2.

The visualization of the basic sine wave dataset resulted in an oval consisting of overlapping loops, whose shape represents the shifts in variation over time, while the number of loops matches the number of cycles in the dataset (Figure 2.a). Regarding the transformations applied, changing the minimum and maximum values changed the shape's height, which matches the changes to the amplitude of each cycle (Figure 2.b). This was followed by an overall consistent increase to the values, which resulted in a matching gradual position shift for the ovals representing each cycle (Figure 2.c). Finally, increasing frequency resulted in stronger value variations. These were represented with increases to the width of the cycle's loops, making the larger variations also more visually noticeable (Figure 2.d).

The model also continues to be able to handle multiple time-series, where for each time point we calculate the average similarity with every other data point. For instance, CroP and Time Curves were presented in a previous paper to represent a gene expression time-series RNA-Seq dataset which described the fluctuations of expression across a network of proteins in reaction to the HIV-1 infection over a period of 24 hours [6]. Bending the timeline revealed cyclical patterns, where the same groups of proteins presented similar increases or decreases at different points in time. As we tested the same dataset on the current model, we were able to visualize the same groups displaying the same temporal patterns, as shown in Figure 1.

B. *Analyzing Temporal Patterns*

A limitation of the Time Curves layout is that while they can show the existence of general behavior patterns, they provide no means to further understand their nature without additional methods. As such, we are interested in updating the model to not only portray the fluctuations that originate existing patterns, but also provide the means to dig-down and discern which data points are responsible for these behaviors.

Our first challenge was to provide graphical representations of the dataset at each time point, to better understand the similarities and differences between the instances positioned by Time Curves without having to rely on additional views. Scalability is a consistent issue when dealing with large datasets, as it may not be possible to properly portray hundreds of thousands of data points at each temporal instance through a simple glyph. To this end, we instead chose to portray data clusters, as each one can represent a consistent group of data points possessing similar temporal profiles. As such, whenever the current dataset is clustered, the visualization in the Time Curve panel is updated by replacing the time nodes with pie chart glyphs (Figure 1.c). Each slice represents one cluster, where the width of its arc represents the number of nodes in the cluster and its color corresponds to the average properties of every node in the group at its corresponding time point. The width and order of the slices is consistent across every glyph to facilitate their comparison. This provides users with some perspective over the amount of different temporal profiles within the dataset and how their fluctuations relate with each other over time.

To further explore the relationships between each cluster over time, we also introduced a lens-based approach that provides an on-demand visualization to compare each time step within a section of the screen. Right-clicking over the Time Curve panel will create a circular area that follows the mouse, allowing users to brush over multiple time nodes simultaneously, which creates a pie chart visualization next to the lens (Figure 1.d). This pie chart represents the average of the time points selected, representing the same clusters and maintaining the width and order of those used as glyphs, where the colors of each slice now represents the average of those selected. More importantly, the radius of each slice is now variable, mapped to the similarity of each cluster between the selected time points. The similarity between the clusters is determined by calculating the standard deviation between their properties. As such, slices with a wide radius represent clusters whose behaviors are consistent across the selected time points, indicating that the data points in those clusters may be responsible for the initial patterns portrayed by the time curves.

These selections are coordinated with the network panel, where the transparency of each cluster is mapped to their

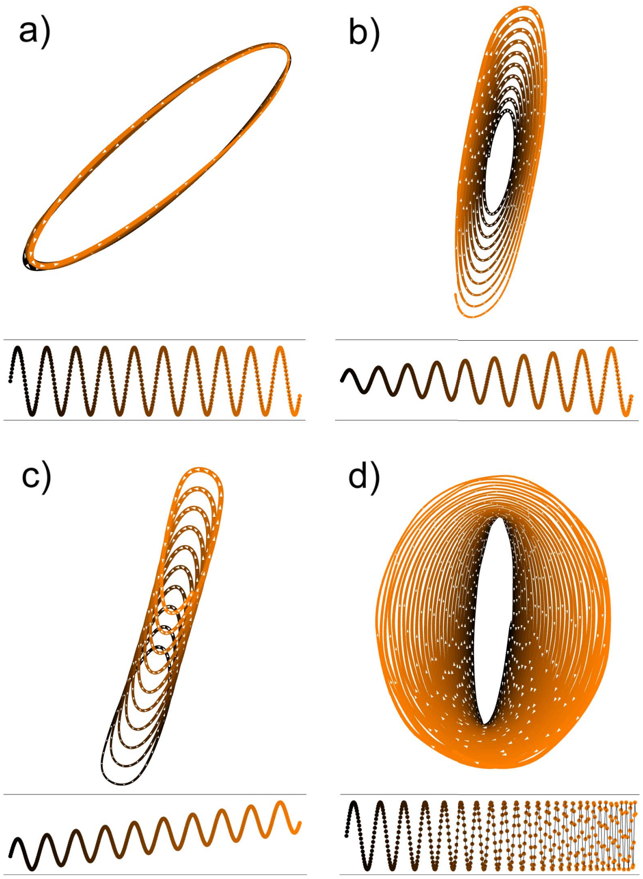


Figure 2. Time curve visualizations of four datasets, with their corresponding line charts below each one. The first dataset describes a sine wave (a) and then three gradual transformations: an increase of its amplitude (b), an overall increase of values (c), and an increase of its frequency (d).

similarity between the selected time points, making it easier to identify significant clusters and isolate them through filtering (Figure 1.b). Additionally, the scroll wheel can be used to increase or decrease the size of the lens, and by holding a key on the keyboard, nodes will remain selected even when no longer within the lens, allowing users to quickly brush through multiple distinct areas of the Time Curve in order to compare specific time points between them.

When analyzing the previously described HIV-1 dataset through these new tools, it was possible to quickly identify the groups of nodes that were presenting similar behaviors at the same points in time. For instance, when using the lens on the group of time points representing 4, 10 and 16 hours, we can see from both the pie chart and the network that most nodes present the same behaviors at these three instances, as shown in Figure 1. In particular, we can discern a group of nodes whose expression peaks at these time points, and three other clusters that show nodes presenting either valleys of expression or decreasing value tendencies. Furthermore, we can also easily identify the one small cluster of nodes which has no part in this particular pattern which of expression. The new methods make the identification of these nodes and their behaviors quicker, allowing users to select and isolate them so they can be studied further.

V. Time Paths

Time Paths is a layout that smoothens an existing Time Curve visualization by redrawing it with a brush controlled by parameter-based attraction forces. This layout was created visually explore Time Curves, giving users the option to dynamically transform the original visualization to either reduce visual noise in favor of portraying overall behavior patterns, or emphasize variations in the data through abstraction.

A. *Force-Directed Drawing*

The brush that draws the Time Path consists of a moving point which is first placed at the initial time point on the original Time Curve, and it is then pulled towards the following time point using a spring, calculated once again using Hooke's law and a fixed attraction strength. As the brush is subjected to this attraction force, intermediate points are left behind which map the brush's route. After a set number of intermediate points have been placed, the brush is then pulled towards the next time point, in sequential order. However, the new spring does not immediately replace the previous one, as momentum is applied. In this instance, we define momentum as a percentage value that determines how quickly the attraction force to the previous time point dissipates and the attraction force to the next point takes its place. The brush continues moving across time points sequentially until the final one, creating a new curve through the intermediate points left in its path.

Through this, every edge becomes defined by a set of points which provides us with increased control over their visual representation, as color and size can be changed for each segment. This allowed us to not only add gradual transitions between the color and size of two time nodes, but also introduce new elements, such as drawing arrows across edges to convey direction. Additionally, we also added the option to view the arrow particles moving across the edges in their respective directions. The speed of the particles adapts to the properties of each edge, increasing when edges are far apart, thus conveying the intensity of the changes between time points through movement.

B. *Parameterization*

We can define two main variables that control the brush's trajectory: the number of intermediate points, and momentum. These are parameters that can be changed through sliders in order to dynamically create new visualizations with different levels of sensitivity to the original Time Curves. Decreasing the momentum will result in the brush converging towards the next time point faster, while increasing it will cause forces to dissipate slowly and create wider loops. Meanwhile, the number of intermediate points between each time point controls the detail of each curve and their transitions.

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. Additionally, overlapping intermediate points will be removed in order to clean the resulting artifact and hasten its drawing speed.

C. *Experimentation*

To test the Time Paths layout, we chose two single time-series gathered from real events. These datasets were chosen because they describe two cyclical behaviors with different trends and minor variations throughout, allowing us to not only test how the different parameters affect the resulting visualization, but also demonstrate the ability of the model to smoothen existing curves and reduce visual clutter, while highlighting overall patterns and trends.

The first dataset describes "Wolfer's Sunspot Numbers", a yearly measurement of sunspots, small dark areas caused by concentrations of the magnetic field flux on the sun's surface, from 1770 to 1869 [22]. This time-series contains 100 time points and presents cycles with similar minimums but varying peak values, resulting from periodic value increases of different intensities.

The second dataset depicts "Monthly Milk Production", measuring pounds per cow from January 1962 to December 1975 [23]. The dataset contains 168 time points and is characterized by a yearly production cycle with minor jumps in variation and a consistent increasing trend until the final five years, which present a stabilization in production.

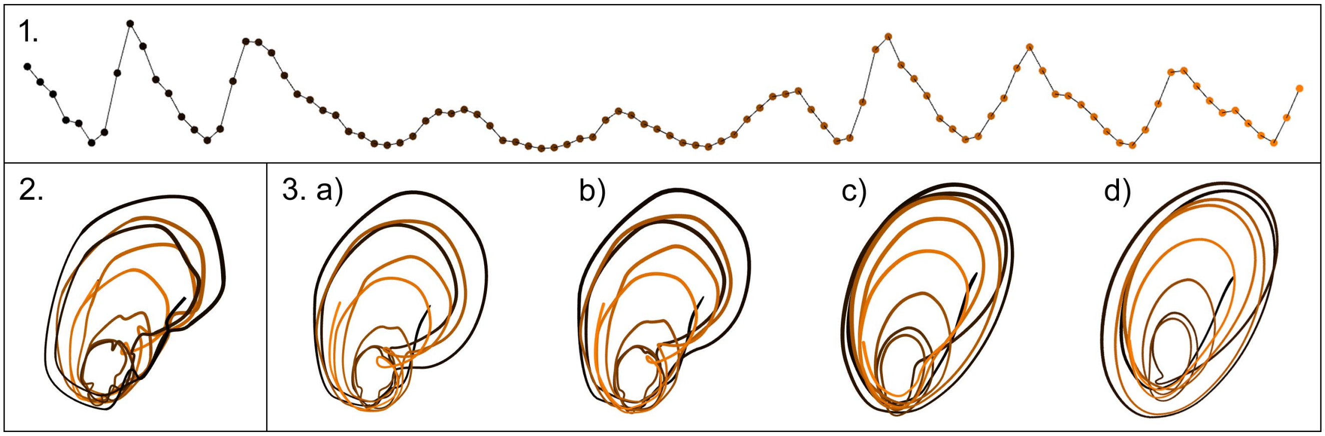


Figure 3. Visualizations of the ”Wolfer’s Sunspot Numbers” time-series, depicted as a line chart (1.), a Time Curve (2.), and then Time Paths created using different parameters (3.) where the level of smoothening is increased (from a) to d)).

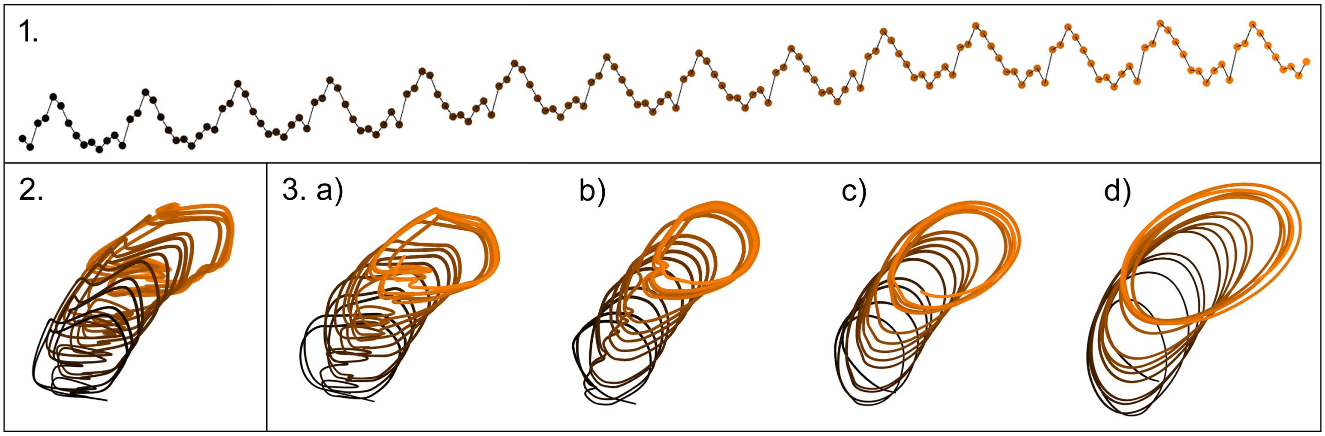


Figure 4. Visualizations of the ”Monthly Milk Production” time-series, depicted as a line chart (1.), a Time Curve (2.), and then Time Paths created using different parameters (3.) where the level of smoothening is increased (from a) to d)).

In both cases, the Time Curve depicted each cycle with circular patterns whose sizes and positions matched the respective variations and trends observable in the dataset, which is consistent with previous results (Figure 2). The Time Path layout was then applied with four different sets of parameters, increasing the level of smoothening with each one. The results are depicted in Figures 3 & 4.

The first set of parameters (a) is the default, resulting in a general smoothening of the time curve, but maintaining sensitivity to smaller variations. More notably in Figure 3.a, short variation spikes are drawn as loops, illustrating and highlighting these instances. The following two parameter sets feature an increased level of abstraction, caused first by a reduction in the number of intermediate points (b), which reduces the path’s detail, and then an additional increase in momentum, which results in a slower change in trajectory for the brush between time points. The third set in both tests appears to have removed almost all visual clutter, representing only the overall cycles through loops. Finally, the last parameter set significantly increases momentum, resulting in exaggerated depictions of their behaviors, in particular those featuring significant changes, such as the largest value increases in the final five cycles of sunspots dataset (Figure 3.d). In general, Time Paths were able to smoothen the original visualizations, reducing visual perturbances in favor of depicting the main cyclical behaviors, particularly in the Sunspot dataset.

VI. Conclusions and Future Work

In this paper, we presented new methods to further discover and interpret the behavior patterns represented by the Time Curves layout. We revised our implementation of this layout in an existing visualization framework and then demonstrated its ability to represent different types of temporal behaviors. The first methods seek to promote the analysis of time patterns found across large volumes of data points. Namely, we added glyph representations that allow viewers to discern differences between the data at different time points without the need for additional views. Furthermore, we implemented a lens-based area brush that can show which nodes present the most similar behaviors between any group of time points. These methods helped quickly discover the groups of proteins responsible for cyclical patterns from the HIV-1 infection which had been previously explored.

While these methods serve to analyze temporal patterns by digging-down, we also explore temporal patterns visually through Time Paths, a parameter-based layout that dynamically transforms Time Curve visualizations to represent the original behaviors with different levels of detail or abstraction. Through the experiments performed, the model showed to be able to not only represent different types of behaviors over time, but also smoothen layouts to reduce visual clutter and highlight overall trends. Additionally, Time Paths gives additional control over the representation of edges, which allowed for the creation of animated visual elements that convey direction and variation more easily. These elements combined with interaction, such as brushing the timeline, may allow users to more easily understand sequences of events and pinpoint important moments.

Regarding future work, the Time Paths layout presents some limitations with very large time-series and should be visually explored further, particularly in regard to edge representation and overlap. While the application was able to handle the representation of complex datasets with thousands of time points with minor drops in performance, it was not possible to completely eliminate visual noise. In this regard, we intend to integrate techniques such as node aggregation and edge-bundling to further promote the portrayal of general behaviors and patterns. Additionally, we should consider a more focused range of ideal parameters, as well as autonomously suggest initial parameters for each dataset based on their properties. Automation should also be considered in temporal cluster analysis to reduce the need for brushing, such as detecting temporal patterns and highlighting nodes and clusters based on their impact in these patterns.

Acknowledgment

This work is funded by national funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project CISUC—UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER029266). The first author is funded by the Portuguese Foundation for Science and Technology (FCT), under the grant SFRH/BD/124538/2016.

References

[1] S. Nusrat, T. A. Harbig, and N. Gehlenborg, "Tasks, techniques, and tools for genomic data visualization," in *Comput. Graph. Forum*, 2019.

[2] N. Kerracher, J. Kennedy, and K. Chalmers, "The Design Space of Temporal Graph Visualisation," in *EuroVis - Short Papers*, N. Elmqvist, M. Hlawitschka, and J. Kennedy, Eds. The Eurographics Association, 2014.

[3] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[4] S. I. O'Donoghue, B. F. Baldi, S. J. Clark, A. E. Darling, J. M. Hogan, S. Kaur, L. Maier-Hein, D. J. McCarthy, W. J. Moore, E. Stenau, J. R. Swedlow, J. Vuong, and J. B. Procter, "Visualization of biomedical data," *Annual Review of Biomedical Data Science*, vol. 1, no. 1, pp. 275–304, 2018. [Online]. Available: https://doi.org/10.1146/annurev-biodatasci-080917-013424

[5] 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 Transactions on Visualization and Computer Graphics*, vol. 22, pp. 1–1, 10 2015.

[6] A. Cruz, P. Machado, and J. P. Arrais, "CroP—Coordinated Panel visualization for biological networks analysis," *Bioinformatics*, 09 2019, btz688. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz688

[7] D. Rosenberg and A. Grafton, *Cartographies of time: A history of the timeline*. Princeton Architectural Press, 2013.

[8] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner, "Timelines revisited: A design space and considerations for expressive storytelling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2151–2164, Sep. 2017.

[9] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner, "Visual analysis of large graphs: State-of-the-art and future research challenges," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1719–1749, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01898.x

[10] A. Lex, H.-J. Schulz, M. Streit, C. Partl, and D. Schmalstieg, "Visbricks: Multiform visualization of large, inhomogeneous data," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2291–2300, 2011.

[11] V. Bruder, M. Hlawatsch, S. Frey, M. Burch, D. Weiskopf, and T. Ertl, "Volume-based large dynamic graph analytics," in *2018 22nd International Conference Information Visualisation (IV)*, 2018, pp. 210–219.

[12] 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 Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 1–10, Jan 2016.

[13] C. Niederer, H. Stitz, R. Hourieh, F. Grassinger, W. Aigner, and M. Streit, “Taco: Visualizing changes in tables over time,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 677–686, Jan 2018.

[14] B. Bach, E. Pietriga, and J. Fekete, “Graphdiaries: Animated transitions and temporal navigation for dynamic networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 5, pp. 740–754, May 2014.

[15] D. A. G. Aguilar, R. Therón, and F. J. García-Peñalvo, “Semantic spiral timelines used as support for e-learning,” *Journal of Universal Computer Science*, vol. 15, no. 7, pp. 1526–1545, 2009.

[16] R. A. Silva, J. M. Pires, N. Datia, M. Y. Santos, B. Martins, and F. Birra, “Visualising hidden spatiotemporal patterns at multiple levels of detail,” in *2018 22nd International Conference Information Visualisation (IV)*, 2018, pp. 294–302.

[17] H. Hochheiser, E. H. Baehrecke, S. M. Mount, and B. Shneiderman, “Dynamic querying for pattern identification in microarray and genomic data,” in *2003 International Conference on Multimedia and Expo. ICME ’03. Proceedings (Cat. No.03TH8698)*, vol. 3, July 2003, pp. III–453.

[18] P. Craig, A. Cannon, R. Kukla, and J. Kennedy, “Matse: The microarray time-series explorer,” in *2012 IEEE Symposium on Biological Data Visualization (BioVis)*, Oct 2012, pp. 41–48.

[19] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann, “Interactive lenses for visualization: An extended survey,” *Computer Graphics Forum*, vol. 36, no. 6, pp. 173–200, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12871

[20] 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 Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.

[21] J. Rychlewski, “On hooke’s law,” *Journal of Applied Mathematics and Mechanics*, vol. 48, no. 3, pp. 303 – 314, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021892884901370

[22] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. Berlin, Heidelberg: Springer-Verlag, 1986.

[23] J. Minichino, “Recurrent neural networks course project: time series prediction and text generation,” 2017, accessed: 2019-12-02. [Online]. Available: https://github.com/techfort/aind2-rnn