TempNetViz: a GUI for temporal graphs visualization

# Summary

Temporal network analysis is an emerging and rapidly expanding research area concerned with the study of how graphs evolve over time. Graphs provide a natural representation of complex systems and have therefore become indispensable across a wide range of scientific domains, including neuroscience, biology, sociology, and natural language processing. While traditional network analysis has focused primarily on static structures, most real-world systems are inherently dynamic: their connectivity patterns change, sometimes subtly and sometimes dramatically, over time. As a result, understanding both the structural properties of a network and their temporal evolution is essential for uncovering the mechanisms that drive system behavior.

Temporal network analysis addresses this need by jointly examining topology and dynamics, thereby enabling researchers to capture the interplay between stable structural features and transient temporal variations. The field is intrinsically multidisciplinary, drawing upon diverse methodologies from graph theory, dynamical systems, statistics, and machine learning. This methodological diversity, can pose significant challenges for researchers, particularly at the stage of initial data exploration.

In this context, effective visualization is needed to provide an intuitive entry point into the data and help guide subsequent analytical decisions. To this aim, we developped TempNetViz to visualize temporal network data, support early-stage exploration and highlight potential structural and temporal patterns. Particular attention was given to ease of usability, to make data exploration as intuitive as possible. TempNetViz was build using igraph and networkx, two of the most popular graph visualization packages in Python `[@csardi2006igraph; @hagberg2008exploring]`.

# Statement of need

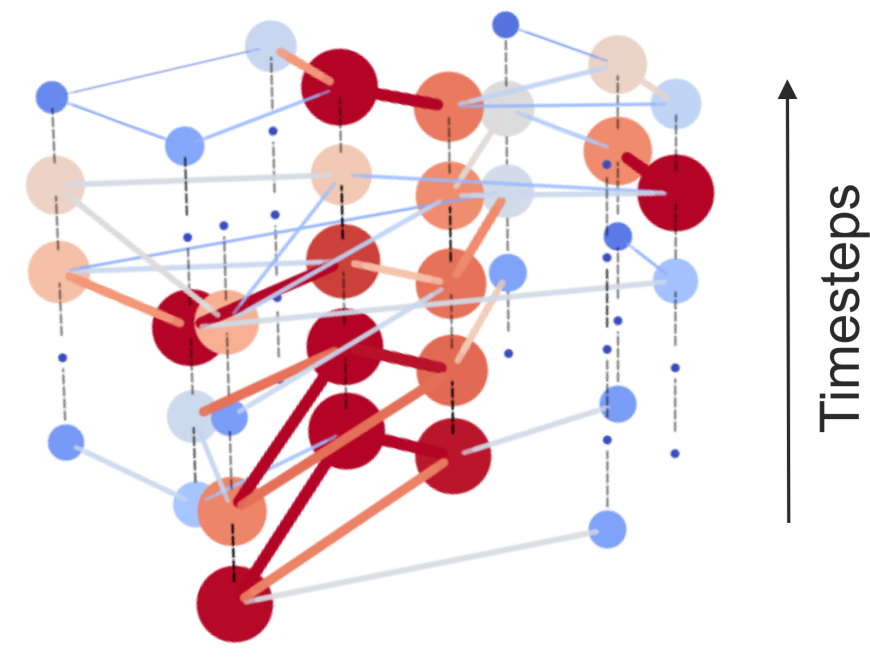
There are, to the best of our knowledge, no available graphical user interfaces (GUIs) designed specifically for the interactive visualization of temporal graphs. Although some established graph visualization GUIs provide ways for displaying temporal networks `[@bastian2009gephi; @shannon2003cytoscape]`, they were not tailored for this purpose. The available features are therefore limited and rely on specialized internal data formats or external plugins. This represents a substantial learning barrier that must be overcomed before any analysis can be conducted. Moreover, it makes systematic comparison across different data representations impossible.

While numerous Python packages for temporal graph analysis are available `[@passos2024networkx; @thompson2021teneto; @hackl2021analysis]`, they either offer limited visualization capabilities or are more oriented towards modeling and classification. Leveraging the strength of the existing dependencies also requires some coding expertise, which can represent a significant obstacle for researchers lacking extensive programming experience. To address these limitations, we developed, we developped TempNetViz to provide and accessible and user-friendly way to visualize temporal graphs.

# Key features

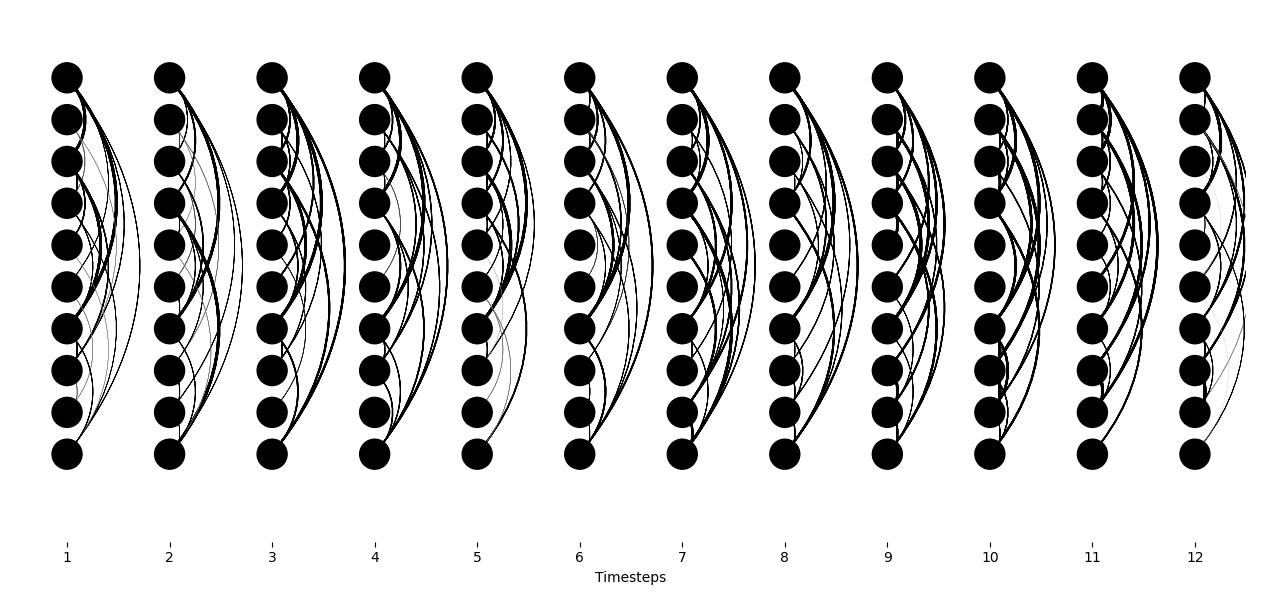
The main advantage of TempNetViz is to allow the user to switch effortlessly between different representations of the data. Once the analysis parameters have been chosen (for details, see the [documentation]), there are 4 possible ways of displaying the results. Below, we briefly present them and discuss their advantages and drawbacks.

The first one, called multi-layer representation, displays each time step as an individual graph within a layer. The layers corresponding to the different time steps are then stacked on top of each other, from bottom to top. The result is displayed in 3 dimensions (see Fig. 1) and can be rotate with the mouse to show different parts of the multi-layer graph. This is perhaps the most complete way of representing the data, but it can become overcrowded when too many timesteps are displayed at once.



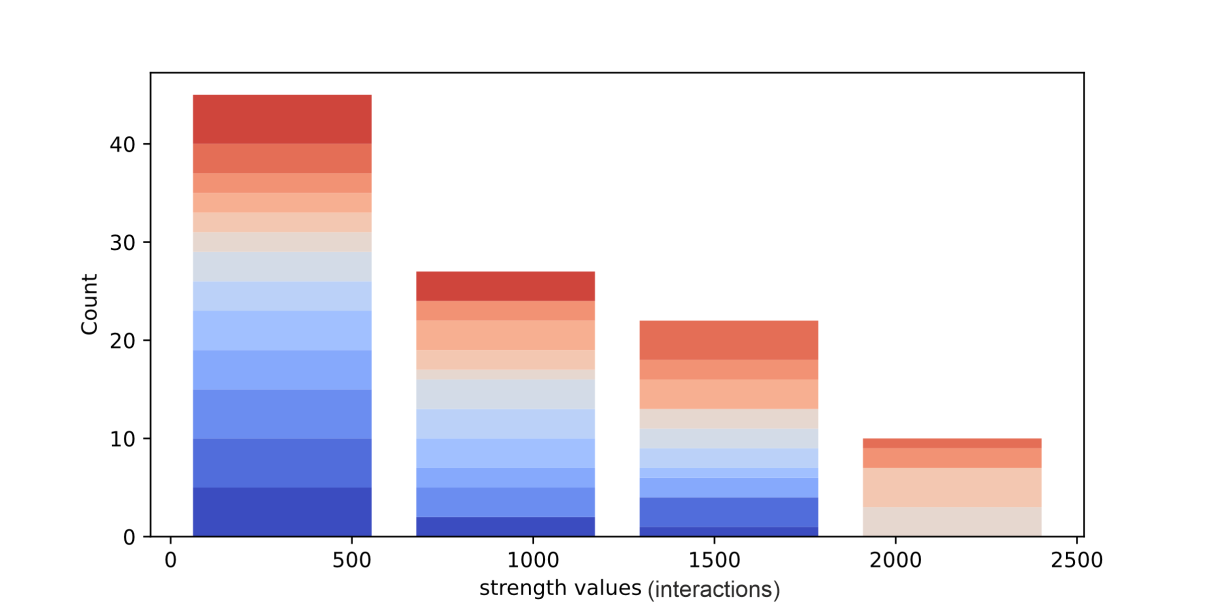
**Fig. 1**: **Example of multi-layer representation.** The color of the nodes indicate how strongly connected they are to others, from blue (low connectivity) to red (strong connectivity). The same color mapping is applied to edges between node to represent the magnitude of the edge.

Another way to show the data is to display the graph associated with each timestep sequentially to create an animation. While this makes the structure less explicit, it makes the observation of the time evolution more intuitive. A slider is also provided to allow a finer control over the steps that are being displayed. The third way to represent the data is through the so-called temporal layout `@linhares2023visualisation`. In this view, the nodes are ordered along the y-axis, and the edges between each nodes are shown at each time step (see Fig. 2).



**Fig. 2**: **Example of temporal layout.** Each column represents a time step, each row a node. Edges between two nodes are shown via curved lines.

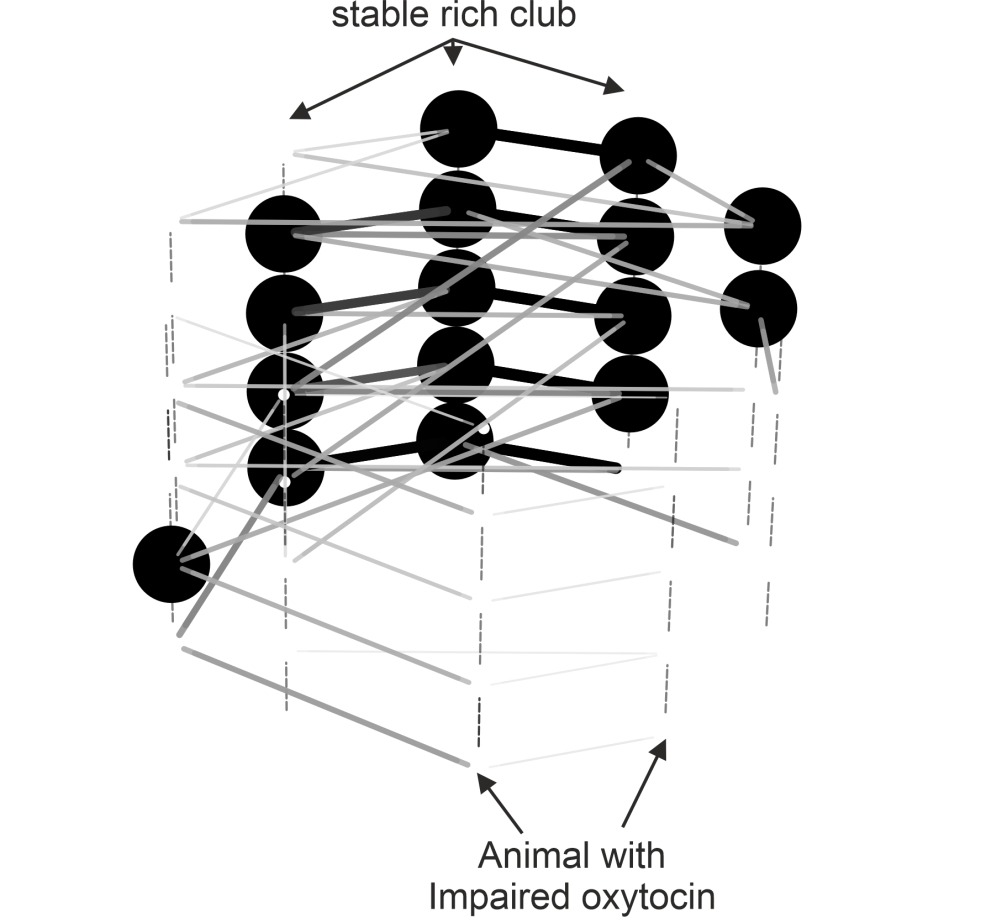
This representation makes the structure of the graph less obvious (no attempt is made at displaying strongly connected nodes closer to each other), but makes it very easy to see global changes of activity in the temporal graph. Finally, TempNetViz also gives the possibility to plot the values of various graph descriptors (see metric section in the [documentation]) as a function of time via stacked histograms. This entirely discards the information of structure, to make evolution of the data on a global scale more explicit (see Fig. 3).



**Fig. 3**: **Example of stacked histogram representation.** Color indicates time steps, from deep blue (first time step) to deep red (last time step). In this example, we see that the largest strength values (>1500) are only observed at later timesteps, indicating an overall increase in activity towards the end of the experiment.

# Research application

TempNetViz was used to study the formation of stable cliques (called rich-clubs) in mouse societies `@nelias2025stable`. In this article, it has been found that in groups of mice that evolving in a semi-naturalistic environments, highly social and stable cliques tend to form. Moreover, it was found that mice that having impaired cortical oxytocin signaling were not able to enter such stable rich-clubs, preserved overall social motivation. These findings helped to underscore the role of oxytocin in tuning sensory systems into a social processing state. In Fig. 4, we show how the main finding of this article can be seen using TempNetViz. In this example, we first prune the graph and retain only the edges between mutual nearest neighbors of 3rd order (graph-cut in TempGraphViz). Then, the nodes which are part of the rich club are displayed (rich-club metric in TempGraphViz). Nodes which are consistently part of the rich club for at least 4 out of 5 time steps are considered part of the stable rich club.



**Fig. 4: Example of stable rich club observation.** The graphs were first pruned via mutual nearest neighbors of order 3.Nodes which are part of the rich-club (via ‘metric’ menu in TempNetViz) of degree k > 3 are shown in black. We consider that node are part of the stable rich-club if they are part of the rich club in 4 out of 5 time steps. In this example, the nodes corresponding to mice with impeded oxytocin signaling were excluded from the stable rich club.

# Acknowledgement

The work was funded by BMBF 3R consortium grants ‘NoSeMaze1’ (161L0277A) and ‘NoSeMaze2’ (16LW0333K) to W.K., Leibniz Association program grant ‘Learning resilience’ (K430/2021) to W.K., Boehringer Ingelheim Foundation grant ‘Complex Systems’ to W.K., BMBF CRCNS grant ‘Oxystate’ (01GQ1708) to W.K, DFG CRC 379 Project C03 to W.K., and the DFG Clinician Scientist Program ‘Interfaces and Interventions in Complex Chronic Conditions’ (EB187/8-1) to J.R.

# References